A Multi-Class Audio Classification Method With Respect To Violent Content In Movies Using Bayesian Networks

Theodoros Giannakopoulos, Aggelos Pikrakis, *Member, IEEE,* and Sergios Theodoridis, *Senior Member, IEEE*
University of Athens,
Department of Informatics and Telecommunications,
Panepistimioupolis, 15784, Athens, Greece
e-mail: {tyiannak, pikrakis, stheodor}@di.uoa.gr

***Abstract*— In this work, we present a multi-class classification algorithm for audio segments recorded from movies, focusing on the detection of violent content, for protecting sensitive social groups (e.g. children). Towards this end, we have used twelve audio features stemming from the nature of the signals under study. In order to classify the audio segments into six classes (three of them violent), Bayesian Networks have been used in combination with the One Versus All classification architecture. The overall system has been trained and tested on a large data set (5000 audio segments), recorded from more than 30 movies of several genres. Experiments showed, that the proposed method can be used as an accurate multi-class classification scheme, but also, as a binary classifier for the problem of violent - non violent audio content.**

I. Introduction

During the last years, a huge increase of video data (but also of all kinds of multimedia content) has occurred. The provided multimedia content is becoming easily accessible by large portions of the population, with limited central control. It is therefore obvious that the need of protection of sensitive social groups (e.g. children) is imperative. In this paper, we present a method for automatic characterization of video content based on the audio information.

The task of detecting violence is difficult, since the definition of violence itself is ambiguous. One of the most widely accepted definition of violence is: ”behavior by persons against persons that intentionally threatens, attempts, or actually inflicts physical harm” ([1]). In video data, most violent scenes are characterized by specific audio signals (e.g. screams and gunshots). The literature related to violence detection is limited and, in most of the cases, it examines only visual features ([2], [3]). In [4] the audio signal is used as additional information to visual data. In particular, a single audio feature, namely the energy entropy, is used in order to detect abrupt changes in the audio signal, which, in general, may characterize violent sounds. Though, the usage of energy entropy as a feature for violent detection can only be used in combination with other audio or visual features, since it only detects abrupt changes and it could therefore lead to the classification of a non violent impulsive noise (e.g. a door closing) as violent. A more detailed examination of the audio features for discriminating between violent and non-violent sounds was presented in [5]. In particular, eight audio features, both from the time and frequency domain, have been used, while the binary classification task (violent and non violent) was accomplished via the usage of Support Vector Machines.

In this paper, we have focused on more audio features in order to detect violence in audio signals but also to give a more detailed characterization of the content of those signals. Therefore, facing the problem as a binary classification task (violent/non-violent) would not be adequate. In addition, such a treatment of the problem would be insufficient in terms of classification accuracy. For example the sound of a non-violent impulsive sound (e.g. a thunder or a door closing) is more similar to a gunshot (violent) than to speech (non violent). It is therefore obvious, that the binary approach would lead to the grouping of distinct sounds, which is undesirable. Thus, we treat the problem as a multi-class audio classification problem. In particular, we have defined six classes (3 violent and 3 non-violent), motivated by the nature of the audio signals met in most movies. The non-violent classes are: *Music*, *Speech* and *Others* (non violent sounds not belonging to music nor speech, e.g. wind, water etc). As violent audio classes we have defined: *Shots*, *Fights* (beatings) and *Screams*.

II. Proposed method

For each audio segment, a number of audio features and respective statistics is calculated, leading to a 12-D feature vector. Afterwards, each class is modelled by a separate Bayesian Network (BN) classifier. Each BN is used as an estimator of the probability that the input audio sample belongs to the respective class. At a final step, the maximum BN probability determines the ”winner” class. In the following paragraphs a more detailed description of the adopted methods is presented.

A. Audio Features

At a first step, 12 audio features are extracted for each segment on a short-term basis, i.e. each segment is broken into a sequence of non-overlapping short-term windows (frames), and for each frame a feature value is calculated. This process leads to 12 feature sequences. Afterwards a statistic is calculated for each sequence, leading to a 12-D feature vector for each audio segment. The features and the statistics used are described below.

1) Zero Crossing Rate (1): Zero crossing rate (ZCR) measures the number of time-domain zero crossings, divided by the frame's length ([6]). It is computed using the equation: $ZCR = \frac{1}{N}\sum_{n=1}^{N-1}\frac{|sgn\{x(n)\}-sgn\{x(n-1)\}|}{2}$, where $sgn(.)$ stands for the sign function, i.e., $sgn\{x(n)\} = +1$ if $x(n) \geq 0$ and -1 if $x(n) < 0$. The *average value* of the feature sequence was computed as the final feature value.

2) Spectrogram Feature (2 features): The spectrogram is firstly calculated using a Short-Time Fourier Transform. At a second step, the mean value of the spectrogram for each window is calculated, leading to a single-dimension feature vector. From this feature sequence, two statistics are extracted and used as final features: a) the *standard deviation* and b) the *maximum value* of the sequence.

3) Chroma Vector Features (2 features): The chroma feature vector has been widely used in music detection algorithms ([7]). It is computed by the logarithmic magnitude of the Discrete Fourier Transform F: $v_k = \sum_{n\in S_k}\frac{F_t(n)}{N_k}, k \in 0..11$, where S_k is a subset of the frequency space and N_k is the number of elements in S_k. Each of the bins S_k expresses one of the 12 pitch classes existing in western music, and therefore each of the chroma bands is separated

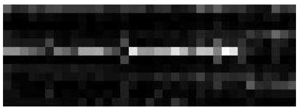

Fig. 1. Music Chroma

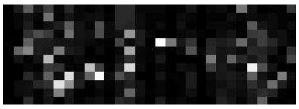

Fig. 2. Speech Chroma

by one semitone. This feature specifically addresses the properties of western musical signals. The chroma vector v_k is computed for each frame i of the audio segment, resulting in a matrix V with elements $V_{k,i}$.

Two features are calculated from the above feature vector:

- **Chroma Feature 1:** The first chroma-based feature is extracted by calculating the deviation between chroma coefficients $k \in 0..11$ in each frame i. For this feature, non-overlapping windows of 100 msecs have been adopted. Furthermore, the *mean value* of the feature sequence was used as the final statistic value.
- **Chroma Feature 2:** The second feature based on the chroma vector is a measure of deviation between successive frames for each chroma element. This stems from the observation that in music segments there is at least one chroma element with low deviation for a short period of time (fig. 2), while in speech segments, the deviation of each chroma element is high (fig. 2). To compute this second chroma-based feature, a sort-term window of 20 msecs has been adopted, while *the minimum deviation* of the chroma coefficients was computed for every 10 frames, i.e. a mid-term window of 200 msecs was used. Finally, the *median value* of the mid-term statistic is computed.

In Figure 3, the histograms of the second chroma-based feature (i.e. the median value of the second chroma feature vector) is presented for three classes: Music, Speech and Shots. It is obvious that for music signals, the value of the feature is generally small.

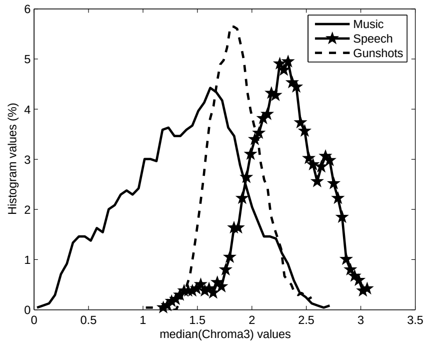

Fig. 3. Histogram of the median value of the Chroma2 feature sequences of ”Music”, ”Speech” and ”Shots” audio segments.

4) Energy Entropy (1): This feature is a measure of abrupt changes in the energy level of an audio signal. It is computed by further dividing each frame into K sub-windows of fixed duration. For each sub-window i, the normalized energy σ^2 is calculated: this is the sub-window's energy, divided by the whole frame's energy. Afterwards, the energy entropy is computed using the equation $H = -\sum_{i=0}^{K-1} \sigma^2 \cdot log_2(\sigma^2)$. K was chosen to be 10, while the adopted statistic for this feature is the *maximum value*.

5) Spectral Rolloff (1): Spectral rolloff ([6]) is defined as the frequency bin $m_c^R(i)$ bellow which $c\%$ of the magnitude distribution of the DFT coefficients is concentrated, i.e. $\sum_{m=0}^{m_c^R(i)} m \cdot |X_i(m)| = \frac{c}{100} \sum_{m=0}^{N-1} m \cdot |X_i(m)|$. This feature is a measure of skewness of the spectral shape. In the current work, we define c to be equal to 80%. Furthermmore, the *median value* was used as a statistic.

6) Mel-frequency Spectral Coefficients (4): The filter bank used for the computation of the MFCCs consists of 40 triangular bandpass filters, with bandwidth and spacing determined by a constant mel-frequency interval. The first 13 filters are linearly-spaced with 133.33Hz between center frequencies and are followed by 27 log-spaced filters, whose filter centers are separated by a factor of 1.0711703 in frequency. The adopted filter bank covers the frequency range $0-8$KHz, suggesting a sampling rate of $16KHz$. In the current work, the first three MFCCs were computed. The *maximum value* and the *maximum to mean* ratio were used as statistics for the first MFCC, the *standard deviation* for the second and the *median value* for the third.

7) Pitch (1): To calculate the pitch contour, the autocorrelation pitch detection method has been adopted. As a statistic, we have used the *zero ratio* (i.e. the percentage of frames with zero pitch) of the pitch sequence. This is a measure of harmonicity of the input audio segment.

TABLE I
WINDOW SIZES AND STATISTICS FOR EACH OF THE ADOPTED FEATURES

	Feature	Statistic	Window (msecs)
1	Spectrogram	σ^2	20
2	Chroma 1	μ	100
3	Chroma 2	$median$	20 (mid term:200)
4	Energy Entropy	max	20
5	MFCC 2	σ^2	20
6	MFCC 1	max	20
7	ZCR	μ	20
8	Sp. RollOff	$median$	20
9	Zero Pitch Ratio	$-$	20
10	MFCC 1	max/μ	20
11	Spectrogram	max	20
12	MFCC 3	$median$	20

B. Classification Method

1) Multiclass Classification Scheme: In order to achieve multiclass classification, the ”One-vs-All” (OVA) classification scheme has been adopted. This simple but very accurate approach for the multiclass classification task ([8]) is based on decomposing the K-class classification problem into K binary sub-problems. In particular, K binary classifiers are used, each one trained to distinguish the samples of a single class from the samples in the remaining classes, i.e. each class is opposed to all the others. For example, for the present audio classification task, one of the single binary classifiers is trained to distinguish a speech signal for non-speech signals. In the current work, we have chosen to use Bayesian Networks (BNs) for building those binary classifiers. As described below, the BNs are used to determine the probability that a sample belongs to one of the classes.

2) Binary Classifiers: In this paragraph, a description of the Binary Classifiers, that compose the OVA architecture, is presented. At a first step, the 12 feature values $v_i, i = 1 \dots 12$ described in Paragraph II-A, are grouped into three 4D separate feature vectors:

$$V^{(1)} = [v_1, v_4, v_7, v_{10}] \quad (1)$$

$$V^{(2)} = [v_2, v_5, v_8, v_{11}] \quad (2)$$

$$V^{(3)} = [v_3, v_6, v_9, v_{12}] \quad (3)$$

This grouping was randomly applied and in future work a more sophisticated combination could be used, taking into account the statistical independence of each feature dimension. Afterwards, for each one of the 6 binary sub-problems, three k-Nearest Neighbor classifiers are trained on the respective feature space. In particular, each kNN classifier KNN_i^j, $i = 1 \dots 6$ and $j = 1 \dots 3$ is trained to distinguish between class i and all i' (not i), given the feature vector $V^{(j)}$. This leads to three binary decisions for each binary classification problem. Thus, a 6x3 matrix R is defined as follow:

$$R_{i,j} = \begin{cases} 1, & \text{if the sample was classified in class} \\ & \quad i\text{, given the feature vector } V^{(j)} \\ 0, & \text{if the sample was classified in class} \\ & \quad \textit{not i}\text{, given the feature vector } V^{(j)} \end{cases} \tag{4}$$

Let us consider, the following result matrix:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

For example, the fact that $R_{1,1} = 1$, indicates that KNN_1^1 (i.e. the KNN classifier of the first binary sub-problem that functions on the feature space of the $V^{(1)}$ feature vectors) decided that the input sample is music. The other two kNN classifiers of the same binary sub-problem decided that the input sample is non-music. The emerging subject here is to decide to *which class the input sample will be classified, according to R.* An obvious approach would be to apply a majority voting rule for each binary sub-problem. Though, in the current work BNs have been adopted: each binary subproblem has been modelled via a BN which combines the individual kNN decisions to produce the final decision, as described in the sequel.

In order to classify the input sample to a specific class, the kNN binary decisions of each subproblem (i.e. the rows of matrix R) are fed as input to a separate BN, which produces a probabilistic measure for each class. BNs are directed acyclic graphs (DAGs) *that encode conditional probabilities* between a set of random variables. In the case of discrete random variables, for each node A, with parents B_1, ..., B_k a conditional probability table (CPT) $P(A|B_1, ..., B_k)$ is defined. In this paper, the BN shown in figure 4, has been used as a scheme for combining the decisions of the kNN individual classifiers. We will refer to this type of BN as the BNC (Bayesian Network Classifier, [9]). Nodes $R_{i,1}$, $R_{i,2}$ and $R_{i,3}$ correspond to the binary decisions of the kNN individual classifiers for the i-th binary sub-problem and are called hypotheses (also rules or attributes) of the BN, while node Y_i is the *output* node and corresponds to the true binary label. Y_i, like the elements of R, is 1 if the input sample really belongs to class i, and it is 0, otherwise.

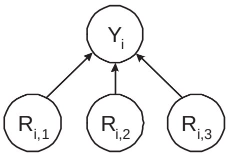

Fig. 4. BNC architecture

In the BN training step, the CPTs of each BN i are learned according to the set ([10]):

$$S^{(i)} = \{(R_{i,1}^{(1)}, R_{i,2}^{(1)}, R_{i,3}^{(1)}, s_{i,1}), \dots, (R_{i,1}^{(m)}, R_{i,2}^{(m)}, R_{i,3}^{(m)}, s_{i,m})\} \tag{6}$$

where m is the total number of training samples, $R_{i,j}^{(k)}$ is the result of j-th kNN classifier ($j = 1, \dots, 3$) for the j-th feature vector of the k-th input sample ($k = 1, \dots, m$), and $s_{i,k}$ is the *true binary label* for the k-th input sample and for the i-th binary subproblem. In other words, $s_{i,k}$ is defined as follow:

$$s_{i,k} = \begin{cases} 1, & \text{if the } k\text{-th sample's true class label is } i \\ 0, & \text{if the } k\text{-th sample's true class label is } i' \text{ (not } i) \end{cases} \tag{7}$$

Each set $S_{(i)}$ is generated by validating each individual kNN classifier (with results $R_{i,j}$) with a test set of length m, in our case a set of m audio segments with known true class label.

Each BN i, makes the final decision for the i-th binary sub-problem, based on the conditional probability $P_i(k) = P(Y_i(k) = 1|R_{i,1}^{(k)}, R_{i,2}^{(k)}, R_{i,3}^{(k)})$, i.e. the probability that the k-th input sample's true class label is i, given the results of the individual kNN classifiers. The process of calculating P_i is called *inference* and it is in general a very time consuming task. However, for the adopted BNC architecture no actual inference algorithm is needed, since the required conditional probability is given by the CPT itself, that has been learned in the training phase. Another advantage of the specific architecture *is that no assumption of conditional independence between the input nodes is made*, like e.g. in the Naive Bayesian Networks. After the probabilities $P_i(k)$, $i = 1, \dots, 6$ are calculated for all binary subproblems, the input sample k is classified to the class with the largest probability, i.e.

$$WinnerClass(k) = \arg\max_i P_i(k)$$

III. Experimental results

A. Datasets and System Training

In order to train and test the proposed system, 6 datasets D_i, $i = 1 \dots 6$ consisting of 200 minutes of movie recordings have been compiled. Almost 5000 audio samples have been extracted and manually labelled as ”music”, ”speech”, ”others”, ”shots”, ”fights” and ”screams” (almost 800 samples per class). The duration of those audio segments varies from 0.5 to 10 seconds. The data was collected from more than 30 films, covering a wide range of genres (e.g. drama, adventure). Some of the films were chosen not to contain violence, and were therefore used only for populating the non-violent classes. The data collection process was carried out by 4 different persons and the true label of each audio segment was set only according to the audio information. In other words, when establishing ground truth, the labellers were working directly on the audio stream of each movie and were actually using auditory sense alone. In order to train the binary sub-classifiers used in the OVA scheme, six more datasets D_i' have been created, each one containing audio samples from all other classes, than i. For example D_4' contains segments that are *not* labelled as ”shots”. After the datasets D_i and D_i', D_i, $i = 1 \dots 6$ have been created, 20% of the audio samples are used for populating the individual kNN classifiers. At a second step, the BNs are trained, via the validation of the respective kNN classifiers, as described in Paragraph II-B. Towards this end, 60% of the datasets are used. The remaining 20% of the audio data is used for testing the final system.

B. Overall System Testing

In order to test the overall classification system, hold-out validation has been used. Therefore, each of the datasets D_i and D_i' were randomly separated as explained above and experiments were executed for different selection of the subsets. In total, 100 iterations were executed. The normalized average confusion matrix (C) is presented in Table II. For example $C_{2,2}$ is the percentage of the speech data that was indeed classified as speech, whereas $C_{6,1}$ is the percentage of ”Screams” segments that were classified as ”Music”.

TABLE II
AVERAGE CONFUSION MATRIX

	Classified					
True ↓	Mu	Sp	Ot	Sh	Fi	Sc
Music	63.31	5.79	13.48	3.67	5.67	8.08
Speech	2.10	85.06	5.40	0.96	3.77	2.71
Others	9.42	4.31	69.01	8.60	4.85	3.81
Shots	2.12	1.25	1.94	78.69	13.89	2.10
Fights	2.73	5.35	1.50	14.54	69.10	6.77
Screams	5.93	4.17	2.69	5.00	8.12	74.09

The diagonal of C is also the recall R_i of the classification results, i.e. the proportion of data with true class label i, that were correctly classified in that class. On the other hand, the precision of each class $Pr_i, i = 1 \dots 6$ (i.e the proportion of data classified in class i, whose true class label is indeed i) is defined as: $Pr_i = \frac{C_{i,i}}{\sum_{j=1}^{6} C_{ji}}$. The recall and precision values of each class are presented in Table III. The overall classification accuracy (i.e. the percentage of the data that were correctly classified) of the proposed method is 73.2%.

TABLE III
RECALL AND PRECISION PER CLASS

	Mu	Sp	Ot	Sh	Fi	Sc
RECALL:	63.3	85.1	69.0	78.7	69.1	74.1
PRECISION:	73.9	80.3	73.4	70.6	65.6	75.9

The percentage of 73.2% refers to the classification accuracy of the multi-class classification problem. Though this is a high performance rate according to the nature of the problem, one may prefer to use the proposed classification scheme as a binary classifier. For example, the confusion between ”Shots” and ”Fights” is quite large ($CM_{4,5} = 13.89$ and $CM_{5,4} = 14.54$). This means that a large amount of data that should classified as ”Shots” was classified as ”Fights” (and vise versa), but in both cases the content can be also characterized as violent. In general, this could be achieved by classifying each sample with class label 1, 2 or 3 as "Non-Violent" and the samples with class labels 4,5 or 6 as ”Violent”. It is obvious that the recall and precision values for the violent class would therefore be computed using the following equations:

$$Re_{violence} = \frac{\sum_{i=4}^{6}\sum_{j=4}^{6} C_{ij}}{\sum_{i=4}^{6}\sum_{j=1}^{6} C_{ij}} \tag{8}$$

$$Pr_{violence} = \frac{\sum_{i=4}^{6}\sum_{j=4}^{6} C_{ij}}{\sum_{i=1}^{6}\sum_{j=4}^{6} C_{ij}} \tag{9}$$

Applying equations 8 and 9 given the computed confusion matrix, the violence recall was found equal to 90.8% and the violence precision equal to 86.6%. This means that the overall binary classification accuracy was almost 89%.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a multi-class audio classification system for movies, with respect to violent content. In total, six audio classes were adopted (three of them violent). Exhaustive examination resulted in a number of audio features stemming from the nature of the signals in the specific classification problem. The classification scheme was generally based on the One Versus All architecture. Each class was modelled using a Bayesian Network which was used as an estimator of the respective class probability, given the input sample. To extract the above probability, each BN was used as a combination scheme for classifying a set of three audio feature vectors into the classes of each binary sub-problem of the OVA architecture.

The proposed scheme was tested using more than 3 hours of audio recordings from more than 30 movies, covering a wide range of genres. The overall performance of the multi-class classification system was found to be equal to 73.2%. This is a high classification performance, taking into account the number of classes and the fact that some classes are quite similar (i.e. the classes ”Shots” and ”Fights”). Finally, the proposed system could also be used as a binary classifier for the ”Violent” - ”Non Violent” problem. In this case of binary classification, almost 9% of the violent data was incorrectly classified (false negative rate), while less than 14% of the non-violent data were classified as violent (false alarm rate). The overall binary classification error is therefore almost 11%.

To sum up, the proposed method can be used both as a multi-class audio classification system, but also as a binary classifier, resulting (as expected) in different performance rates. For example, one could use the system for blocking violent content in movies with a high performance rate (binary problem), while more detailed semantic information could be obtained from the six-class classification results, with an error rate of almost 26%. In the future, new features could be examined and used, in order to achieve boosted performance of the classification task. On the other hand, more classes could be added in the classification problem, in order to have a more detailed description of the audio data. Furthermore, an audio segmentation algorithm could be implemented and combined with the audio classification scheme. Finally, the audio classification system could be combined with synchronized visual cues for increased classification performance.

REFERENCES

[1] A.J. Reiss, J.A. Roth, eds. (1993). Understanding and Preventing Violence. Washington, DC: National Academy Press.

[2] Vasconcelos, N.; Lippman, A., Towards semantically meaningful feature spaces for the characterization of video content, in Proc. *International Conference on Image Processing, 1997,* Pages: 25 - 28 Vol.1

[3] A. Datta, M. Shah, N. V. Lobo, ”Person-on-Person Violence Detection in Video Data”, in *IEEE International Conference on Pattern Recognition*, Canada, 2002.

[4] J. Nam, A.H. Tewfik, ”Event-driven video abstraction and visualisation”, *Multimedia Tools and Applications*, 16(1-2), 55-77, 2002

[5] Giannakopoulos T., Kosmopoulos D., Aristidou A., Theodoridis S., ”Violence Content Classification using Audio Features”, *Hellenic Artificial Intelligence Conference SETN-06*, LNAI 3955, 502-507, Heraklion, Greece, 2006

[6] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 3d edition*. Academic Press, 2005.

[7] Mark A. Bartsch and Gregory H. Wakefield, ”Audio Thumbnailing of Popular Music Using Chroma-Based Representations”, in *IEEE Trans Multimedia*, Vol. 7, No. 1, Feb 05

[8] Ryan Rifkin, Aldebaro Klautau ”In Defense of One-Vs-All Classifcation”, in *The Journal of Machine Learning Research*, Volume 5, Pages: 101 - 141, December 2004

[9] A. Garg, V. Pavlovic and T.S. Huang, “Bayesian Networks as Ensemble of Classifiers”, Proceedings of the IEEE International Conference on Pattern Recognition, pp. 779-784, Quebec City, Canada, August 2002.

[10] D. Heckerman, “A Tutorial on Learning With Bayesian Networks”, Microsoft Research, MSR-TR-95-06, Mar. 1995