Audio Engineering Society

Convention Paper

Presented at the 142nd Convention
2017 May 20–23 Berlin, Germany

Extending Temporal Feature Integration for Semantic Audio Analysis

Lazaros Vrysis[1], Nikolaos Tsipas[2], Charalampos Dimoulas[3] and George Papanikolaou[4]

[1,2,3,4] *Aristotle University of Thessaloniki, 54124, Greece*

lvrysis@auth.gr

ABSTRACT

Semantic audio analysis has become a fundamental task in contemporary audio applications, making the improvement and optimization of classification algorithms a necessity. Standard frame-based audio classification methods have been optimized and modern approaches introduced engineering methodologies attempting to capture the temporal dependency between successive feature observations, forming the so-called texture windows. In this paper, an enhancement of this type of processing, known as temporal feature integration, is proposed, by employing and testing alternative deployable measures. Under this scope, new functions capturing the texture window shape are introduced and evaluated. The ultimate goal of this work is to highlight the best performing measures allowing the formation of robust aggregated feature-sets, able to increase performance in audio classification tasks.

1 Introduction

Semantic audio analysis is ubiquitous in contemporary audio applications; General Audio Detection and Classification (GADC), Voice Activity Detection (VAD), audio pattern classification and various recognition tasks are common [2], where speech and music classes are dominant in the encountered classification taxonomies [1]. These kinds of processing are useful -and often deployed- on mobile computational environments [3]; there is no doubt that expectations, both in terms of automatic audio pattern recognition accuracy and reduced computational requirements are high. Thus, algorithms and processing workflows should always be under review and revision. Standard classification algorithms are executed in two major steps; the extraction of audio features on short-time audio frames, followed by the utilization of classification systems for training and testing purposes on the resulting feature data, thus, defining a per frame audio analysis strategy. This kind of processing provides satisfactory results in most cases, but generally inside every sound sample there are parts that are not representative of the particular event; these segments are the ones that are most likely to be misclassified [4]. There is information and patterns in the time-based sequence of the processing frames that can be exploited in order to improve the accuracy of various classification tasks.

1.1 Related Work

Modern approaches make use of additional processing steps by incorporating the knowledge that is offered by successive frames into measures, a process called temporal feature integration [5]. Temporal feature integration techniques fall into two

major categories, depending on the stage at which integration takes place. If the process is executed at the feature extraction level, then the integration is called *early*, while if it is performed at the classifier level it is called *late* [6]. Early integration methods attempt to capture a more distinctive view of the signal and route more representative data to modeling algorithms versus simple frame-based approaches, thus improving performance. The variability inside a class can be reduced, resulting in finer modelling of the common attributes amongst the samples of the same sound category. Additionally, many short-time instances are grouped together and the amount of data derived to the -usually- slow classifiers is reduced, relieving the classification process. Early integration methods are divided into four groups [4], [7]. First, a simple way to merge the information which is provided by many subsequent frames is the computation of their statistical instances (like *Mean*, *Variance*, *Median*) over each texture window; others suggest the use of autoregressive *(AR)* models for capturing their evolution in time [8]. Another strategy estimates the temporal dependency between successive feature observations by exploiting the information provided by the spectrum of these features [9], and, one more method, which is referred to as stacking, consists in concatenating all the feature vectors observed over a texture window into a single vector where the temporal dependency between the features is then modelled by a classifier [10]. Regarding late integration methods, the deployment of *HMM*-type classifiers or the application of decision making mechanisms on many partial decisions belong to the most common practices [11]. These strategies lead to a performance boost against typical short time window analysis techniques with each one being able to cumulatively increase the accuracy of the evaluated system. Statistical feature integration proves to be efficient and the simplest among others [12].

1.2 Paper Overview

The current paper attempts to enhance the statistical feature integration methodology by extending and thoroughly evaluating the measures that can be deployed, introducing also functions for capturing the shape of a texture window as well; not solely considering successive feature values as independent random variables. A wide set of typical statistical instances is tested *(Mean, Standard Deviation, Kurtosis, Skewness, Max, Min),* well known measures for feature extraction (i.e. *Crest Factor* and *Flatness*) are applied on texture windows, while new functions, like *Mean Crossing Rate (MCR)* and *Mean Sequential Absolute Difference (MSAD)* are presented. The target of this work is to identify the best performing measures for early temporal integration, focusing on simple feature engineering, avoiding complexity and forming a compact and robust set of aggregated features which can improve performance in audio classification tasks.

The rest of the paper is organized as follows: in section 2 the proposed methodology is presented, section 3 covers all aspects of the experimental testing and evaluation of the method and, finally, section 4 exposes conclusions and our intentions for further work.

2 Proposed methodology

As already mentioned statistical feature integration methods fall into the group of early integration techniques; specifying data aggregation transformations on the feature extraction level. The time series of subsequent short-frame feature vector values form a larger frame, which is called texture window. The proposed metrics attempt to capture information that lies into the succession of consecutive standard short-frame feature values. Let us describe the approach more formally. Assume that *Z[k]* is the *U*-dimensional feature vector for the *k-th* frame, where $Z[k]=[z_1[k], z_2[k],\ldots,z_U[k]]$ are the components of this vector. If a number of *Q* integration functions is applied to each z_i component over the *n-th* texture window, which contains a sequence of *L* frames from *k-L+1* to *k*, a *UxQ*-dimensional feature vector $W[n]=[w_1[n],w_2[n], \ldots,w_{UxQ}[n]]$ is created. Focusing on a single component *x* of the original vector, applying a function *f* for temporal integration, results to the X_F component of the integrated feature vector, which can be expressed as:

$$X_F = f(x[k-L+1], \ldots, x[k]) \quad (1)$$

A wide range of statistical measures can be used in place of function *f*, including the *Mean Value*,

Standard Deviation, Skewness or Kurtosis, defined as follows:

Mean Value (MEAN). *MEAN* is defined as the sum of the values of a feature, divided by the number of values (Eq. 2).

$$X_{MEAN}[n] = \frac{1}{L}\sum_{m=k-L+1}^{k} x[m] \quad (2)$$

Standard Deviation (STD). *STD* is used to quantify the amount of variation of feature-values over a texture window. *STD* is computed according to Eq. 3.

$$X_{STD}[n] = \frac{1}{L}\sum_{m=k-L+1}^{k} x[m] \quad (3)$$

Skewness (SKEW): Being a standard statistical measure, *Skewness* is calculated by dividing the third central moment *($\mu_3[n]$)* of feature-values by their *Standard Deviation* raised to the power of three, as defined in Eq. 4.

$$X_{KURT}[n] = \frac{\mu_3[n]}{\sigma^3[n]} \quad (4)$$

Kurtosis (KURT): *Kurtosis* is calculated by dividing the third central moment *($\mu_4[n]$)* of feature-values by their *Standard Deviation* raised to the power of four, as defined in Eq. 5.

$$X_{KURT}[n] = \frac{\mu_4[n]}{\sigma^4[n]} \quad (5)$$

While these measures have been used widely, providing robust performance, they partly capture the temporal information of successive features. More metrics can be deployed in order to exploit the knowledge that is hidden inside the time-series of the features; some of them are presented below.

Relative Standard Deviation (RSD): *RSD* is defined as the ratio of the *Standard Deviation* to the *Mean* of a feature's values over the texture window (Eq. 6).

$$X_{CV}[n] = \frac{X_{MEAN}[n]}{X_{STD}[n]} \quad (6)$$

High Crest Factor (HCF): Similar to *Crest Factor* measure that is used in waveforms**,** *HCF* is calculated by dividing the *Maximum* by the *Mean* value of feature-values inside a texture window (Eq. 7).

$$X_{HCF}[n] = \frac{MAX(x[k-L+1],\dots,x[k])}{X_{MEAN}[n]} \quad (7)$$

Low Crest Factor (LCF): In contrast to the *HCF* measure, *LCF* is calculated by dividing the *Minimum* by the *Mean* value of a feature inside a texture window (Eq. 8).

$$X_{LCF}[n] = \frac{MIN(x[k-L+1],\dots,x[k])}{X_{MEAN}[n]} \quad (8)$$

Mean Absolute Sequential Difference (MASD): Like the *Standard Deviation (STD)* measure, *MASD* purports to quantify the amount of variation of a feature's values inside a texture window, taking also into account the frequency of the changes. It is calculated as the *Mean* value of the summed up absolute differences of successive feature values, as defined in Eq. 9.

$$X_{MASD}[n] = \frac{1}{L-2}\sum_{m=k-L+2}^{k} |x[m]-x[m-1]| \quad (9)$$

Mean Squared Sequential Difference (MSSD): *MSSD* is a similar measure to *MASD*. In this case, the squared differences -instead of absolute- between successive feature-values are calculated. The formula is exposed in Eq. 10.

$$X_{MSSD}[n] = \frac{1}{L-2}\sum_{m=k-L+2}^{k} |x[m]-x[m-1]|^2 \quad (10)$$

Mean Crossing Rate (MCR): Inspired by the well-known *Zero Crossing Rate (ZCR)* that is directly applied on raw signals, *MCR* estimates the alternations of successive feature-values inside a texture window, in respect to their *Mean* value (Eq. 11).

$$X_{MCR}[n] = \frac{1}{L-1}\sum_{m=k-L+1}^{k} 1_{R_{<1}}(d[m]) \quad (11)$$

where

$$d[m] = [x[m] - X_{MEAN}[n]][x[m-1] - X_{MEAN}[n]] \quad (12)$$

Flatness (FLA): Like *Spectral Flatness*, temporal *Flatness* is calculated by dividing the geometric mean of the feature's values by their arithmetic mean, inside a texture window, according to Eq. 13.

$$X_{FLA}[n] = \frac{\sqrt[L]{\prod_{m=k-L+1}^{k} x[m]}}{\frac{\sum_{m=k-L+1}^{k} x[m]}{L}} \quad (13)$$

3 Experimental Results

In order to perform objective evaluation of the proposed measures, a comparative approach was followed. As part of this approach, the performance of the new metrics is compared to standard temporal integration and simple frame based methods, by conducting feature ranking tests and executing typical audio classification tasks, utilizing a common, standardized and pre-annotated dataset.

3.1 Dataset

The performance of the proposed methodology was evaluated by setting up two classification tasks, a *Speech/Music/Other* ***(SMO)*** and a *Speech/Music* ***(SM)***. To do so, the popular *GTZAN* Music/Speech dataset was employed and extended by a third class *(Other)*. As the main goal of this work is to highlight efficiency considerations between existing and newly introduced metrics, and, having in mind that solving a *Music/Speech* problem, utilizing only the *GTZAN* dataset is a quite easy task, we decided to make a tougher one. To achieve this, a third class *(Other)* was added, containing all audio signals except speech and music (i.e. environmental sounds, human and animal bioacoustics, weather phenomena, engines, motors, other machinery and all kinds of noise). In order to keep balance between the classes, *Other* class was designed according to the specifications of the *GTZAN* dataset; all three classes have the same length of 1920s, and follow an uncompressed 22,050Hz/16bit/mono audio format. The dataset is publicly available at *http://research.playcompass. com/files/LVLib-SMO-1.zip*. Concerning train and test subset formation issues, it is worth noting that all testing and evaluation procedures employed a 3-fold cross-validation strategy. Table 1 presents class and durations information about the used dataset.

Class	Music	Speech	Others
Duration	1920	1920	1920

Table 1. Class-separated audio duration for the used dataset.

3.2 Feature formation

Taking into consideration the objectives of this work, and, in order to engineer both standard frame-based and temporal-integrated features, a baseline 22-dimension feature vector and two major feature formation strategies were deployed. The selected audio features cover a wide range of temporal, spectral and cepstral aspects of the signal, while similar metrics have been successfully employed in various audio classification tasks including voice activity detection, speech/non-speech discrimination and speaker diarisation [1]. Regarding the feature processing architecture, long- and short-frame manufacturing pipelines were designed. The first approach, processes long blocks of audio and directly computes the values of the features from raw audio data, while the second one, executes short-frame feature calculation, so as to form texture windows, resulting into aggregated features, after performing statistical integration. Based on experience derived from previous work in the field [5], short-frame duration is few dozen of milliseconds, while long frames are about a second long. Aiming to match the temporal resolution for frame-based and temporal-integrated features and, in respect to the sampling rate of the recordings, the following setup was decided: 32768 samples long-block size with 50% overlap, 512 samples short-block size with 50% overlap, 128 texture window length with 50% overlap.

More specifically, each baseline feature is extracted for both short- and long-frame formats, following a default *Hanning* window configuration, while a pre-preprocessing step, scaling all features to have zero mean and unit variance, is applied afterwards. Next, three different feature processing pipelines are employed for formatting the final feature sets. First, the 22-dimension *Standard Frame Based Feature Set* ***(SFB-FS)*** is directly derived from long-frame fea-

tures, following the *Standard Frame Based* ***(SFB)*** procedure. Second, the complete set of short-block features is aggregated in order to calculate mean and variance values, based on the *Standard Temporal Integration* ***(STI)*** procedure. Since each aggregated feature is represented by its *Mean* and *Variance* statistics over the texture window, the total number of components comprising the final feature vector is equal to two times the baseline number of features. Thus, the *Standard Temporal Integration Feature Set* ***(STI-FS)*** consists of 44 aggregated values in addition to the 22 features of the *SFB-FS***.** Finally**,** 7 of the proposed metrics *(RSD, HCF, LCF, MASD, MSSD, MCR, FLA)* were computed on all baseline features but *MFCCs*, applying the *Enhanced Temporal Integration* ***(ETI)*** methodology, and extending the *STI-FS* by 70 additional feature values, forming the *Enhanced Temporal Integration Feature Set* ***(ETI-FS)***. The baseline features that were exploited for each feature formation strategy are presented in Table 2. A standard notation is followed throughout the manuscript for aggregated features, where subscripts symbolize the integration function and baseline text the source feature (i.e. CEN_{STD} denotes the *Standard Deviation* of *Spectral Centroid*).

#	Feature	Symbol	SFB	STI	ETI
1	Energy	ENE	✓	✓	✓
2	ZCR	ZCR	✓	✓	✓
3	S. Flatness	FLA	✓	✓	✓
4	S. Flux	FLU	✓	✓	✓
5	S. Rolloff	ROL	✓	✓	✓
6	S. Centroid	CEN	✓	✓	✓
7	S. Spread	SPR	✓	✓	✓
8	S. Kurtosis	KURT	✓	✓	✓
9	S. Skewness	SKEW	✓	✓	✓
10	S. Slope	SLO	✓	✓	✓
11-22	MFCCs (12)	MFCC	✓	✓	

Table 2. Baseline audio features.

A decision not to include *MFCCs* and *MAX, MIN* measures for calculating the corresponding aggregated feature vector components was taken, because this would result in significant increase of the final vector's dimension, while preliminary tests indicated low discriminative power for these attributes. Feature-vector dimensions for all three derived features sets are outlined in Table 3.

Subset	SFB-FS	STI-FS	ETI-FS
Length	22	66	156

Table 3. Vector-dimension for each feature set.

A *Docker* container including *Yaafe* [13] and *Sonic Visualizer* [14] including various *VAMP* plugins [15] was employed for feature extraction, while various custom scripts were developed in *VBA* and *MATLAB* for implementing the statistical feature integration functions.

3.3 Performance evaluation

It is obvious that feature integration leads to high-dimensional vectors, an outcome that generally does not necessarily deliver better performance. The well-known aspect of the "curse of dimensionality" suggests that a smaller feature set, containing only the most salient features is preferable, while at the same time it reduces computational complexity and load. Therefore running times are shortened and performance improvements can be achieved [16]. This work purports to introduce metrics concentrating discriminative information into single features, thus resulting to more powerful feature sets, while keeping feature sets size small. In this context, feature selection and ranking was a top-priority, in order to isolate salient features. Regarding this process, bibliographical suggestions, empirical investigations and more sophisticated quantitative algorithmic evaluation of all the initial features were taken into account. Many methodologies have been proposed for composing the optimal selection out of a wider set of available features; some approaches try to estimate the significance of an attribute by calculating certain measures, while others iteratively test candidate feature subsets, trying to maximize classification performance [19]. A representative collection of this kind of algorithms was utilized, following typical testing procedures.

The evaluation of the proposed metrics was completed by conducting appropriate classification tests, following the *SMO* and *SM* taxonomies as previous-

ly described. For this purpose, the standard evaluation measure of *Accuracy* was employed. *Accuracy* provides an overall evaluation of the achieved recognition score by estimating the ratio of the total number of correctly classified instances to the total number of samples.

3.4 Results / Discussion

The essential implementations for estimating the discriminative efficiency of the proposed features took place using ranking algorithms of the *Weka* and *RapidMiner Studio* software platforms [20]. Feature ranking tests were carried out utilizing the *ETI* feature set, since it is a superset of *SBF* and *STI* sets, including a full range of the formatted features, while three different methodologies were applied. First, feature weighting was performed by calculating the significance of each feature with respect to the class attribute making use of the *Information Gain* measure. Secondly, the efficiency of the features was estimated by computing their relevance to the class attribute, setting as weights the coefficients of a hyperplane calculated by an *SVM* and, finally, an evolutionary feature subset optimization process was carried out, delivering the most salient features, exploiting a genetic algorithm. Table 4 exposes the corresponding results; the top 20 performing features for each procedure are presented.

Weighting by Information Gain	**Weighting by SVM**	**Evolutionary Subset Selection**
$MFCC2_{STD}$	$MFCC2_{STD}$	ZCR_{MASD}
$MFCC3_{STD}$	ENE_{FLA}	ENE_{FLA}
$MFCC4_{STD}$	ENE_{LCF}	$MFCC3_{MEAN}$
ENE_{FLA}	ENE_{MSSD}	$MFCC3_{STD}$
$MFCC5_{STD}$	$MFCC2_{MEAN}$	$MFCC5_{MEAN}$
ENE_{LCF}	FLA_{FLA}	FLA_{SKEW}
ZCR_{RSD}	KUR	FLA_{FLA}
FLU_{MEAN}	SPR_{MCR}	FLU_{RSD}
ZCR_{STD}	ENE_{RSD}	FLU_{LCF}
$MFCC1_{STD}$	$MFCC3_{STD}$	CEN_{HCF}
ENE_{RSD}	FLU_{LCF}	CEN_{LCF}
ZCR_{FLA}	FLA_{MEAN}	SPR_{MCR}
SKE_{SKEW}	SPR	SLO_{HCF}
$MFCC6_{STD}$	CEN_{MEAN}	ENE_{RSD}
ZCR_{HCF}	SLO_{MEAN}	ENE_{SKEW}
SKE_{STD}	ENE_{MASD}	MFCC5
KUR_{RSD}	FLU_{SKEW}	ENE_{HCF}
KUR_{HCF}	SPR_{MEAN}	ENE_{MCR}
ROL_{LCF}	FLA_{MASD}	ZCR_{FLA}
SKE_{LCF}	MFCC4	MFCC6

Table 4. Top 20 salient features in respect to three different feature selection algorithms.

Many occurrences of the newly introduced features in the top places imply that they outperform their competitors and are more powerful. As we can notice, most of the high-ranked attributes are baseline features, decorated with standard and enhanced temporal integration transformations, in contrast to non-aggregated standard frame-based. *STD* metric is clearly a top performer while *MASD*, *LCF*, *RSD* measures do pretty well; it seems that new integration methods can be effective. Fig. 1 demonstrates a 3D view of the *Music*, *Speech* and *Other* instances, based on a vector-space composed by selected proposed features.

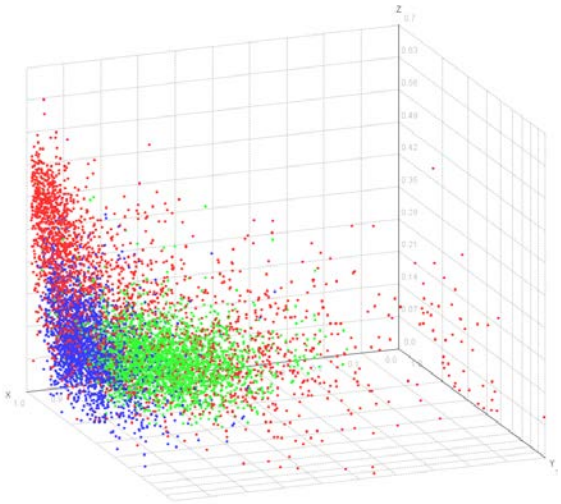

Figure 1 Speech (Green), Music (Blue) and Other (Red) instances positioned on the 3D space, constructed by the ENE_{FLA} (x-axis), FLA_{HCF} (y-axis) and SPR_{MCR} (z-axis) components.

At this point, it is necessary to note that, especially non-exhaustive feature ranking procedures give only an estimation of the selected features quality; so, additional evaluation under realistic classification tasks becomes mandatory. Thus, the performance of each feature set was evaluated by solving the aforementioned *Speech/Music/ Other (SMO)* and *Speech/Music (SM)* problems, exploiting a variety of classification modelling methods. The experiments took place utilizing *Naïve Bayes* ***(NV)***, *Logistic Regression* ***(LR)***, *Support Vector Machine* ***(SVM)*** *and Artificial Neural Network* ***(ANN)*** *classifiers* under *RapidMiner Studio's* standard setup. An optimal subset of features was preferred for each of the original feature sets.

The dimension for each vector was decided after reviewing classification performance considerations in respect to different dimensions. After experimenting, the 35 best performing features were used for the *STI* and *ETI* feature sets, while 20 features selected for the *SFB* set. Figure 1 demonstrates the performance evolution for the *LR* classifier under *SMO* scheme, in respect to the number of ranked features used.

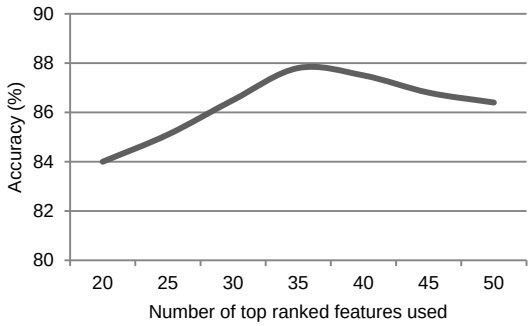

Figure 2 Classification accuracy for LR algorithm in respect to number of features used, under SMO (Speech, Music, Others) scheme.

Table 5 demonstrates performance scores, in terms of accuracy, under the SMO discrimination task for each feature set, exploited by all four classifiers. The results reveal superior performance for the proposed features, under all setups. It is worth mentioning that *SBF-FS* has generally poor performance; this behavior was more or less expected, as *SBF-FS* is quite small, incapable for handling such a challenge effectively. Additionally, we can notice that the *ANN* Classifier outperforms all other algorithms, delivering adequate scores, even if driven by weak feature sets.

	NB	**LR**	**SVM**	**ANN**
SBF-FS	54.8	59.0	57.3	69.1
STI-FS	69.6	78.4	77.1	81.9
ETI-FS	75.4	87.5	86.3	86.9

Table 5 Classification ratings (Accuracy %) in the 3-class SMO problem (Speech, Music, Others).

Finally, a similar setup was used for the easier, and more common, *SM* classification problem. In general, accuracy measurements are much higher than the previous task, as expected after the elimination of one class. Analyzing the individual results, we can note similar considerations as before; *STI-FS* includes a powerful package of features demonstrating that the process of temporal feature integration can improve performance in classification tasks. Furthermore, *ETI-FS*, which extends statistical measures, brings further improvements proving the discriminative ability of the proposed measures.

	NB	**LR**	**SVM**	**ANN**
SBF-FS	67.9	76.3	76.8	82.4
STI-FS	89.8	94.8	94.9	94.2
ETI-FS	91.6	97.1	96.3	96.5

Table 6. Classification ratings in the 2-class SM problem (Speech, Music).

4 Conclusion / Further work

This work attempts to extend temporal feature integration methods by proposing a robust and lightweight set of measures, aiming at improving performance on generic audio classification tasks. These measures rely heavily on statistical processing; functions modeling the succession of consecutive instances are incorporated as well. A solid evaluation of the proposed approach was presented and positive results were observed. Nevertheless, further testing, utilizing a broader dataset should be

conducted, in order to identify and isolate the most powerful integrating functions with respect to the baseline features. Furthermore, an exhaustive comparison of all early and late integration methods would be valuable to highlight the best performing approaches (or combinations of them).

References

[1] N. Tsipas, L. Vrysis, C. Dimoulas, G. Papanikolaou, "Efficient audio-driven multimedia indexing through similarity-based speech/ music discrimination", *Multimedia Tools and Applications*, pp. 1-19 (2017).

[2] L. Vrysis, N. Tsipas, C. Dimoulas, G. Papanikolaou, "Mobile audio intelligence: From real time segmentation to crowd sourced semantics", *Proceedings of the Audio Mostly 2015 on Interaction with Sound*, p. 37 (2015).

[3] L. Vrysis, N. Tsipas, C. Dimoulas, G. Papanikolaou, "Crowdsourcing Audio Semantics by Means of Hybrid Bimodal Segmentation with Hierarchical Classification" *Journal of the Audio Engineering Society*, 64(12), pp. 1042-1054 (2016).

[4] S. Ntalampiras, I. Potamitis, N. Fakotakis, "Exploiting temporal feature integration for generalized sound recognition", *EURASIP Journal on Advances in Signal Processing*, 807162 (2009).

[5] N. Tsipas, L. Vrysis, C. Dimoulas, G. Papanikolaou, "Methods for Speech/Music Detection and Classification", *MIREX 2015*, (2015).

[6] J. Flocon-Cholet, J. Faure, A. Guérin, P. Scalart, "An investigation of temporal feature integration for a low-latency classification with application to speech/music/mix classification", *Audio Engineering Society Convention 137*, (2014).

[7] A. Meng, P. Ahrendt, J. Larsen, L. Hansen, "Temporal feature integration for music genre classification", *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5), pp. 1654-1664 (2007).

[8] A. Meng, "Temporal feature integration for music organization", *Ph.D. dissertation, Technical Univ. of Denmark*, (2006).

[9] M. McKinney, J. Breebart, "Features for audio and music classification", *Int. Symp. Music Inf. Retrieval*, pp. 151–158 (2003).

[10] M. Slaney, "Semantic-audio retrieval", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. IV-4108 (2002).

[11] C. Joder, S. Essid, G. Richard, "Temporal integration for audio classification with application to musical instrument classification", *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1), pp. 174-186 (2009).

[12] N. Tsipas, P. Zapartas, L. Vrysis, C. Dimoulas, "Augmenting social multimedia semantic interaction through audio-enhanced web-tv services", *Proceedings of the Audio Mostly 2015*, p. 34 (2015).

[13] B. Mathieu, S. Essid, T. Fillon, J. Prado, G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software", *ISMIR 2010*, pp. 441-446 (2010).

[14] C. Cannam, C. Landone, M. Sandler J. Bello, "The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals", *ISMIR 2006*, pp. 324-327 (2006).

[15] J. Salamon, E. Gómez, "Mir.edu: An open-source library for teaching sound and music description", *ISMIR 2014*, (2014).

[16] R. Kotsakis, G. Kalliris, C. Dimoulas, "Investigation of salient audio-features for pattern-based semantic content analysis of radio pro-

ductions," *Proceedings of the 132nd AES Convention*, pp. 513-520 (2012).

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, "The WEKA data mining software: an update", *ACM SIGKDD explorations newsletter*, 11(1), pp. 10-18 (2009).

[18] V. Goyal, "A Comparative Study of Classification Methods in Data Mining using RapidMiner Studio", *International Journal of Innovative Research in Science & Engineering*, (2014).

[19] B. Schowe, "Feature selection for high-dimensional data with RapidMiner", *Proceedings of the 2nd RapidMiner Community Meeting and Conference*, (2011).

[20] R. Kotsakis, G. Kalliris, C. Dimoulas, "Investigation of broadcast-audio semantic analysis scenarios employing radio-programme-adaptive pattern classification", *Speech Communication*, vol. 54, no. 6, pp. 743-762 (2012).