Training audio events detectors with a sound effects corpus

Isabel Trancoso[1]*, José Portêlo*[1]*, Miguel Bugalho*[1]*, João Neto*[1]*, António Serralheiro*[2]

[1]INESC-ID Lisboa/IST, Portugal
[2] INESC-ID/Military Academy
Isabel.Trancoso@inesc-id.pt

Abstract

This paper describes the work done in the framework of the VIDIVIDEO European project in terms of audio event detection. Our first experiments concerned the detection of non-voice sounds, such as birds, machines, traffic, water and steps. Given the unavailability of a corpus labelled in terms of audio events, we used a relatively small sound effect corpus for training. Our initial experiments with one-against-all SVM classifiers for these 5 classes showed us the feasibility of using this type of data for training, thus avoiding the extremely morose task of manual labelling of a very high number of audio events. Preliminary integration experiments are quite promising.
Index Terms: audio segmentation, event detection

1. Introduction

This work was done in the framework of the European project VIDIVIDEO whose goal is to boost the performance of video search engines by forming a 1000 element thesaurus, detecting instances of audio, video or mixed-media content. The project approach is to apply machine learning techniques to train many, possibly weaker detectors, describing different aspects of the audio-video content instead of modeling a few of them carefully. The combination of many single class detectors will render a much richer basis for the semantics. The integration of cues derived from the audio signal is essential for many types of search concepts. Our role in the project is to contribute towards this integration with three different modules: audio segmentation, speech recognition, and detection of audio events. This paper concerns the last module.

Audio events detection (AED) is a relatively new research area with ambitious goals. Typical AED frameworks are composed of at least two parts: feature extraction and audio event inference. Optionally, there may be an intermediate stage of key audio effect detection. The feature extraction process deals with different type of features, such as: total spectral power, subband power, brightness, bandwidth, pitch frequency, MFCC (Mel-frequency cepstral coefficients), PLP (Perceptual Linear Prediction), ZCR (Zero Crossing Rate), etc. Brightness and bandwidth are, respectively, the first and second order statistics of the spectrogram, and they roughly measure the timbre quality of the sound. Many of these features are common to the audio segmentation and speech recognition modules.

The key audio effect detection phase (not always present) detects a discrete set of predefined audio effects usually model by HMMs (Hidden Markov Models). This phase is typically used to simplify the inference phase and, at the same time, explore the HMMs capabilities for time modelling, to correctly model the feature variations in time. Moreover, the HMMs also model interconnections between key audio effects (e.g. an explosion being preceded by a car crash).

In the inference process, various machine learning methods are used to provide a final classification of the audio event:

- Rule-Based Approaches (RB) [7]
  These approaches are normally adopted after retrieving key audio events using HMMs. Using these key audio events, a simple set of rules can be build to identify the final set of audio events (e.g. a whistle in a soccer game corresponds to a fault).
- Gaussian Mixture Models (GMMs) [4], [3], [6]
  In these approaches, each event is considered to have an underlying Gaussian model. The method identifies each Gaussian model and separates it from the rest, thus allowing for event retrieval. Other types of mixture models are also present in the literature, but Gaussian models are the most common.
- Support Vector Machines (SVMs) [4], [5], [6]
  Together with GMMs, these are the most popular approaches. The SVM method maps the input feature values (or key audio events if HMMs are used) to a higher dimensional space, where a hyperplane can be found to separate the inputs. This mapping is done using a kernel function. Several kernel functions are used for audio event detection problems, such as linear [6], polynomial and radial basis function (RBF) [5].
- Bayesian Networks [2]
  A Bayesian network, in an audio event detection problem, is a directed acyclic graph model where each arc defines probabilistic relationships between the semantic concept nodes. This model not only captures the relationships between the audio events, but also has the advantage of being more robust to noise. One of the problems of this approach is that not only we have to train the model but we have also to define the structure of the model. If there is some prior knowledge about the relationships between the events, this knowledge can be incorporated in the model.

This work followed the popular trend of using SVMs. Multiple-Class classification is achieved by combining many one-against-all classifiers. In this approach, a classifier is built for each class that distinguishes between elements of that class or belonging to any other class. The classification can then be done by running each classifier and choosing the class for which the classifier generates the highest value from its decision function. This approach has the advantage that detectors can be trained for a new class without having to retrain the existent detectors.

Before describing our SVM experiments, we shall describe our still very short corpus (section 2). The next section analyses this corpus in terms of speech/non-speech detection (section 3), and the following one (section 4) describes our experiments

with one-against-all classifiers for the classes which were most popular in our corpus. Finally, we discuss how the results produced by these classifiers can be integrated in an application that allows the user to search for semanatic concepts (section 5), and present our conclusions and plans for future work.

2. Corpus

The corpora initially available in the VIDIVIDEO project were targeted at demonstrating the feasibility of the project in three different application scenarios: broadcast news, cultural heritage documentaries, and surveillance. Unfortunately, none of the available files was labelled in terms of audio events. This problem led us into investigating the possibility of training audio events with a corpus of sound effects, which are intrinsically labelled, as each file typically contains a single type of sound. For this purpose, a pilot corpus of 422 short files, totalling 6.8h, was provided in November by B&G, one of the partners of the project. The corpus is structured into 5 broad categories. Table 1 shows a breakdown of the number of files and duration for each category.

Table 1: Constitution of the pilot sound effects corpus.

Category	# Files	Duration (min.)
City	83	68.9
Human	91	70.4
Industry	114	134.3
Office	86	68.9
Rural	48	68.4

For the time being, none of the documentaries or BN shows available in the VIDIVIDEO project is manually labelled in terms of audio events. For the purpose of preliminary testing the one-against-all detectors, we manually labelled 3 cultural heritage documentaries (provided by FRD, another partner in the project), of approximately 1h each, with a reduced number of audio concepts.

3. Analysis of the sound effect corpus

Most of the broadcast news shows and documentaries contain speech. Our intention is to ignore these segments for the purpose of audio events detection and concentrate only on the segments which our speech/non-speech classifier [1] marks as non-speech. Hence, a first experiment tried to verify whether this module detected the presence of speech in our sound effects corpus (down-sampled to 16kHz). As can be seen in Table 2, the results confirmed our expectations of finding speech mostly in the Human category files, with very few false alarms elsewhere.

In our opinion, the false alarms in the City, Industry, and Office classes can be reduced, given the very high frequency contents which point towards non-speech, if higher bandwidth features are included in the speech/non-speech classifier. The false alarm in the Rural class is more acceptable, and harder to exclude.

For the files in the Human class, the high speech detection rate was the expected result. In fact, if the speech/non-speech detector had been trained with the type of material present in these files (laughing, crying, screaming, etc.), instead of just Broadcast News material, the detection rate would likely be even higher. Sounds made with hands and feet (clapping, walking, climbing) were obviously never detected as speech (22 files). Baby voices were not detected as speech. Humming was very often detected as speech. Some sounds were in the borderline: laughing, crying, screaming, coughing. Other sounds were always detected as non-speech: sneezing, sighing, hiccuping, vomiting, burping, snoring, whistling. The detection of cocktail party voices as speech was very dependent on how close some of these voices were to the microphone, which makes them more intelligible. Synchronous voices (but not a chorus) were not detected as speech.

Table 2: Performance of the speech/non-speech classifier in the pilot sound effects corpus. The second column indicates the number of files where speech has been detected.

Category	# Files	Description
City	2	car horn (2s) telephone ringing (32s)
Human	18	laughing, crying, screaming coughing, humming, yawning, talking, etc.
Industry	1	sharpening stone (6s)
Office	1	burglar alarm (34s)
Rural	1	mixture of birds (37s)

The results of this first experiment suggest the use of the speech/non-speech classification module as the first classification stage for audio events detection. This allows the separation of the second stage into two parts: human audio events detection (laughing, crying, screaming, coughing, humming, yawning, etc.), and non-human audio events detection. Given the very few examples in the Human class (produced by very few speakers), we decided to concentrate our first efforts in the second detector. The files classified as containing speech in the Human class were not therefore used in the experiments reported below, not even for "world" models. This means that our target will not be to detect these sounds in the background while someone is speaking, but rather detect them in the absence of speech.

4. One-against-all detectors

With the goal of building simple one-against-all detectors, we have built "class-specific" and "world" models for five initial classes, for which the corpus contained more examples: birds, machines, traffic, walking/climbing, and water. Our experiments were done using the LIBSVM tool [8]. Although other kernels were considered (gaussian and polynomial) the linear kernel was initially chosen for two reasons: speed of training, and good preliminary results.

The first experiments were aimed at testing different feature combinations. Given the reduced number of files for most concepts, no attempt was made at this stage to apply any feature selection technique such as information gain, Chi Square test or principal components analysis. Table 3 shows the results of training SVMs for the five classes, with different combination of features. Generally, best results were obtained with a combination of either PLP or MFCC (12 coefficients + energy + deltas) and 3 additional features: ZCR, brightness and bandwidth. The results are shown in terms of F-measure. Due to the scarcity of files for each class (the total number is shown in the second column), in these initial experiments, the available files were just subdivided into training and development. The "world" model was typically built using between 43 to 46 files,

of which on average 13 were used in the development set.

Table 3: Results (F-measure) of training SVMs for five classes, with different combination of features, for the development set.

Class	#files	PLP	MFCC	PLP+3	MFCC+3
Birds	26	0.33	0.58	0.79	0.74
Machines	22	0.85	0.87	0.82	0.84
Traffic	7	0.01	0.00	0.83	0.80
Walking	31	0.74	0.16	0.55	0.01
Water	9	0.65	0.66	0.66	0.66

Another important parameter that we varied was the length and interval of the analysis windows. The results shown in the table were achieved with analysis windows of 0.5s, updated every 0.25s. These results did not improve using a smaller window size and/or interval, even for classes for which a finer time scale was intuitevely more adequate. For instance, for the walking/climbing class, the F-measure changed from 0.55 to 0.51 (PLP+3 feature set), when using 100ms windows, updated every 50ms.

These initial results involving a reduced number of files were obtained using manually tuned energy thresholds for each file. This threshold is another relevant parameter, given its implications in the number of segments to be processed in each file. Further experiments were done in order to automatically set this threshold. Comparable results were obtained setting this threshold at 10% of the maximum energy level.

The next problem to be addressed was the speed of SVM training. In comparison with other classification data (e.g., medical data), the amount of data needed to be considered for AED is much larger. For this reason, it is important to consider the speed of SVM training. First, we used the LIBSVM scaling tool to scale the parameters to the default LIBSVM scale (-1 to 1). This boosts the SVM training process[8]. In fact, large differences in the scale of parameters may negatively influence the speed of the algorithm convergence.

The LIBSVM tool also allows for the generation of probability estimates. The initial SVM algorithm only generates a classification, the probability estimates are derived from the example distance to the plane. Although we generate this information in our classifiers, we decided not to use the probability estimates option during the feature selection experiments, and only use it in the final classifiers. This decision was made since the implementation of the probability estimates in the LIBSVM tool requires to use five fold cross validation which largely increases the training time.

The high frequency contents of many types of sounds in our pilot sound effect corpus led us into investigating the use of a higher sampling frequency (44.1 kHz) and corresponding high frequency values for the computed features. The number of PLP/MFCC coefficients (including energy) was set at 20, plus corresponding deltas for these exploratory experiments.

Table 4 shows the results obtained for the development test, using several of these parameter settings together: automatically set threshold, higher sampling frequency, scaling, no probability estimates. For some of the classes (machines, traffic, water), the results significantly improve. Although it is very difficult to derive conclusions on the basis of such short training/development sets, the results are consistent with the information gain measure, which showed that for the bird semantic concept the lowest frequencies had more information, hence justifying the small degradation.

Table 4: Results (F-measure) of training SVMs for five classes, with higher frequency features, for the development set.

Class	#files	PLP	MFCC	PLP+3	MFCC+3
Birds	26	0.72	0.55	0.73	0.52
Machines	22	0.88	0.82	0.89	0.86
Traffic	7	0.88	0.87	0.88	0.87
Walking	31	0.62	0.61	0.72	0.78
Water	9	0.77	0.78	0.73	0.74

The following experiments were made by subdividing the files into training, development and test. With such a small amount for training, results were naturally much worse. We tried to compensate this by improving the "world" model, using between 101 to 114 files, subdivided into training (average 82 files), development (average 16 files), and testing (average 16 files, as well). The results for the test set are shown in Table5 in terms of F-measure, for the PLP+3 feature set, with all parameters equal to the ones used for obtaining Table 3. These experiments also indicated that the minimum amount of data for training must be much larger than currently available.

Table 5: Results (F-measure) for the five classes, in the development and test sets.

Class	Dev	Test
Birds	0.56	0.60
Machines	0.57	0.35
Traffic	0.63	0.04
Walking	0.30	0.33
Water	0.23	0.28

5. Integration

The aim of this section is to illustrate how the above detection results can be integrated in an interactive application for searching for different types of semantic concepts in a video. The application was originally designed by the University of Amsterdam, the coordinator of the project, in Windows [9]. The system allows users to import MPEG-7 annotation files in a database system. The database can be queried by means of a GUI, created using OpenGL libraries, that presents the results in a user friendly format. The GUI presents the shots sorted by the confidence of containing the semantic concept.

For the purpose of illustrating the integration into this tool, we took our small set of 3 documentary files which had very few examples of one of our classes (birds). For each of these files, the results of the bird detector were transformed into an MPEG-7 format. This format contains information about which video segments contain the concepts (start and end) and the confidence level derived from the probability estimates generated (each 250ms) by the final SVM classifier.

Testing the detector with the documentary files revealed the need for a postprocessing stage. In fact, if one considered as positive the segments that contained all the consecutive frames that had more than 50% probability estimate, this would generate too many segments of small duration, caused by the probability estimates dropping below the threshold during very few frames where the concept exists, or by very short false alarms. This motivated the use of duration thresholds as well in a postprocessing phase. If the duration of the segment containing

the concept is less than 2s, the segment will be considered as "world". Likewise, if within a segment considered as "class", the concept is not found for a period less than 1.5s, it will also be considered as part of the class. These duration thresholds should be optimized for each class.

After postprocessing, a confidence value is generated for each segment in the MPEG-7 file as the mean value of the probability estimates of the included frames. Table 6 shows the results for the 3 documentary videos that contained the bird concept, before and after postprocessing. Figure 1 illustrates the performance of the postprocessing stage for one of the documentaries. Figure 2 shows the integration of these results in the application demo for the same documentary. The graph show the confidence levels for the top shots containing the bird concept. Except for the first 5 shots (keyframes shown below), all other shots had zero confidence. The 5th shot included birds in the background before the person started to speak.

Table 6: Results (F-measure) for the documentary data set using the bird classifier, before and after postprocessing.

Documentary	Before	After
CastilliDellaLunigiana	0.24	0.63
Populonia	0.39	0.58
Kosovosodo	0.47	0.90

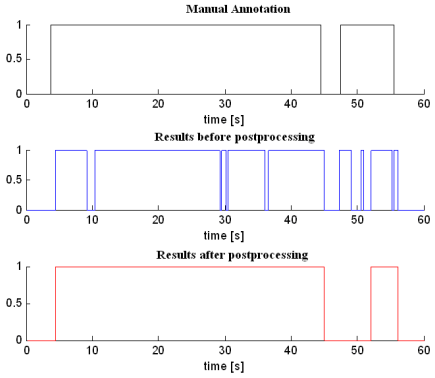

Figure 1: Detection results for the bird classifier in the Kosovosodo documentary.

6. Conclusions and future work

These initial experiments with training one-against-all audio event detectors with a pilot sound effect corpus showed us the feasibility of using this type of data for training, thus avoiding the extremely morose task of manual labelling using a very high number of audio events.

A much larger corpus with a significant number of examples of more than 50 classes is currently being collected. This will allows us to redo our preliminary experiments, in order to full test different feature combinations, explore the possibility of extracting features with a higher sampling frequency, better adequate the length and update interval of the analysis window to the different time structure of the sounds to be detected, evaluate different kernels, etc.

One of the most interesting aspects of this work will be the future integration of audio and video derived cues. For instance,

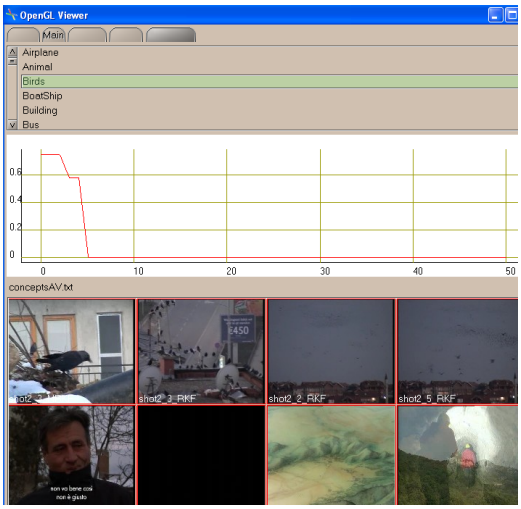

Figure 2: GUI Interface show the application of the bird classifier in the Kosovosodo documentary.

when searching for a dog, the detection of a barking sound with a sufficiently high confidence may reinforce the detection of the image of a dog in the corresponding keyframe, thus contributing towards the detection of the semantic concept.

7. Acknowledgements

The authors would like to thank their colleague Hugo Meinedo, author of the audio segmentation module, and their partners in the VIDIVIDEO project.

8. References

[1] Amaral, R., Meinedo, H., Caseiro, D., Trancoso, I., and Neto J., "A Prototype System for Selective Dissemination of Broadcast News in European Portuguese", EURASIP Journal on Advances in Signal Processing, Hindawi Publishing Corporation, vol. 2007, n. 37507, May 2007.

[2] Cai, R., Lu, L., Hanjalic, A., Zhang, H., and Cai, L., "A flexible framework for key audio events detection and auditory context inference", IEEE Trans. on Speech and Audio Processing, 2005.

[3] Cheng, W., Chu, W. and Wu, J., "Semantic context detection based on hierarchical audio models", Proc. 5th ACM SIGMM Int. Workshop on Multimedia information retrieval, pages 109-115, 2003.

[4] Chu, W., Cheng, J., Wu, J., and Hsu, J., "A study of semantic context detection by using SVM and GMM approaches", Proc. IEEE Int. Conf. on Multimedia and Expo, 2004.

[5] Guo, G. and Li, S., "Content-based audio classification and retrieval by support vector machines", IEEE Trans. on Neural Networks, 14(1):209-215, 2003.

[6] Moncrieff, S., Dorai, C. and Venkatesh, S., "Detecting indexical signs in film audio for scene interpretation", Proc. IEEE Int. Conf. on Multimedia and Expo, 2001.

[7] Xu, M., Maddage, NC., Xu, C., Kankanhalli, M. and Tian, Q., "Creating audio keywords for event detection in soccer video", Proc. IEEE Int. Conf. on Multimedia and Expo, 2003.

[8] Chang, C. and Lin, C., "LIBSVM: a library for support vector machines", Manual, 2001. Online: http://www.csie.ntu.edu.tw/˜cjlin/libsvm

[9] VIDIVIDEO - Periodic Activity Report, Year 1, January 2008.