# HOW TO PUT IT INTO WORDS – USING RANDOM FORESTS TO EXTRACT SYMBOL LEVEL DESCRIPTIONS FROM AUDIO CONTENT FOR CONCEPT DETECTION

*Po-Sen Huang*[†], *Robert Mertens*[‡], *Ajay Divakaran*[*], *Gerald Friedland*[‡], *Mark Hasegawa-Johnson*[†]

[†]Beckman Institute, ECE Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[‡]International Computer Science Institute, 1947 Center Street, Suite 600 Berkeley, CA 94704, USA
[*]SRI International Sarnoff, 201 Washington Road, Princeton, NJ 08540, USA
{huang146, jhasegaw}@illinois.edu,{rmertens, fractor}@icsi.berkeley.edu,ajay.divakaran@sri.com

## ABSTRACT

This paper presents a system that uses symbolic representations of audio concepts as words for the descriptions of audio tracks, that enable it to go beyond the state of the art, which is audio event classification of a small number of audio classes in constrained settings, to large-scale classification in the wild. These audio words might be less meaningful for an annotator but they are descriptive for computer algorithms. We devise a random-forest vocabulary learning method with an audio word weighting scheme based on TF-IDF and TD-IDD, so as to combine the computational simplicity and accurate multi-class classification of the random forest with the data-driven discriminative power of the TF-IDF/TD-IDD methods. The proposed random forest clustering with text-retrieval methods significantly outperforms two state-of-the-art methods on the dry-run set and the full set of the TRECVID MED 2010 dataset.

***Index Terms***— Multimedia Event Detection, Audio Classification, Random Forests, Term Frequency, Inverse Document Frequency

## 1. INTRODUCTION

The amount of electronically available multimedia content increases on a daily basis. On YouTube alone, the amount of video content grows by 48 hours of video every minute[1]. While this huge amount of data makes different kinds of content available, it makes it harder and harder to find specific video or audio documents due to the sheer mass of data that has to be searched. Effective indexing and classification techniques can be used to tackle this problem and hence comprise a currently very active research area. One research question in this field is multimedia concept detection, which addresses the problem of finding videos that fit a given concept such as *Batting a run* or *Making a cake* in the TRECVID 2010 MED (Multimedia Event Detection) challenge[2]. The general approach for multimedia event detection is to learn properties of the concept to be detected by automatically training a classifier on a set of representative videos and then applying this classifier to another set of videos.

With visual methods being the most prominent approach [1], more semantic approaches like OCR (optical character recognition) and ASR (automatic speech recognition) have also been used in conjunction with vision based approaches [2]. Recently, non-ASR audio based approaches have entered the scene more widely. While most approaches focus on detecting the presence of sub-concept sounds which describe certain kind of activities or environment such as "outdoor rural" [3], audio-holistic approaches that use a video's total audio information for event detection have also proven to be effective [4]. Both classes of approaches come with their own advantages and disadvantages. Sub-concept-based approaches often rely on human annotation (which is highly subjective) and use only a fraction of the information available in training video. Results obtained with these approaches are, however, easily explainable. Holistic approaches use all of the audio information available in a video and do not require any annotation of the audio stream. The main disadvantage of audio-holistic approaches is that they disregard many overlapping pieces of information.

The approach described in this paper combines the strengths of both holistic and sub-concept-based approaches by dividing the audio-holistic approach into three automatic steps: automatic audio word learning by random forests, audio word weighting according to their importance, and event retrieval and classification by support vector machines. The remainder of this paper is organized as follows: Section 2 will

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusion contained herein are those of the authors and should not be interpresented as necessarily representing the official policies or endorsement, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government. The authors thank Dr. Qian Yu and Dr. Saad Ali for their suggestions on random forests, Dr. Yu-Gang Jiang [3] and Sourish Chaudhuri [7] for detailed discussions on their experiments.

[1]http://youtube-global.blogspot.com/

[2]http://trecvid.nist.gov/

briefly describe related work. Section 3 gives an overview of the approach. The evaluation setups, along with experimental results on the TRECVID 2010 MED dataset, are described in Section 4. Section 5 concludes the paper.

## 2. RELATED WORK

While there is a wealth of literature available on the use of visual features for concept detection and video classification (see [1] for an overview) this paper deals with audio-based detection and classification methods. Of course, in a real-life application, audio- and video-based methods can and should be used in conjunction where available. Even back in the last millennium, multimodal analysis was used for the semantic characterization of video content [5]. A number of audio based approaches have been employed for video scene segmentation or video classification. These approaches can be separated into supervised techniques that are trained for the detection of specific sound categories like [3] or unsupervised techniques that extract sound categories based on audio mining techniques like [6, 7].

The two approaches that bear most similarities to the approach introduced in this paper are the ones described in [6] and in [7]. Lu and Hanjalic proposed an iterative spectral clustering method to decompose an audio data stream into audio elements, and proposed the TF-IDF and TD-IDD approach for audio element discovery [6]. In contrast to our approach, the approach from Lu et al. [6] does, however, use a spectral clustering method, which is a clustering method within each audio document. Our approach, based on random-forest clustering, learns discriminative audio clusters from all training audio documents efficiently. The approach from Chaudhuri et al. [7] uses audio segmentation and describes segments with Hidden Markov Models. It does, however, not use a subsequent processing stage, such as our methods with text-inspired techniques, that enhances the discrimination.

## 3. SYSTEM OVERVIEW

The system works in three major steps shown in Figure 1: First, we extract MFCC features and learn a random forest dictionary using training data. We can view each leaf node in the random forest as an audio word. Second, given a data sample, we can use its leaf node IDs ($N$ audio words) to represent the data. Third, the audio words are weighted according to their term frequency (TF), inverse document frequency (IDF), term duration (TD), and inverse document duration (IDD). We then use the weighted histogram as a feature vector to represent each audio clip and then use support vector machine for retrieval and classification.

### 3.1. Random Forest Vocabulary

Random forests have been used for classification and regression tasks [8]. A typical random forest consists of a set of binary decision trees. During training stage, each non-leaf node in each tree is assigned a binary test that is applicable to any data sample. Based on the result of the test, a sample can go to one of the two children of a given non-leaf node. In this way, a sample can be passed through each of the trees, starting from its root and ending up in one of its leaves.

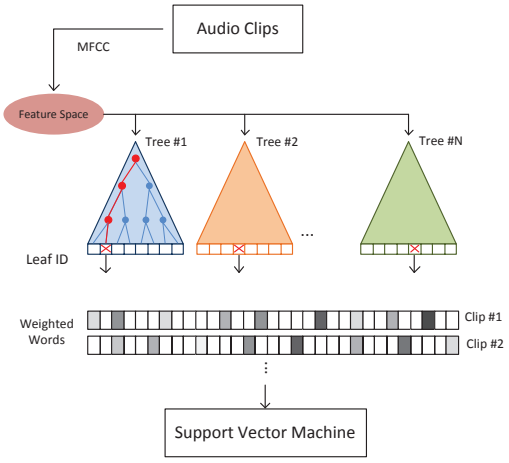


**Fig. 1**. System Overview

In a random forest, all the trees are trained with the same parameters, but on the different training sets. For each training set, we randomly select the same number of data from the original set and the data can be chosen more than once and can be absent. At each node of each tree, only a random subset of variables ($\sqrt{\text{feature dimension}}$) is used to find best split. It has been shown that combining together several trees trained in a randomized way can achieve better generalization and stability compared with a single deterministic decision tree [8].

In addition to classification or regression [8], random forests have been shown to provide a flexible data-driven framework that can provide good multi-class classification and is hence likely to be useful for clustering [9]. In a forest, samples end up in the same leaf node, if the subsets of their feature dimensions are close. Hence, if two data samples are close (similar feature vectors), then they should fall into close leaf nodes in a random forest. In this paper, based on this idea, our random forest clustering algorithm is as follows: Suppose we label all the leaf nodes and use their indices as clustering IDs. Given a data sample, it will end up at a leaf node of each tree. We can use their clustering IDs as **audio words** to represent the data sample.

### 3.2. Weighted Audio Words Approach

By random forest clustering, each data sample can be described using multiple audio words. In the semantic multimedia events, some sounds commonly happen across all events such as silence sounds, but some of them happen uniquely within certain events. Inspired by the *term frequency* and *in-*

*verse document frequency* in text analysis [10], Lu and Hanjalic proposed audio keyword discovery using TF-IDF framework [6] to find key audio segments. Here, we further expand this idea to random forest audio words for event retrieval and classification.

By the analogy to text analysis, we can think of each clustered ID as an *audio word* and each audio clip as an *audio document*.

*Term frequency* (TF) is defined as follows:

$$\text{TF}(c_i, D_k) = \frac{\sum_j n_j P(c_i = c_j | c_j \in D_k)}{\sum_j n_j} \quad (1)$$

where $n_j$ is the occurrence number of audio term $c_j$ in the audio document $D_k$. $P(c_i = c_j | c_k \in D_k)$ is the probability that audio term $c_i$ equals $c_j$ in document $D_k$. Note that in our experiments, this probability is a delta function: audio terms are deterministically labeled with the term label that maximizes the output score of the random forest.

Similarly, we can define *term duration* (TD) as

$$\text{TD}(c_i, D_k) = \frac{\sum_j d_j P(c_i = c_j | c_j \in D_k)}{\sum_j d_j} \quad (2)$$

where $d_j$ is the duration of audio term $c_j$ in the document $D_k$.

Similar to *inverse document frequency* (IDF) in text document analysis, IDF of an audio term can be defined as the log of the number of all documents divided by the number of documents containing the audio element.

$$\text{IDF}(c_i) = \log \frac{|D|}{\sum_k P(c_i \in D_k)} \quad (3)$$

where $|D|$ means the total number of documents and $P(c_i \in D_k)$ is the probability of term $c_i$ in document $D_k$.

Similarity, *inverse document duration* (IDD) is defined as the log of the duration of all documents divided by the duration of audio term $c_i$ in all documents.

$$\text{IDD}(c_i) = \log \frac{\sum_k d_{D_k}}{\sum_k \text{TD}(c_i, D_k)} \quad (4)$$

where $d_{D_k}$ is the total duration of audio document $D_k$.

Some text analysis applications have benefited from the use of logTF and logTD; therefore these features are also considered in our experiments [10].

Finally, for each audio document, we weight each audio element according to the indicators mentioned above, assuming indicators are independent of each other. For each audio clip $D_k$, we can represent it by the feature vector

$$\text{TF}(D_k) = [\text{TF}(c_1, D_k), \ldots, \text{TF}(c_M, D_k)], \quad (5)$$

and similarly for the feature vectors $\log \text{TF}(D_k)$, $\text{TFIDF}(D_k)$, $\log \text{TFIDF}(D_k)$, and $\text{TFIDFTDIDD}(D_k)$, where the combination terms are the product of weights of individual indicators, and $M$ is the total number of words.

By incorporating event label information, we use a support vector machine with intersection kernel for the multiclass event retrieval/classification problem.

## 4. EVALUATION

We use the TRECVID 2010 Multimedia Event Detection dataset (MED) from NIST MED10 evaluation task. The MED data consist of 1746 clips of training data, totaling 56 hours in length, and 1724 clips of test data, totaling 59 hours. The recordings are multimedia content uploaded by public users. Each video clip is labeled as one of 4 concepts: *batting in a run*, *making a cake*, *assembling shelter*, and *other*. The class *other* consists of all videos that do not belong to the first 3 classes. Participants in the NIST MED evaluation were required to retrieve recordings from the test set. In this paper, we only use audio in the recordings. Our framework can potentially be combined with video. Moreover, we do not use any annotations besides class labels. Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from the video soundtrack. We use a frame period of 10 ms with an analysis window of 25 ms in the feature extraction. To balance the computational complexity and the detection resolution, a sliding window of 1s with 0.5s overlap is used to segment the frame sequence. At each window position, the mean and variance of the frame level features are computed and used to represent the corresponding one-second-long audio segment [6].

To evaluate our performance, we compare our results with the MED10 winning team [3], that has achieved the best reported results, and a recent *audio unit descriptors* (AUDs)-Hidden Markov models (HMM) approach [7].

### 4.1. Dry-Run Evaluation

Jiang et al. used a bag-of-MFCC approach with vocabulary of 4000 words learned by the k-means algorithm. Classification/retrieval is performed using a soft-weighted 4-nearest neighbor projection into the 4000 learned centers, followed by SVM scoring using a $\chi^2$ kernel [11].

Jiang et al. reported their results in the dry-run validation set in terms of average precision (AP) – MAP is the average of AP, averaged over the test database [3]. The dry-run validation set consists of 473 clips from the MED training set and the rest of MED training set is used for development. In Table 1, we report our results in terms of MAP. We empirically choose a vocabulary size of around 4000 words by controlling the number of trees and the maximum depth of each tree in the random forest. From the experimental results, we can observe that (1) using logarithms for TF and TD helps in boosting the performance. (2) SVM intersection kernel can slightly improve the results compared with $\chi^2$ kernel. Our results show large differences in AP across the four classes, paralleling the results of [3] in this respect, e.g., in our RF_l8_n25 logTFIDF classifier, the average precision is 0.217, 0.810, 0.426 for the events *Assembling shelter*, *Batting in run*, *Making a cake* respectively. The *Batting in run* event have more audio structure, but *Assembling shelter* and *Making a cake* events vary widely.

**Table 1**. Mean average precision on the dry-run evaluation set. The random forest vocabulary RF_li_nj (X) represents a random forest with j trees, the maximum tree depth i, and vocabulary size X.

| Vocabulary Type (Size) | Mean Average Precision | | | | |
|---|---|---|---|---|---|
| | $\chi^2$ kernel | | | | |
| K-means (4000) [3] | 0.404 | | | | |
| Weighting | TF | logTF | TFIDF | logTFIDF | TFIDFTDIDD |
| RF_l8_n20 (3778) | 0.446 | 0.452 | 0.451 | 0.452 | 0.414 |
| RF_l8_n25 (4669) | 0.455 | 0.457 | 0.474 | 0.475 | 0.430 |
| RF_l8_n30 (5619) | 0.444 | 0.449 | 0.472 | 0.473 | 0.421 |
| | intersection kernel | | | | |
| Weighting | TF | logTF | TFIDF | logTFIDF | TFIDFTDIDD |
| RF_l8_n20 (3778) | 0.445 | 0.451 | 0.462 | 0.467 | 0.406 |
| RF_l8_n25 (4669) | 0.456 | 0.462 | 0.480 | **0.484** | 0.425 |
| RF_l8_n30 (5619) | 0.452 | 0.455 | 0.481 | **0.484** | 0.425 |

### 4.2. Full Set Evaluation

Chaudhuri et al. proposed AUD-HMM and reported their results using MED10 data in terms of classification accuracy [7]. They learn language models over sequences of acoustic units. They used the whole MED10 training set for development and the whole testing set for evaluation. Since the majority clips of training and testing set belong to the *other* class (1580 of 1746 clips in the training set and 1559 of 1724 clips in the testing set), they reported their results for both the 3-class and 4-class classification tasks (ignoring *other* vs. including *other* class). In Table 2, we report our results in terms of classification accuracy using SVM intersection kernel [3].

While the results from Chaudhuri et al. are worse than the majority guess for the *other* class, our results consistently outperform the majority guess in the 4 class case. The *64 symbols 2-gram* achieves the best performance in the 3 class case, but its performance deteriorates in the 4-class case because of missed *other* class detections.

**Table 2**. Average classification accuracy on the full evaluation set. RF_li_nj (X) represents a random forest with j trees, the maximum tree depth i, and vocabulary size X. The "Majority Guess" chooses the majority class as predicted results.

| System | Weighting | 3-class | 4-class |
|---|---|---|---|
| RF_l7_n20 (2256) | TF | 70.70 | 92.01 |
| | logTF | 70.06 | 92.01 |
| | TFIDF | 73.89 | 91.46 |
| | logTFIDF | **75.16** | 91.46 |
| | TFIDFTDIDD | 66.88 | 91.95 |
| RF_l8_n15 (3202) | TF | 71.97 | 92.07 |
| | logTF | 72.61 | 92.13 |
| | TFIDF | 70.70 | 91.70 |
| | logTFIDF | 72.61 | 91.70 |
| | TFIDFTDIDD | 71.34 | 91.58 |
| RF_l8_n20 (4304) | TF | 70.70 | 92.13 |
| | logTF | 71.97 | 92.13 |
| | TFIDF | 70.70 | 91.70 |
| | logTFIDF | 71.97 | 91.82 |
| | TFIDFTDIDD | 68.79 | 91.89 |
| Majority Guess | | 35.15 | 90.43 |
| 64 symbols 2-gram [7] | | 81.61 | 73.61 |
| 200 symbols 3-gram [7] | | 55.63 | 77.08 |

[3]Note that MAP is a better metric to discriminate between different methods in the case of *other* class clips, of which there are many, as MAP also considers the ranking of classification confidence.

## 5. CONCLUSION

This paper has introduced an approach for audio-based concept detection that leverages the power of established methods from text retrieval. In order to make audio features applicable to text retrieval methods, we have employed random forests as an abstraction step to cluster audio features across videos. These more abstract representations can be used as words in text retrieval methods.

Experiments show that our proposed approach improves the performance significantly over two state-of-the-art methods on the dry-run set and the full set of the TRECVID MED 2010 dataset. For future work, we will try to add information about the temporal context to the clustering algorithm and to compare sound segments identified through clustering by human subjects in perceptual studies.

## 6. REFERENCES


[1] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retr.*, vol. 2, pp. 215–322, April 2009.

[2] H.D. Wactlar, T. Kanade, M.A. Smith, and S.M. Stevens, "Intelligent access to digital video: Informedia project," *Computer*, vol. 29, no. 5, pp. 46–52, 1996.

[3] Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang, "Columbia-UCF TRECVID2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in *NIST TRECVID Workshop*, Gaithersburg, MD, November 2010.

[4] R. Mertens, H. Lei, L. Gottlieb, G. Friedland, and A. Divakaran, "Acoustic super models for large scale video event detection," in *Proceedings of the 2011 joint ACM workshop on Modeling and representing events*. 2011, pp. 19–24, ACM.

[5] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E.K. Wong, "Integration of multimodal features for video scene classification based on HMM," in *Multimedia Signal Processing, IEEE 3rd Workshop on*, 1999, pp. 53–58.

[6] L. Lu and A. Hanjalic, "Audio keywords discovery for text-like audio content analysis and retrieval," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 74–85, Jan. 2008.

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

[8] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[9] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, 2006.

[10] Edda Leopold and Jörg Kindermann, "Text categorization with support vector machines. how to represent texts in input space?," *Mach. Learn.*, vol. 46, pp. 423–444, March 2002.

[11] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. of CIVR '07*, Amsterdam, NL, 2007, pp. 494–501.