Interspeech 2018
2-6 September 2018, Hyderabad

Unsupervised Temporal Feature Learning Based on Sparse Coding Embedded BoAW for Acoustic Event Recognition

Zhang Liwen[1], *Han Jiqing*[1], *Deng Shiwen*[2]

[1]Harbin Institute of Technology, China
[2]Harbin Normal University, China
Lwzhang9161@126.com, jqhan@hit.edu.cn

Abstract

The performance of an Acoustic Event Recognition (AER) system highly depends on the statistical information and the temporal dynamics in the audio signals. Although the traditional Bag of Audio Words (BoAW) and the Gaussian Mixture Models (GMM) approaches can obtain more statistics information by aggregating multiple frame-level descriptors of an audio segment compared with the frame-level feature learning methods, its temporal information is unreserved. Recently, more and more Deep Neural Networks (DNN) based AER methods have been proposed to effectively capture the temporal information in audio signals, and achieved better performance, however, these methods usually required the manually annotated labels and fixed-length input during feature learning process. In this paper, we proposed a novel unsupervised temporal feature learning method, which can effectively capture the temporal dynamics for an entire audio signal with arbitrary duration by building direct connections between the BoAW histograms sequence and its time indexes using a non-linear Support Vector Regression (SVR) model. Furthermore, to make the feature representation have a better signal reconstruction ability, we embedded the sparse coding approach in the conventional BoAW framework. Compared with the BoAW and Convolutional Neural Network (CNN) baselines, experimental results showed our method brings improvements of 9.7% and 4.1% respectively.

Index Terms: acoustic event recognition, temporal feature learning, bag of audio words, sparse coding.

1. Introduction

In the real audio world, there exist various acoustic events, detecting and recognizing these sound elements can provide a significant help for many audio signal processing applications including auditory scene analysis, audio semantic comprehension, speech enhancement and audio content retrieval. For instance, in a lecture speech content transcription task, the recording usually contains many non-speech elements like applauses, laughter and music. In order to extract the pure speech content, detection of these non-speech elements is the essential precondition [1]. Further, as in the enormous internet multimedia data, recognizing variable kinds of audio events can be very helpful to category these files [2] and establish efficient indexes [3, 4, 5, 6]. In addition, AER technology can also be used in surveillance area for safety concern [7].

As the crucial component of AER, one of the most important goals of feature learning is to capture the underlying characteristics in audio signals and obtain the discriminative feature representations. In the past few decades, many efforts had been made towards this direction. For example, inspired by the achievements of speech recognition, Mel-Frequency Cepstral Coefficients (MFCC) were extracted from the audio signals as the frame-level features for acoustic event detection [8, 9]. However, unlike the speech signals, acoustic events have no specific semantic units, directly mapping the frame-level features to manual labels may bring highly confusions in models. In this regard, more attentions were drawn to the high-level feature learning methods. Such as building GMMs for spectro-temporal features of audio segments [10, 11], and applying the BoAW [12, 13, 14] method to obtain a segment-level feature representation by computing histogram of audio words sequence after feature encoding. In contrast with the frame-level feature learning methods, the high-level features learned by these methods contain more context information, however the temporal information in the original signals is unreserved. As it is generally known, 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, thus it is reasonable to improve the performance of AER by using the temporal information.

More recently, following the huge success in machine learning, various DNN based feature learning methods were employed to capture temporal information in audio sequence. In [15], Deep Belief Networks (DBN) was used to extract bottleneck features and the temporal information was reserved by sequential frames within a sliding window batched as network input. In [16], the continuous frame-level features were scaled into a high-dimension supervector, then a fully DNN structure was employed to learn high-level features using this supervector as input. Similarly, inspired by VGG Net [17], Naoya *et al.* [18] proposed two novel CNN structures, and attempted to obtain the feature representations which can reserve the temporal information by using these deep structures with large input fields. All these methods have improved the performance of AER system significantly. However, all these methods tried to capture the local changes in a fixed-length time window. They are not designed to model temporal dynamics of the entire audio signal within the associated events, and they usually require manually annotated labels during the feature learning.

In this study, we proposed a novel unsupervised temporal feature learning method based on BoAW framework for AER, which is inspired by the Rank-Pooling [19] strategy. Our 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 SVR [20], this method is very efficient during the whole feature learning procedure, and since the parametric representation strategy is used, we do not require negative data during feature learning. At last, we trained a supported vector machine (SVM) classifier to recognize all the acoustic events occurred in test data. Furthermore, to make the feature representation possess a better signal reconstruction

10.21437/Interspeech.2018-1243

ability, the sparse coding method was embedded in BoAW framework. Compared with BoAW and two different deep CNN baselines, experimental results showed our method brings absolute improvements of 9.7%, 6.5% and 4.1% respectively.

2. Temporal Feature Learning Based on BoAW

The whole temporal feature learning procedure consists of two major components: the BoAW feature learning and the temporal feature learning. And the manually annotated labels are not required throughout the entire temporal feature learning procedure.

2.1. Learning Bag of Audio Words

BoAW is one of the effective methods for audio content retrieval, sound activity detection and AER, it mainly consists of two steps: dictionary learning and feature encoding. In this paper, the classical k-means algorithm Lloyd [21] is applied to cluster MFCC features of each frame in training dataset for the dictionary learning. After that, the generated codebook is used to do feature encoding frame-by-frame with the Euclidean distance between MFCC vectors and code words as similarity measurement. Then, we calculate each code word occurrence frequency in corresponding audio segments to generate the BoAW histograms.

In contrast with the frame-level feature learning approach, BoAW can transform arbitrary length sequence into a fixed dimensional vector, which can be regarded as a higher-level representation to describe an entire acoustic event, while the whole process is totally unsupervised. However, the traditional BoAW framework only calculates the occurrence frequency of each audio word in corresponding sequence, the chronological order is out of consideration. So that these words cannot be placed in correct order according to the histogram, which directly causes the temporal information missing. To handle this problem, we first split a single audio file into several segments with equal length, then we compute histogram for each segment, so that every single file will be represented by a continuous histograms sequence *i.e.*, BoAW features. At last these BoAW features will be inputted into the temporal feature learning component as illustrated in section 2.3.

2.2. Sparse Coding for BoAW

In section 2.1, we use Euclidean distance as the similarity measurement to do feature encoding, and a distance threshold is set to control the approximate accuracy. Although this feature encoding method achieved a good performance in our experiments, its reconstruction ability to the original signals is still limited. For this problem, we further introduce the sparse coding with l_1 regularization into both dictionary learning and feature encoding steps during BoAW feature learning.

This sparse coding method essentially can be regarded as a regularized linear regression problem, where the regularization parameter is added to control the tradeoff between the approximation accuracy and the sparsity of representations. In the particular implementation process, we employ an online study strategy from [22] during dictionary learning, and the representing coefficients are regularized by the l_1-norm so that the sparsity can be satisfied. Meanwhile, to avoid dictionary from having arbitrarily large values which may lead to arbitrarily small values of coding results, we also constrain each code word to have an l_2-norm less than or equal to one. This sparse coding embedded dictionary learning is given as,

$$\begin{aligned} &\min_{D\in\Omega,\alpha\in\mathbb{R}^{k\times n}} \frac{1}{2}\|X - D\alpha\|_2^2 + \lambda\|\alpha\|_1, \\ &\Omega \triangleq \{D\in\mathbb{R}^{m\times k}\ s.t. \forall j = 1,\dots,k,\ d_j{}^T d_j \le 1\}. \end{aligned} \tag{1}$$

where X is the training dataset, its each column represents the MFCC feature of a single audio frame, D is the dictionary with k different code words d_j, α contains the representing coefficients regularized by l_1-norm with a parameter λ.

After dictionary learning, the codebook can be used to do feature encoding for MFCC of each audio frame. Solving the feature encoding problem as in (2) is essentially consistent with the dictionary learning problem above, which is a linear regression problem with constraints. The only difference is that the goal is to find the best coding results with a fixed codebook D,

$$c^* \triangleq \arg\min_c \frac{1}{2}\|x - Dc\|_2^2 + \lambda|c|_1 \tag{2}$$

where c^* is the coding result used as a representation of x on D, the parameter λ is used to control the tradeoff between the approximation accuracy and sparsity, when it equals 0, the problem will degenerate into a classical regression problem. Once the representation of each frame is generated, the histogram vector will be calculated for each audio segment using the same way as BoAW feature learning.

2.3. Temporal Feature Learning Based on SVR

The performance of an AER system is highly depending on the latent temporal dynamics constrained by the chronological order of the corresponding audio signal. As mentioned in section 2.1, the traditional BoAW framework did not take into account the order of the code words, which would lose the temporal information. In this study, we proposed a temporal pooling strategy based on a regularized SVR to capture the temporal variations in audio signals. This strategy takes segment-level descriptors and their time indexes as the input sequence and the labels respectively. Hence, our proposed method builds a direct relationship between audio sequence and its chronological order, and it does not require fixed-length input and extra labels. As the processing steps of our temporal feature learning shown in Fig. 1, we first smooth the BoAW features $c_1 \cdots c_t$ of each audio to improve the robustness, then we do a non-linear expansion for the smoothed sequence $v_1 \cdots v_t$ to embed the descriptors into higher dimension space, because of the high complexity of the audio signals. At last, after the temporal pooling with expanded feature sequence, we get a dimension fixed temporal feature u for each audio.

2.3.1. Feature Smoothing

The audio signals generally contain high variety due to the complexity of acoustics environments, which may probably cause disappointed regression effects. To reduce the effect of violent variations and improve the robustness of the AER system, each input sequence should be smoothed before the temporal pooling. In this paper, we use Time Varying Mean (TVM) [19] method to smooth the BoAW features sequence of each audio.

$$v_t = \frac{m_t}{\|m_t\|}, \mathit{where}\ m_t = \frac{1}{t}\sum_{\tau=1}^{t}\frac{s_\tau}{\|s_\tau\|} \tag{3}$$

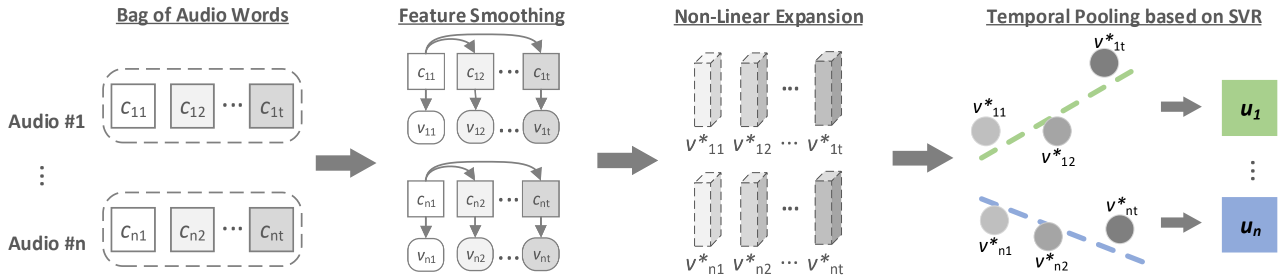


Figure 1: *Processing steps of proposed temporal feature learning based on BoAW.*

As shown in (3), where m_t is the mean vector of input frames: $s_1 \cdots s_t$, then each mean vector m_t should be normalized to obtain the smoothed TVM vector v_t.

2.3.2. *Non-Linear Feature Mapping*

In order to incorporate non-linearities, we employ non-linear feature maps [23] on each smoothed BoAW feature. The idea is to assume a non-linear point-wise operator $\Phi(\cdot)$ which can map the input vector x to a higher-dimensional feature $\Phi(x)$. Then our feature learning method will capture the latent temporal variations in a more complex non-linear space.

In this paper, we adopt χ^2 kernel and *posneg* kernel [19] 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,

$$\begin{aligned} &K_{hell}(x,y) = \sqrt{x}^T \sqrt{y}\,, \\ &where\ \sqrt{x} = \sqrt{x^+} + i\sqrt{x^-} = \hat{x}^+ + i\hat{x}^-. \end{aligned} \tag{4}$$

where x^+ and x^- represents the non-negative and the negative part of the input respectively, hence, directly using (4) can bring a very complex kernel. To avoid such complexity, we use the simplified version of *Hellinger* kernel called *posneg* as,

$$\begin{aligned} &K_{\mathrm{Re}\{hell\}} = [\hat{x}^+, \hat{x}^-][\hat{y}^+, \hat{y}^-]^T = K_{hell}(x^*, y^*), \\ &where\ x^* = [x^+, x^-]^T. \end{aligned} \tag{5}$$

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.

2.3.3. *Temporal Pooling Based on SVR*

For a single audio file, once all the MFCC extraction, BoAW feature learning, feature smoothing, and non-linear mapping complete, we will obtain the input feature sequence $V = [v_1, \ldots, v_T]$ for the temporal pooling. During this step, our goal is to effectively encode the latent temporal information D which reflects the variation of each representation v_t in V from arbitrary time t to $t + 1$. To achieve this goal, we use a linear function $\Psi_u(V\ ;\ u)$ with the parameters u to reconstruct the dynamic information D by using the regression technology, then this feature learning problem will be transformed into a optimization problem as shown in (6).

$$\arg\min_u \|D - \Psi_u\| \tag{6}$$

The chronological order of the audio signals reflects the occurrence location of each element in the audio and the evolutionary trend of the whole sequence, so that D can be regarded as a ranking rule used to place each element by obeying such order. Hence, in order to capture this temporal information, the feature learning must be constrained by the chronological order of the audio. The basic idea is to find the optimal parameters vector u for $\Psi_u(V\ ;\ u)$ in (6), which must satisfy all the constraints: for any $a < b$, we have $u^{\mathrm{T}}v_a < u^{\mathrm{T}}v_b$, where a and b are the time indexes of the input sequence. In order 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 (7),

$$\arg\min_u \left\{ \frac{1}{2}\|u\|^2 + \frac{C}{2}\sum_{t=1}^{T}\left[\left|t - u^{\mathrm{T}}v_t\right| - \varepsilon\right]_{\geq 0}^2 \right\} \tag{7}$$

where v_t is the feature vector at time t in the corresponding input sequence, and t is taken as the label, $[\cdot]_{\geq 0} = \max\{\cdot, 0\}$ is the ε-insensitive loss function [20], the regularization factor C is a tradeoff between the flatness of fitting function $\Psi(t, V; u)$ 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 corresponding audio file.

3. Experiments

3.1. Dataset and Features

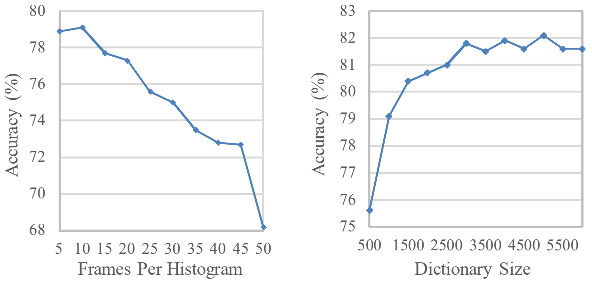
Our methods are evaluated on the acoustic event dataset provided by [18], which contains 5223 recorded files with 28 different kinds of acoustic events. And each file lasts from 3 to 12 seconds, the total duration is about 768 minutes. Before feature extraction, all audio samples were transformed into the uniform waves: 16 kHz sampling rate, 16 bits/sample, mono channel, and the dataset was split into the training set (75%) and the testing set (25%) with the same scheme as in [18]. We extracted 40 dimensional MFCC features with 0th cepstrum coefficients, log-energy and their delta and delta-delta for each audio sample as the frame-level descriptors. Except for sparse coding embedded BoAW feature learning, 25ms frame size and 10ms shift length were used for the rest experiments.

3.2. Experimental results and Discussions

3.2.1. *BoAW as High-Level Descriptors*

In our study, we use the BoAW to aggregate multiple frame-level MFCCs to obtain the high-level descriptors. Obviously, the best frame number of MFCC features for one BoAW histogram calculation needs to be found, so that the generated temporal feature can preserve sufficient temporal dynamics without losing too much discriminative ability. Meanwhile, the optimal codebook size is also needed to be verified.

To investigate the optimal frame number for one BoAW histogram and the proper dictionary size, we conducted first two sets of experiments, and χ^2 kernel was used for both non-linear feature maps and SVM classifiers through all the experiments in this section. In the first set, we fixed the dictionary size to 1000 to find the best frame number for each BoAW histogram. As the results illustrated in Fig. 2 (a), the performance peaked when 10 frames were selected for each histogram. Then, in the second set, we continuously observed the performance changes under different dictionary sizes from 500 to 6000 with 10 frames MFCC per histogram. Results in Fig. 2 (b) show the recognition accuracy roughly increased when dictionary size ranged from 500 to 5000, and reached peak (82.1%) when dictionary size became 5000.



(a) (b)

Figure 2: *Accuracy under different frame number for one BoAW histogram using fixed dictionary size: 1000 (a), and under different dictionary sizes using 10 frames per BoAW feature (b).*

In addition, BoAW takes MFCC of each audio frame as the coding primitive, and the frame size is a tradeoff between each code word explanatory ability and the generalization of the whole codebook. Therefore, in this section we conducted another set of experiments to further observe the performance changes under frame size at different scales using the optimal configuration found in last two sets of experiments. From the results shown in Table 1, we see the best performance (82.7%) was achieved when frame size was 720 samples (45ms).

Table 1: *Accuracy under different frame sizes using 10 frames per BoAW feature and 5000 for dictionary size.*

Frame Size (samples)	Accuracy (%)
400	82.1
480	81.7
560	81.8
640	82.0
720	**82.7**
800	82.1
880	81.1

3.2.2. BoAW with Sparse Coding

In this section, we verified the effectiveness of sparse coding for our BoAW feature learning. Similarly, a set of experiments was conducted to find the proper dictionary size for sparse coding. In this set, the frame size for MFCC extraction and the frame number per BoAW feature were determined as the optimal configuration in the former experiments. Moreover, based on the experimental results in [22] the regularization coefficient in (1) and (2) was selected as 0.15, and two types of kernels: χ^2 and *posneg* were adopted for non-linear feature maps. Experimental results in Table 2 show that when dictionary size became 756, the system achieved the highest accuracy: 83.8%. Hence, the sparse coding can effectively improve our system with much smaller dictionary size.

Table 2: *Accuracy under different dictionary sizes for BoAW with sparse coding.*

Dictionary Size	Accuracy (%)	
	χ^2	*posneg*
504	81.2	82.6
630	82.1	82.8
756	**82.5**	**83.8**
882	82.4	83.3
1008	82.3	82.8

Much smaller codebook size makes our sparse coding embedded method much more efficient than the BoAW based method. This means the temporal pooling can capture more temporal information with less frames per histogram without gaining too much computation costs. Results in Table 3 show that the performance improved as the number of frames per histogram decreases for the dictionary with 756 entries. Here, when the number of frames is one, the coding result for each frame is directly used as the temporal pooling input.

Table 3: *Accuracy under smaller frame number per histogram for BoAW with sparse coding.*

Frames per Histogram	Accuracy (%)	
	χ^2	*posneg*
10	82.5	83.8
5	83.0	84.1
1	**83.1**	**84.4**

3.2.3. State-of-the-art Comparison

We compared our proposed temporal feature learning methods to BoAW + SVM as in [12], and two deep CNNs: CNN_A and CNN_B proposed by [18]. Results in Table 4 show that, without implementing data augmentation, our proposed method based on BoAW outperforms three baseline systems with 8.0%, 4.8% and 2.4% improvements respectively. Moreover, by introducing the sparse coding into BoAW framework, our method achieved further 1.7% improvement compared with the proposed method without sparse coding.

Table 4: *Accuracy of our methods and baseline methods.*

Methods	Accuracy (%)
BoAW + SVM	74.7
CNN_A	77.9
CNN_B	80.3
BoAW + Temporal-Pooling + SVM	82.7
BoAW_SC + Temporal-Pooling + SVM	**84.4**

4. Conclusions

In this study, we proposed an efficient novel temporal feature learning method for AER task, which can effectively capture the temporal dynamics for an entire audio data with arbitrary duration, and the whole learning procedure is unsupervised. Moreover, by introducing l_1-norm sparse coding into BoAW framework, the system recognition accuracy achieved further improvements. Future work will focus on the data augmentation techniques applied for our proposed methods, and further explore enhancements on the temporal pooling strategy for other applications in audio signal processing.

5. Acknowledgements

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

6. References

[1] A. Ozerov, A. Liutkus, and R. Badeau, “Informed source separation: Source coding meets source separation,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASSPA)*, vol. 2, 2011, pp. 8–11.

[2] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, “Consumer video understanding: A benchmark database and an evaluation of human and machine performance,” in *Proc. ICMR*, 2011.

[3] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, “Audio keywords generation for sports video analysis,” *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 4, no. 2, pp. 1–23, 2008.

[4] X. Zhuang, X. Zhou, M. A. Hasegawa-johnson, T. S. Huang, “Real-world acoustic event detection,” *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543-1551, 2010.

[5] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, “Contextdependent sound event detection,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 1, pp. 1-13, 2013.

[6] D. Giannoulisy, E. Benetosx, D. Stowelly, M. Rossignolz, M. Lagrangez and M. Plumbley, “Detection and Classification of Acoustic Scenes and Events: an IEEE AASP Challenge,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[7] W. Choi, J. Rho, D. K. Han, and H. Ko, “Selective background adaptation based abnormal acoustic event recognition for audio surveillance,” in *IEEE Conference on Advanced Video and SignalBased Surveillance, AVSS*, 2012, pp. 118–123.

[8] C. Zieger, “An HMM based system for acoustic event detection,” *Multimodel technologies for perception of humans*, pp. 338-344, 2008.

[9] A. Temko, C. Nadeu, and J. I. Biel, “Acoustic Event Detection: SVM-Based System and Evaluation Setup in CLEAR’07,” *Multimodel technologies for perception of humans*, pp. 354-363, 2008.

[10] Z. Huang, Y. Cheng, K. Li, V. Hautamaki, C. Lee, “A Blind Segmentation Approach to Acoustic Event Detection Based on I-Vector,” in *Proc. Interspeech*, pp. 2282–2286, 2013.

[11] K. Lee, and D. P. W. Ellis, “Audio-Based Semantic Concept Classification for Consumer Video,” *IEEE Transactions on Audio Speech & Language Processing* 18.6(2010):1406-1416.

[12] S. Pancoast, M. Akbacak, “Bag-of-Audio-Words Approach for Multimedia Event Classification,” in *Proc. Interspeech*, 2013.

[13] H. Lim, M. J. Kim, and H. Kim, “Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation,” in *Proc. INTERSPEECH*, 2015, pp. 3325–3329.

[14] X. Lu, P. Shen, Y. Tsao, C. Hori, and H. Kawai, “Sparse representation with temporal max-smoothing for acoustic event detection,” in *Proc. INTERSPEECH*, 2015, pp. 1176–1180.

[15] I. Mcloughlin, et al, “Robust Sound Event Classification Using Deep Neural Networks,” *IEEE/ACM Transactions on Audio Speech & Language Processing* 23.3(2017):540-552.

[16] S. Mun, et al, “Deep Neural Network Bottleneck Features for Acoustic Event Recognition,” in *Proc. INTERSPEECH* 2016: 2954-2957.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015, pp. 1–14.

[18] N. Takahashi, et al, “Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition.” In *Proc. INTERSPEECH* 2016:2982-2986.

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

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

[21] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans* 28.2(1982):129-137.

[22] J. Mairal, F. Bach, et al, “Online Learning for Matrix Factorization and Sparse Coding,” *Journal of Machine Learning Research* 11.1(2010):19-60.

[23] A. Vedaldi, and A. Zisserman, “Efficient additive kernels via explicit feature maps,” *IEEE Transactions on Pattern Analysis & Machine Intelligence* 34.3(2012):480-492.