ACOUSTIC SCENE CLASSIFICATION USING MULTI-LAYERED TEMPORAL POOLING BASED ON DEEP CONVOLUTIONAL NEURAL NETWORK

Technical Report

Zhang Liwen

Harbin Institute of Technology
School of Computer Science and Technology,
Harbin, Heilongjiang, China
lwzhang9161@126.com

Han Jiqing

Harbin Institute of Technology
School of Computer Science and Technology,
Harbin, Heilongjiang, China
jqhan@hit.edu.cn

ABSTRACT

The performance of an Acoustic Scene Classification (ASC) system is highly depending on the latent temporal dynamics of the audio signal. In this paper, we proposed a multiple layers temporal pooling method using CNN feature sequence as input, which can effectively capture the temporal dynamics for an entire audio signal with arbitrary duration by building direct connections between the sequence and its time indexes. We applied our novel framework on DCASE 2018 task 1, ASC. For evaluation, we trained a Support Vector Machine (SVM) with the proposed Multi-Layered Temporal Pooling (MLTP) learned features. Experimental results on the development dataset, usage of the MLTP features significantly improved the ASC performance. The best performance with 75.28% accuracy was achieved by using the optimal setting found in our experiments.

Index Terms— acoustic scene classification, temporal pooling, convolutional neural networks, support vector machine

1. INTRODUCTION

Environmental sound classification is one of the most crucial components in the computational auditory analysis and the essential preprocessing stage for the robust speech recognition system. With the organizations of the DCASE workshop [1, 2] in 2016 and 2017, more attentions have been drawn to the Acoustic Scene Classification (ASC) task. Thanks to all these submissions in last two years, these researchers have provided many excellent ideas and useful experiences for the new DCASE 2018 [3] challengers including us.

Roughly scanning the previous submissions in last two years, it is easy to find that, deep learned solutions were very popular with the researchers. Such as CNN [4, 5, 6, 7], RNN [8], DNN [9], GAN [10], and all of them had achieved good results in the ASC tasks of the challenge. In the DCASE 2017 leaderboard, the rank first was won by the GAN system proposed by [10], the second and the third place belonged to the CNN systems proposed by [5] and [4] respectively. Obviously, CNN is a powerful model for ASC task. We also use a CNN model as one of the crucial components for our proposed framework.

As the audio signal is a continuous sequence restrained by its chronological order, and its high-level semantics contain in the temporal structure of the sequence. Hence, it is reasonable to improve the performance of AER by using the temporal information. In DCASE 2017, A. Schindler *et al.* [7] proposed a CNN architecture which harnesses information from increasing temporal resolutions of Mel-Spectrogram segments. In contrast with the work of [7], we focus on how to capture the latent temporal information of the entire audio sample. To achieve this goal, we proposed a multiple layer temporal feature learning framework using CNN features as input, we call it Multi-layered Temporal Pooling (MLTP). Our temporal pooling method can map an audio sequence with arbitrary duration to a fixed length feature representation which can effectively capture the temporal information of the entire sequence. With the employment of Support Vector Regression (SVR) [11], this method is very efficient during the whole feature learning process. After generating of the temporal features for all the audio samples, we train a Support Vector Machine (SVM) with the ones on training set to conduct classification on testing dataset. Compared with the baseline system [3], experimental results showed our method brings absolute improvements of 15.2%. More details about our framework will be covered in Section 2.

2. PROPOSED FRAMEWORK

The framework of our proposed CNN based Temporal Pooling method is illustrated in Figure 1. The whole procedure mainly consists of three stages: the frame-level feature extraction for the waveform audio segment, the patch-level FC feature learning using pre-trained CNN model and the high-level temporal feature learning using our proposed temporal pooling method. Once the temporal features for all the audio segments in development dataset are generated, they will be classified with SVM.

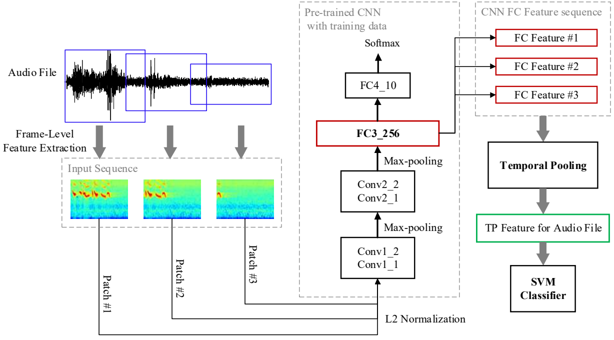

Figure 1: Processing of proposed framework

2.1. Proposed CNN Architecture

Inspired by [4, 12], we proposed a CNN followed a VGG style network for our classification task. The model architecture is similar with the one proposed in [12], which is shown in Table 1. Experimental results showed that, when we used the original network setting in [12] on the ASC development dataset, the model overfitting is rather serious. Hence, three modifications had been made to alleviate the overfitting of the model: (1) reducing the filter numbers by half in each convolutional layer and using filters with larger size; (2) enlarging the pooling size of the last max-pooling layer and adding dropout operation to each max-pooling layer; (3) using smaller full-connected layers and removing one of them. With these modifications, the performance of our CNN architecture had significantly improved and the parameters scale had been reduced from 2.3×10^8 to 3.8×10^5.

Furthermore, to accelerate convergence, we also performed Batch Normalization (BN) [13] for each convolutional layer and full-connected layer (except for the last FC layer) during training. More details about the network setting will be further discussed in Section 3.2.

Table 1: The architecture of proposed CNN.

Layer Type	Layer Description
Conv	Conv1_1 5×5 (1, 32) - BN - ReLu Conv1_2 5×5 (32, 32) - BN - ReLu
Pool	Max-Pooling 2×2 - Dropout (0.3)
Conv	Conv2_1 5×5 (32, 64) - BN - ReLu Conv2_2 5×5 (64, 64) - BN - ReLu
Pool	Max-Pooling 13×4 - Dropout (0.3)
FC	FC3_256 - BN - ReLu - Dropout (0.5)
	FC4_10 - ReLu
Softmax	10-way
#param: 3.8×10^5	

2.2. Multi-Layered Temporal Pooling

In this section, we present the main idea of proposed Multi-Layered Temporal Pooling method and how it works in our proposed framework. The performance of an ASC system is highly depending on the discriminative information in the categorical prior knowledge and the latent temporal dynamics of the audio signal. For one training audio sample, the proposed CNN can capture the effective discriminative information from its small patches and label, however, the individual learning mechanism would lose some important temporal dynamics of the whole sequence. Motivated by [14, 15], we proposed a temporal feature learning method based on a regularized SVR to capture the temporal variations in audio signals. As the framework shown in Figure 1, the temporal pooling method takes the mid-level CNN features as the input. Hence, to adequately model such deep learned complex features sequence, we extend the temporal pooling into the hierarchical multi-layered structure, we call it Multi-Layered Temporal Pooling.

For better understanding our MLTP method, a simple MLTP architecture with two layers is illustrated in Figure 2. Let $X^{(l)} = \{x_1^{(l)}, \ldots, x_t^{(l)}\}$ represent the input feature sequence for layer l. In this temporal pooling layer, we first conduct a non-linear feature mapping $\Psi(\cdot)$ for $X^{(l)}$, which is used for capture the complex dynamic information contains in the audio sequence.

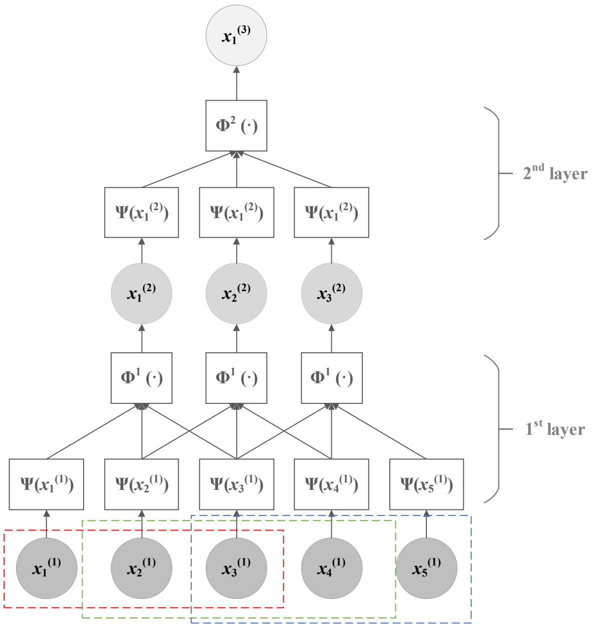

Figure 2: Multi-Layered Temporal Pooling with two layers.

In our work, we adopt χ^2 kernel and *posneg* kernel [16] to do non-linear feature mapping respectively. The latter one can be regarded as a simplified version of *Hellinger* kernel, where the primal form of *Hellinger* kernel function is given as,

$$K_{hell}(x,y)=\sqrt{x}^T\sqrt{y}\,,\qquad (1)$$
$$where\ \sqrt{x}=\sqrt{x^+}+i\sqrt{x^-}=\hat{x}^+ + i\hat{x}^-.$$

where x^+ and x^- represents the non-negative and the negative part of the input respectively. Directly using (1) can bring a very complex kernel, hence, we use the simplified version of *Hellinger* kernel called *posneg* as,

$$K_{\mathrm{Re}\{hell\}}=[\hat{x}^+,\hat{x}^-][\hat{y}^+,\hat{y}^-]^T=K_{hell}(x^*,y^*),\qquad (2)$$
$$where\ x^*=[x^+,x^-]^T.$$

where $K_{\mathrm{Re}\{hell\}}$ is the real part of K_{hell}, x^* represents the expansion of x which divides the original feature into the non-negative and the negative parts. Due to the using of ReLu in our CNN architecture, each activation layer only produces the positive values, however, the negative parts may also reflect dynamic changes in the sequence. It is somehow reasonable to consider both positive and negative activations of the sequence, therefore, we use the FC outputs before ReLu Units as the MLTP input feature. In our experiments, with the help of *posneg* feature mapping, the ASC performance improved.

Once the non-linear feature mapping complete, we will obtain the expanded feature sequence $\Psi(X^{(l)})$ for the temporal pooling. For better description, we use $V = \{v_1, \ldots, v_T\}$ represents the feature sequence $\Psi(X^{(l)})$. The temporal pooling operation $\Phi(\cdot)$ can be regarded as a sequence encoding, its goal is to transform V into a single fixed dimensional vector $\Phi(V)$ which captures the latent temporal information of the sequence. To achieve this goal, we use a linear function $f(v_t\,;u) = u^{\mathrm{T}}v_t$, *where* $v_t \in V$ with the parameters u to reconstruct the temporal information, and try to learn u such that f should satisfy the constraint given in (3),

$$f\left(v_{t_a}\right) < f\left(v_{t_b}\right),\quad \textit{where}\; t_a < t_b \tag{3}$$

where t_a and t_b are the time indexes of the input sequence. To find this optimal vector, we exploit a point-by-point optimization strategy based on SVR, which makes a direct connection between each v_t and t, the formula is given in (4),

$$\arg\min_{u}\left\{\frac{1}{2}\|u\|^2 + \frac{C}{2}\sum_{t=1}^{T}\left[\left|t-u^{\mathrm{T}}v_t\right|-\varepsilon\right]_{\geq 0}^{2}\right\} \tag{4}$$

where v_t is the feature vector at time t, and t is taken as the label, $[\cdot]_{\geq 0} = \max\{\cdot,\,0\}$ is the ε-insensitive loss function [17], the regularization factor C is a tradeoff between the flatness of fitting function and the reconstruction error tolerance range. When the algorithm meets the convergence conditions, the optimal vector u will be generated as the temporal feature for the input sequence.

For our MLTP, each successive layer captures the temporal dynamics of the output of the previous layer in the sliding window manner, each pooling operation takes the sub-sequence of the input with a fixed length. Hence, we can control the depth of the whole structure and the width of each layer by varying the stride and window size for each layer. According to the experimental experience, a 2-layer MLTP with window size three and stride two is sufficient for our task. After the temporal feature learning, we take the final vector as the representation of corresponding audio segment to train the SVM classifier, and then use it to classify the testing data in the development set.

3. EXPERIMENTS AND RESULTS

3.1. Feature Extraction

Through all the experiments, we convert the audio samples to 48 kHz sampling rate, 16 bits/sample, mono channel (the mono channel samples were generated by averaging the two channels samples). Before training our proposed CNN, we extracted 40, 80 and 120 bands log-mel energies for each converted audio samples respectively with frame size 40ms and 50% overlap. In our work, the frame-level log-mel energies can be regarded as the low-level descriptors, and to obtain the temporal information during the MLTP training, each low-level descriptor should be split into smaller patches with fixed length. Finally, each patch has 50 frames (i.e. 1sec), and patches with three different sizes: (50, 40), (50, 80) and (50, 120) were respectively used as the network inputs to train the CNN models.

3.2. CNN Training Parameters and Results

During training, the initial learning rate was set to 0.1 and decreased in the logarithmic domain every epoch. The training strategy was mini-batch gradient descent based on back propagation with 128 batch size and 0.9 momentum. To alleviate overfitting, the L2-Regularization was added to the cross-entropy loss with a weight decay of 0.001, and we also applied dropout [1] after the first, second max-pooling layer and the last full-connected layer with probabilities 0.3, 0.3 and 0.5.

Before training, we first normalized the log-mel energies for each frame, and then we calculated the means and the standard deviations through time on the training set to standardize all the input patches. The input patches used for the training were split with 25% overlap, but during testing, the patches were extracted with no overlap. In our work, the network is used to generate the mid-level descriptors for the MLTP training, hence we only evaluate the networks after 20 epohs with the patch-wise accuracy.

Table 2: The patch accuracy comparison of the proposed convolutional neural network under different input patches within 20 epochs.

Log-mel Bands	Training set Overlap	Best Acc. at Epoh	Patch Acc. [%]
40	0%	12	62.1
40	25%	10	63.5
80	0%	14	63.7
80	25%	11	65.0
120	0%	8	64.6
120	25%	8	**65.6**

Results in Table 2 showed that, when 120 log-mel bands energies and 25% patch overlap were used, the network achieved the best performance. And by using Batch Normalization for the outputs of each layer, the model can convergence within 20 epohs during all the experiments. Then, we saved the model parameters with the highest patch-size accuracy to extract the mid-level descriptors for the next MLTP training.

3.3. Classifying with MLTP Features

We used the pre-trained CNN model with the best performance mentioned in Section 3.2 to get the last full-connected (FC) layer output for each input patch with 25% overlap in development dataset. Thus, each audio sample there would be represented by a mid-level features sequence formed by these FC vectors. Then, our proposed MLTP method would use this sequence as input to generate the MLTP feature for the corresponding audio sample.

As mentioned in Section 2.2, the MLTP framework is consist of two major components: the non-linear feature mapping and the temporal pooling based on SVR. In this case, we conducted two sets of experiments one by one to find the optimal MLTP configuration for our ASC task. In the first set, we fixed the feature mapping kernel type and SVM classifier kernel function with proposed *posneg* kernel and the χ^2 kernel respectively in advance. Under this condition, we attempted to investigate the optimal SVR penalty factor λ for the MLTP according to the classification accuracy. Results in Table 3 showed that, when the penalty factor λ was set to 1×10^{-5}, the proposed system achieved the best performance. Then this parameter will be fixed in the following experiments.

Table 3: Classification accuracy comparison of the proposed CNN based MLTP method under different SVR penalty factors.

MLTP Architecture	Feature Mapping	λ	SVM Kernel	Acc. [%]

Layers = 2
1st layer:
Win = 3;
Stride = 2;
2nd layer:
Win = -1;
Stride = 0;
Note: -1 represents the whole sequence
	posneg	1	χ^2	72.99

		0.1		72.44
		0.01		72.64
		1×10^{-3}		73.07
		1×10^{-4}		73.03
		1×10^{-5}		**73.11**
		1×10^{-6}		73.03

Table 4: Classification accuracy comparison of the proposed method using different kernels for feature mapping and SVM.

Feature Mapping	SVM Kernel	Acc. [%]
posneg	*posneg*	74.15
	χ^2	73.57
	posneg + χ^2	74.23
χ^2	*posneg*	73.99
	χ^2	73.33
	posneg + χ^2	74.07
posneg + χ^2	*posneg*	**75.28**
	χ^2	74.32
	posneg + χ^2	74.32

By using the optimal penalty factor found in the first set of experiments, we further investigated how different kernel types applied in feature mapping and SVM classifier effect the performance of our proposed method. As the results illustrated in Table 4, when we combined the two types of kernel: *posneg* and χ^2 in the feature mapping, the system achieved better performance than the cases of using them alone. And the best accuracy 75.28% was achieved when the *posneg* kernel was applied for the classifier.

To have a better observation about the performance of our proposed temporal pooling method MLTP_CNN, the class-wise accuracies of the optimal configuration found in the above two sets of experiments are further given in Table 5. Comparing with the baseline system provided by DCASE 2018 [3], the average class-wise accuracy had been improved by our method with absolute 15.6 percent.

Table 5: The class-wise accuracy comparison on the development dataset.

Scene Label	Accuracy [%]	
	Baseline [3]	CNN_MLTP
Airport	72.9	76.6
Bus	62.9	74.0
Metro	51.2	72.8
Metro Station	55.4	75.7
Park	79.1	83.1
Public Square	40.4	56.0
Shopping Mall	49.6	81.7
Street Pedestrian	50.0	70.8
Street Traffic	80.5	87.4
Tram	55.1	74.7
Average	**59.7 (± 0.7)**	**75.3(± 0.1)**

3.4. Submissions

All the experiments shown in Table 2-5 were conducted with the default training/testing split of the DCASE 2018 development dataset. To achieve better final evaluation results for task 1, we utilize the full development dataset to our final model. We submit two outputs for our system using two different SVM penalty factors: 0.05 and 0.1, which have achieved 70.66% and 73.33% accuracy respectively on the Kaggle leaderboard dataset [18]. The other parameters for the submitted system were choosed based on the optimal settings found in our previous experiments on the development set.

4. CONCLUSION AND FUTUREWORK

In this report, we proposed an efficient novel multiple layer temporal feature learning method for ASC task, which can effectively capture the temporal dynamics for an entire audio data with arbitrary duration. The experimental results on the development dataset showed that, our MLTP method indeed improved the ASC performance. Moreover, based on our experience in CNN training part, the pre-trained model with better performance is more helpful to the proposed framework.

However, for the shortcomings in our present work, there is still room for improvements. For the MLTP part, we only use the patches with one second duration, more experiments with multiple time scale patches should be conducted in the future. For the networks part, more suitable frame-level features should be selected, useful data augmentation technology should be considered and better structured networks should be adopted. Furthermore, the present temporal feature learning process is divided into two parts, which is lack of simplicity. Therefore, more concise end-to-end formed framework which can jointly learn the discriminative information and temporal dynamics should be proposed in the future.

5. ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China under grant No. 61471145 and U1736210.

6. REFERENCES

[1] T. Virtanen, et al., *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016).* Tampere University of Technology. Department of Signal Processing, 2016.

[2] A. Mesaros, et al., "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Munich, 2017.

[3] http://dcase.community/workshop2018/.

[4] W. Zheng, J. Yi, X. Xing, X. Liu and S. Peng, "Acoustic Scene Classification Using Deep Convolutional Neural Network and Multiple Spectrograms Fusion", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[5] Y. Han and J. Park, "Convolutional Neural Networks with Binaural Representations and Background Subtraction for

Acoustic Scene Classification", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[6] R. Hyder, S. Ghaffarzadegan, Z. Feng and T. Hasan "BUET Bosch Consortium (B2C) Acoustic Scene Classification Systems for DCASE 2017", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[7] A. Schindler, T. Lidy and A. Rauber, "Multi-Temporal Resolution Convolutional Neural Networks for the DCASE Acoustic Scene Classification Task", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[8] I. Kukanov, V. Hautamäki and K. A. Lee, "Recurrent Neural Network and Maximal Figure of Merit for Acoustic Event Detection", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[9] J. W. Jung, H. S. Heo, I. H. Yang, et al. "DNN-Based Audio Scene Classification for DCASE 2017: Dual Inputfeatures, Balancing Cost, and Stochastic Data Duplication", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[10] S. Mun, S. Park, D. Han and H. Ko, "Generative Adversarial Network Based Acoustic Scene Training Set Augmentation and Selection Using SVM Hyper-Plane", *IEEE AASP Challenge on DCASE 2017 technical reports*, 2017.

[11] A. Smola and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155-161, 1997.

[12] Takahashi, Naoya, et al, "Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition." in *Proc. INTERSPEECH* 2016:2982-2986.

[13] Ioffe, Sergey, and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv:1502.03167, 2015.

[14] B. Fernando, et al. "Modeling video evolution for action recognition." *Computer Vision and Pattern Recognition IEEE*, 2015:5378-5387.

[15] B. Fernando, et al. "Discriminative Hierarchical Rank Pooling for Activity Recognition." *Computer Vision and Pattern Recognition IEEE*, 2016:1924-1932.

[16] B. Fernando, E. Gavves, et al, "Rank Pooling for Action Recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence* 39.4 (2017): 773-787.

[17] A. Smola and V. Vapnik, "Support vector regression machines", *Advances in Neural Information Processing Systems*, vol. 9, pp. 155-161, 1997.

[18] https://www.kaggle.com/c/dcase2018-task1a-leaderboard.