Robust Audio-Codebooks for Large-Scale Event Detection in Consumer Videos

Shourabh Rawat, Peter F. Schulam, Susanne Burger, Duo Ding, Yipei Wang and Florian Metze

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

{srawat,pschulam,sburger,dding,yipeiw,fmetze}@cs.cmu.edu

10.21437/Interspeech.2013-654

Abstract

In this paper we present our audio based system for detecting "events" within consumer videos (e.g. You Tube) and report our experiments on the TRECVID Multimedia Event Detection (MED) task and development data. Codebook or bag-of-words models have been widely used in text, visual and audio domains and form the state-of-the-art in MED tasks. The overall effectiveness of these models on such datasets depends critically on the choice of low-level features, clustering approach, sampling method, codebook size, weighting schemes and choice of classifier. In this work we empirically evaluate several approaches to model expressive and robust audio codebooks for the task of MED while ensuring compactness. First, we introduce the Large Scale Pooling Features (LSPF) and Stacked Cepstral Features for encoding local temporal information in audio codebooks. Second, we discuss several design decisions for generating and representing expressive audio codebooks and show how they scale to large datasets. Third, we apply text based techniques like Latent Dirichlet Allocation (LDA) to learn acoustic-topics as a means of providing compact representation while maintaining performance. By aggregating these decisions into our model, we obtained 11% relative improvement over our baseline audio systems.

Index Terms: Multimedia Event Detection, Video Retrieval, Audio-Codebook Models

1. Introduction

With the growth of multimedia content over the internet, there has been an increased interest in improving the current state-of-the-art in video retrieval, indexing and multimedia analysis. This has led to several research initiatives such as the TRECVID Multimedia Event Detection (MED) and Recounting (MER) tasks. MED [1] is the task of indexing and searching large corpora of multimedia content in order to retrieve from the collection the most relevant videos that show instances of certain predefined events. Examples of such events include "Birthday Party", "Wedding Ceremony", etc and are defined using textual descriptions along with several example videos. The best performing systems on this task including our system, combine features from multiple modalities, including visual and audio. Even though visual information contributes the most, audio is found to be significantly important for events where visual knowledge is either missing or insufficient. Identifying spoken utterances in speech can provide significant clues about the content of the video, for instance, detecting "cake" or "baking" in a "Making a Cake" event or "happy birthday" in a "Birthday Party" event. Similarly, identifying acoustic sounds can be indicative of certain activities in the video, an "engine" sound could confirm the presence of a vehicle while cheering or crowd noises might be indicative of events showing sport activities.

Recent work has looked at various approaches to audio analysis for the task of event detection in consumer videos. One class of approaches makes use of a semantic vocabulary [2, 3], a set of predefined acoustic event detectors learned in a supervised manner over manually annotated data. Though this approach has the power of offering semantic intuition about the event, which is useful for video summarization (MER), it does not scale well to consumer videos which are highly unstructured and unconstrained and require high annotation costs.

The other approach is to learn the concepts or the acoustic units in a completely unsupervised manner from the data itself. A popular technique to learn these units is using a clustering algorithm such as k-means. While these acoustic units may not have an attached semantic label, their distribution can convey information about the event, often referred to as the Bag-of-Audio-Words (BoAW) model. The BoAW model has been successfully applied to several content-based audio retrieval tasks including MED [4, 5].

Recently, a couple of other promising unsupervised techniques have been proposed that have shown improvements over the traditional BoAW model. Gaussian super-vectors, which have been successful for the speaker verification task [6], have also been applied to MED [7, 4]. First a universal background Gaussian Mixture Model (UBM) is trained followed by maximum aposteriori (MAP) adaptation of the means of the mixture components based on data from each video. The super-vector representation is then constructed by stacking the means and variances of the adapted models, and is then used for classification. Approaches like the Acoustic Segment Models (ASM) [8] and the Acoustic Unit Descriptors (AUD) [9] try to model temporal aspects of sound by decoding the audio signal into a sequence of acoustic segments using a fixed state HMM. The video is then represented as a bag-of-words over the n-grams of these acoustic units. While these approaches report improvements over the BoAW models, they are computationally more expensive.

In this paper we detail out some of the parameters and their effects on the performance of the Audio-Codebook model. We draw similarities between the Bag-of-Audio-Words and Bag-of-Visual-Words based on a similar analysis [10, 11, 12, 13] done in the past on visual categorization, and show empirically how different design decisions influence the expressiveness of the Audio-Codebook. Our work is closely related to an earlier analysis done by [5] on Bag-of-Audio-Words model on the MED dataset. We build on their work by performing a more detailed analysis on a much bigger dataset and with more event classes. Major contributions of this paper are:

- We report results on the largest publicly available consumer video dataset. While previous work have reported experiments on smaller datasets and with fewer events, we hope that the scale of these results would be a good reference for future audio based MED research.
- We evaluate low level feature representations that result

in more expressive codebooks and help to capture temporal information.

- We analyze parameters for codebook generation such as sampling method, codebook size, encoding, etc, and their effects on the expressiveness of the resulting audio codebooks. We show that a codebook learned by random sampling can perform almost as well as one generated through a more expensive k-means clustering.
- We evaluate compression techniques including Latent Dirichlet Allocation (LDA) and Agglomerative Clustering to reduce the codebook size while maintaining performance.
- We perform detailed fusion experiments to capture the complementary nature of the various models for MED.

2. Dataset and Setup

We report experiments on the development data from the TRECVID MED task. Our dataset comprises of 28 events that have appeared in MED10 (P001-P003), MED11 (E001-E015) and MED12 (E021-E030) evaluations. To evaluate robustness to new event classes and scalability to larger datasets we validate our approaches on three different evaluation sets:

1. MED-9746 Dataset (18 Events, E001-E015, P001-P003) with 3104 videos for training and 6642 for testing.
2. MED-11746 Dataset (28 Events, E001-E015, E021-E030, P001-P003) with 6584 videos for training (event independent detectors) and 7642 for testing.
3. MED-50328 Dataset (25 Events, E006-E030) with 17857 training videos and 32471 testing videos.

Each target event has around 100-200 positive video samples on average. The remaining videos are unrelated to the target events and are provided as "background" videos.

For the evaluation, we use the PMiss@TER=12.5, which is defined as the point at which the ratio between the Probability of Missed Detection and Probability of False Alarm is 12.5:1. We then compute the average PMiss@TER=12.5 as the average across all event classes.

3. Low Level Feature Representations

3.1. Mel-Frequency Cepstral Coefficients (MFCC)

We use MFCC as our fundamental frame level feature. MFCCs have found wide usage in speech recognition and speaker identification. The single-channel soundtrack is first re-sampled to 16kHz. For our experiments, we compute 20 dimensional MFCCs over 32ms windows with a 10ms hop. We also calculate deltas over 5 frame windows resulting in a 40 dimensional feature vector. Our experiments as well as work in [14] show that having 20 coefficients performs better at event classification than using the traditional 13 coefficients used for speech. We also experimented with double deltas and energy coefficients but they provided little or no gains.

3.2. Stacked Cepstral Features

Previous work [15] in speech recognition have shown that capturing temporal information can be helpful. Conventional approaches involve performing regression across successive frames to calculate deltas or applying transforms to series of stacked cepstral vectors. Such an approach appears appealing for tasks like event classification where it would be helpful to capture temporal knowledge in the codewords of the Audio-Codebook. To incorporate temporal information into our framework, we consider stacking N consecutive 40 dimensional MFCC feature vectors with a 10 ms shift resulting in a $40\times$N dimensional representation, where N is the "stacking" parameter. We experimented with different values for N (3, 5, 7) and found that N=5 with 10ms shifts provides the best trade-off between computational complexity for computing codebooks and event classification performance.

3.3. Large Scale Temporal Pooling Features (LSPF)

While using Stacked Cepstral Features helps in capturing temporal aspects at a granular scale(50ms), we would also like to model longer lasting sounds like cheering, engine noise, music etc which usually span over several seconds. One way to achieve this is to compute statistics or pooling functions over longer temporal windows. Previous work have used such features for music classification [16] and acoustic event detection (AED) [17]. But unlike AED which requires manual annotations, we encode them into an Audio-Codebook via unsupervised clustering. In our work we use a wide range of pooling functions including extremes, moments, percentiles, crossings, peaks, regressions, and means over low level descriptors like MFCC, Chroma, PLP, pitch, loudness, spectral flux, etc. These features try to capture the spectral and temporal shape of the waveform over a short window.

We experiment with various window lengths and shift parameters and found that 2sec windows with 100ms shift provides the best results for both MED (via unsupervised clustering) as well as Acoustic Event Detection(via supervised learning) [2]. The features were computed using the openSmile [18] toolkit resulting in a 6700 dimensional feature set. In order to reduce the dimensions for effective clustering, the features were first standardized to have zero mean and unit variance. This was followed by a feature selection step using Information Gain Criteria based on the labeled data from [2], reducing the dimensions to 5500. To remove correlated features, PCA (Principal Component Analysis) Whitening was performed and top 300 components were considered to form the final representation.

Model	AvgPMiss@TER=12.5
LSPF 4s 100ms 8000 Codebook	0.596
LSPF 2s 100ms 8000 Codebook	0.595
LSPF 1s 100ms 8000 Codebook	0.592
LSPF 0.5s 100ms 8000 Codebook	0.582

Table 1: Performance of Large Scale Pooling Features (LSPF) for different temporal windows on the MED-9746 dataset. In general we observe that smaller windows work better for MED while larger windows perform better on AED. Evaluation was done using the procedure defined in Section 5.1.

4. Audio-Codebook Generation

We use the k-means clustering method for generating the audio codebooks from the low level continuous feature representations defined above. K-means is an unsupervised clustering algorithm that tries to minimize the variance between the k clusters and the training data. In the codebook generation step, we first randomly sample points from the videos in the training set

and then run k-means clustering. The centroids of the resulting clusters form our codebook. Each video is then represented as a histogram of codeword counts by encoding low level feature vectors using the trained codebooks. Thus we obtain a fixed size representation for a variable sized video.

To account for the length of different audio documents we perform L1 normalization on the histogram features before performing classification. In [5], unnormalized histograms were shown to perform better than L1 normalized representations owing to the correlations between event classes and video length. However in order to ensure our models are free from such class biases especially as we scale to bigger datasets and more event classes, we use L1 normalized histograms.

There are several other design decisions that influence the discriminating power of the resulting codebook including the codebook size, the sampling process, and the encoding process.

4.1. Sampling Method

Since it is not feasible to run clustering on all the frames extracted from the audio-signal, one popular approach is to perform random or uniform sampling where each frame in the audio signal has equal probability of getting selected. In [13] the authors show that this leads to an unbalanced Zipf's like distribution in the feature space. Since k-means is a variance based algorithm, given such a distribution, it will assign more and more clusters to the denser areas of the feature space while failing to represent more informative regions. In Table 2 we show that codebooks learned this way are not any more informative as those constructed using random selection. This is because features that occur frequently usually represent sounds that are present in almost every audio, and hence are too generic to provide any discrimination among classes. These can be termed as "Acoustic Stop Words" by drawing relation with stop words in text. Hence over sampling data-points from this space adversely affects the expressive power of the codebook.

Method	AvgPMiss@TER=12.5
16000 Hard Encoded Codebook	0.582
26000 Random Sampled Codebook	0.580

Table 2: Randomly Sampled codebook performs similar to a Hard Encoded codebook learned using K-means [MED-9746]

4.2. Codebook Size

Like the Bag-of-Visual-Words model, the discriminating power of an audio codebook model is governed by the codebook size. The codebook size is determined by the number of clusters K generated by the k-means clustering. There are various trade-offs to be considered for choosing the right number of clusters. One is a trade-off between efficiency and performance. Larger codebooks perform better but at a significantly higher computational cost. Furthermore, the choice also varies with the type of corpus and the low level feature representations. Higher the feature dimensions, larger the size of the codebook that results in the best performance. For feature representations defined in Section 3 we experimentally find that K=4096 for Section 3.1, K=8000 for Section 3.2 and Section 3.3 result in the most expressive codebooks for the MED task.

Figure 1 shows the relationship between performance and size of the audio codebook. We see consistent improvements as the codebook size increases from 100 to 4000 with the improvements becoming less prominent as we reach 16000. This could be attributed to the reasons described in the previous Section 4.1, as k-means will assign major fraction of the codewords to the denser regions while leaving only a few to represent the more informative but less dense regions. Contrary to the findings in [5] where a 1000 codebook provided the best performance, we see improvements even with codebook sizes as high as 16000, suggesting the need for larger codebooks as the event classes increase.

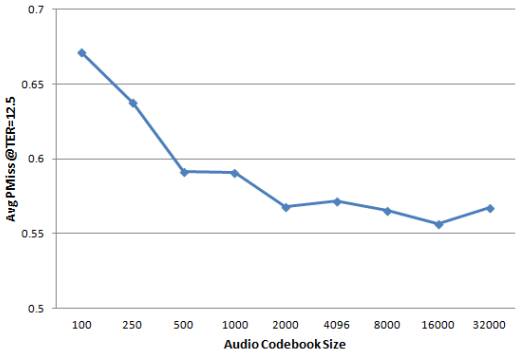

Figure 1: Performance vs Codebook size [MED-9746]

4.3. Soft vs Hard Encoding

Once the codebook has been generated, each video could then be represented in terms of frequency distribution of the codewords. This is usually done in two ways: *hard encoding*, where each continuous feature descriptor is assigned only to the closest cluster centroid. Previous work in computer vision [10] have shown that hard encoding fails to capture the inherent uncertainty of the feature descriptor and a more expressive vocabulary can be achieved by assigning a degree of similarity between feature descriptors and the codewords. This is termed as *soft encoding*. Several methods have been employed to perform soft encoding. In our implementation, we first rank the N closest codewords in the increasing order of their euclidean distance from the feature descriptor and then weigh them by the inverse of their ranks. From results in Table 3 we can see that soft encoding outperforms hard encoding. Furthermore as the size of the codebook grows the improvement of soft over hard encoding becomes more prominent, as with more cluster centroids we have more variability in point assignment.

CodebookSize	Hard Encoding	Soft Encoding
4096 MFCC	0.578	0.571
16000 MFCC	0.582	0.556

Table 3: AvgPMiss@TER=12.5 for Hard and Soft Encoding on MED-9746 dataset using MFCC features.

5. Experiments

5.1. Event Classification and Fusion

For performing event classification, we use Kernel Support Vector Machines (SVM) owing to their ability in capturing nonlinear decision boundaries. We report experiments using two different kernels, Radial Basis Function (RBF) and Chi-Squared. While RBF is a good initial kernel for modeling arbitrary input

space, Chi-Squared kernels usually outperform them for histogram representations. We also report results from the GMM super-vector (GSV-SVM) system on MED task to act as our baseline. We refer reader to [4] for a detailed summary of the system. For feature representation we use the L1 normalized histograms for the Bag-of-Audio-Words model with Chi-Squared Kernel and the super-vector representation for the GSV-SVM with RBF kernel. We employ the one-against-all paradigm for classifier learning by training a binary classifier (in-event or out-of-event) for each of the 28 event classes. In order to combine our different sub-systems, we perform Weighted Average Fusion with uniform weights.

$$Score_{fusion} = \sum_{i=1}^{N} \frac{Score_{Model_i}}{N} \quad (1)$$

5.2. Codebook Compression

It is often desirable to have a compact codebook with higher performance. In order to reduce the dimensions of the learned codebook without impacting performance, we experimented with two approaches.

Similar to the work in [19], we use Latent Dirichlet Allocation (LDA) [20], a probabilistic generative model to discover latent acoustic topics from the codewords. Using hard encoding each video is first represented as a stream of codewords from the audio codebook learned in Section 4. Then a set of topics is learned using the LDA implementation in MALLET [21] , which uses variational inference to approximate the posterior distributions. The learned topic distributions for each video are then used as the representation for classification. As shown in Table 4, though the approach does not improve performance over the corresponding hard encoded codebook, the representation significantly reduces the dimensions from 16000 to 600.

Method	AvgPMiss@TER=12.5
16000 Hard Encoding	0.582
LDA 600 Topics	0.585

Table 4: LDA with 600 Topics performs comparably to 16000 hard encoded codebook on the MED-9746 dataset

Second approach involved performing Agglomerative Clustering with single linkage over the codewords from the 16000 audio codebook, to reduce it to 4096 dimensions. The intuition behind this approach is based on our previous discussion in Section 4.1 on how more clusters are assigned to denser regions in a k-means approach. Hence, in a 16000 codebook, a major fraction of the codewords is concentrated around the dense regions providing redundant information, which can be removed using an agglomerative clustering approach. Table 5 compares the performance of a 16000 audio codebook with the performance of the same codebook after compression using the above approach. Both systems use soft encoding for representation and are evaluated on the MED-11746 dataset.

Method	AvgPMiss@TER=12.5
MFCC-16000-Codebook	0.565
MFCC-16000-Compressed	0.562

Table 5: Codebook compression with Agglomerative Clustering

Method	MED-11746	MED-50328
MFCC-4096	0.571	0.544
GSV-SVM	0.577	0.562
MFCC-16000-Compressed	0.562	0.540
MFCC-Stack5-8000	0.554	0.522
LSPF-8000-2sw	0.595	0.580
LSPF-8000-0.5sw	0.582	0.577
FUSION-ALL	**0.515**	**0.482**

Table 6: Avg PMiss@TER=12.5 for our models

5.3. Results

Table 6 compares the performance for our different models over two datasets MED-11746 and MED-50328. It shows that our baseline systems, 4096 MFCC audio codebook (MFCC-4096) and GSV-SVM provide comparable performance. Furthermore, by training a bigger 16000 audio codebook followed by codebook compression using agglomerative clustering (MFCC-16000-Compressed) as defined in Section 5.2, we achieve improvements in performance without sacrificing compactness. Stacked Cepstral Feature codebook of size 8000 (MFCC-Stack5-8000) remains to be the single best performing system. This suggests that by capturing temporal characteristics at the feature representation level results in a more expressive codebook. LSPF-8000-2sw and LSPF-8000-0.5sw depict the performance of Large Scale Temporal Pooling Features (LSPF), extracted every 100ms over 2sec and 0.5 sec windows respectively. A codebook size of 8000 was used for representation. Though on their own they perform worse than our baseline systems, they provide a 2 percent absolute improvement in performance after performing fusion. FUSION-ALL represents our best overall system after performing late fusion on the above systems. We achieve Average PMiss@TER=12.5 of 0.482 on the MED-50328 Development Set, which is a 11.4% relative improvement over our baseline 4096-Codebook Model and a 14.3% relative improvement over the GMM supervector model.

6. Conclusion

In this work, we have presented our audio-only system for the Multimedia Event Detection (MED) task. In particular, we have shown our analysis of the popular Audio-Codebook model on the MED development dataset, and have performed extensive experiments to show how various design decisions can influence the performance of such a model. We also presented audio representations that could help incorporate temporal information into the codebook model at the feature representation level, leading to improved performance on the MED Task.

7. Acknowledgements

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

8. References

[1] "Trecvid multimedia event detection, evaluation 2012," 2012, http://www.nist.gov/itl/iad/mig/med12.cfm.

[2] S. Burger, Q. Jin, P. Schulam, and F. Metze, "Manual annotation of environmental noise in audio streams," Carnegie Mellon University, Language Technologies Institute, Technical Report CMU-LTI-12-017, Pittsburgh, 2012.

[3] L. Cao, S. Chang, N. Codella, C. Cotton, D. Ellis, L. Gong, M. Hill, G. Hua, J. Kender, M. Merler *et al.*, "Ibm research and columbia university trecvid-2011 multimedia event detection (med) system," *TRECVID Multimedia Event Detection Task (MED)*, 2011.

[4] Q. Jin, P. Schulam, S. Rawat, S. Burger, D. Ding, and F. Metze, "Event-based video retrieval using audio," in *Proc. of Interspeech*, 2012.

[5] S. Pancoasta and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Proc. of Interspeech*, 2012.

[6] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[7] X. Zhuang, S. Tsakalidis, S. Wu, P. Natarajan, R. Prasad, and P. Natarajan, "Compact audio representation for event detection in consumer media," in *Proc. of Interspeech*, 2012.

[8] B. Byun, I. Kim, S. M. Siniscalchi, and C.-H. Lee, "Consumer-level multimedia event detection through unsupervised audio signal modeling," in *Proc. of Interspeech*, 2012.

[9] S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification," in *Proc. of Interspeech*, 2011.

[10] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 7, pp. 1271–1283, 2010.

[11] J. van Gemert, C. Snoek, C. Veenman, A. Smeulders, and J. Geusebroek, "Comparing compact codebooks for visual categorization," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 450–462, 2010.

[12] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, pp. 197–206.

[13] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 604–610.

[14] K. Lee and D. Ellis, "Audio-based semantic concept classification for consumer video," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1406–1416, 2010.

[15] B. Milner, "Inclusion of temporal information into features for speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 1. IEEE, 1996, pp. 256–259.

[16] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, "Temporal pooling and multiscale learning for automatic annotation and ranking of music audio," in *In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR11)*, 2011.

[17] Z. Zhang and B. Schuller, "Semi-supervised learning helps in sound event classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 333–336.

[18] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 1459–1462.

[19] S. Kim, S. Sundaram, P. Georgiou, and S. Narayanan, "An n-gram model for unstructured audio signals toward information retrieval," in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 477–480.

[20] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[21] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.