acoustics

MDPI

Article

An Enhanced Temporal Feature Integration Method for Environmental Sound Recognition

Vasileios Bountourakis [1], Lazaros Vrysis [2,*], Konstantinos Konstantoudakis [3] and Nikolaos Vryzas [4]

[1] Department of Signal Processing and Acoustics, Aalto University, 02150 Otakaari 5, Espoo, Finland; vasileios.bountourakis@aalto.fi

[2] School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

[3] Information Technologies Institute of Thessaloniki, 60361 Thessaloniki, Greece; k.konstantoudakis@iti.gr

[4] School of Journalism and Mass Media Communication, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; nvryzas@auth.gr

* Correspondence: lvrysis@auth.gr; Tel.: +30-6978-917-452

Received: 30 December 2018; Accepted: 29 April 2019; Published: 8 May 2019

Abstract: Temporal feature integration refers to a set of strategies attempting to capture the information conveyed in the temporal evolution of the signal. It has been extensively applied in the context of semantic audio showing performance improvements against the standard frame-based audio classification methods. This paper investigates the potential of an enhanced temporal feature integration method to classify environmental sounds. The proposed method utilizes newly introduced integration functions that capture the texture window shape in combination with standard functions like mean and standard deviation in a classification scheme of 10 environmental sound classes. The results obtained from three classification algorithms exhibit an increase in recognition accuracy against a standard temporal integration with simple statistics, which reveals the discriminative ability of the new metrics.

Keywords: environmental sound recognition; temporal feature integration; statistical feature integration; semantic audio analysis; audio classification

1. Introduction

Environmental Sound Recognition (ESR) is a semantic audio application that has received considerable attention in recent years. The goal of ESR is to capture environmental sounds using audio sensors and assign them to predefined categories (or classes) by applying semantic labels to them. By environmental sounds, we refer to various sounds that are both natural and artificial, other than speech and music, which are present in an acoustic scene. Practical applications of ESR include hearing-aid technology, home-monitoring, audio surveillance, assisting robotics, animal bioacoustics, and information retrieval applications such as keyword-based search in audio archives [1]. An emerging trend in audio is the incorporation of ESR applications into portable or wearable devices [2]. For example, a mobile device could be designed by applying ESR in order to automatically change the notification mode based on the knowledge of the user's surroundings [3]. The increasing interest in the field of ESR has heightened the need for optimized algorithms and processing workflows that could achieve higher recognition accuracy and reduce computational requirements.

Automatic sound recognition is commonly executed in two processing stages. First, a set of audio features is extracted on short-time audio segments (or frames) over which the signal can be considered stationary and then the resulting feature data are used by classification algorithms for training and testing purposes. However, this frame-based method neglects the information conveyed in the temporal evolution of the signal, since it assumes that the observed features in successive frames are statistically independent. Several strategies, often termed *temporal feature integration*, use additional steps in order to incorporate this information into measures, which improves the accuracy of classification algorithms [4]. Although considerable research has been devoted to the frame-based approach aiming at exploring the most appropriate features and classifications algorithms [5–8], rather few studies have addressed the issue of temporal feature integration in the context of sound recognition, with most of them focusing on music signals.

Temporal feature integration techniques generally fall into two major categories. If the integration is performed at the feature extraction level by combining short-time features over a larger frame, which is termed texture window, the process is called early integration, as opposed to late integration, where features are combined on the classifier level [9]. Early integration results in reduced variability among the features of the sound samples within the same semantic category, while the lower amount of data delivered to the classification algorithm reduces computational complexity [10]. A straightforward type of early integration consists in the computation of statistical instances like the mean and variance of the short-time features [4]. In Reference [11], Meng et al. introduce autoregressive models in order to capture the temporal evolution of features in a musical genre classification task. Another temporal integration strategy is presented in Reference [12], where the power spectrum of the feature values across consecutive short frames captures the temporal dependencies for each feature. Joder et al. [13] explore both early and late integration methods in a musical instrument recognition task. A fusion scheme, which combines features integrated at different texture window lengths, is proposed in Reference [14] for generalized sound recognition (environmental sounds and speech). Flocon-Cholet et al. [9] propose several temporal integration methodologies for a speech/music/mix classification task in a low-latency framework so that they could be used in real-time applications. A real-time environmental sound recognition system for Android mobile devices using temporal integration with simple statistics is implemented by Pillos et al. [15]. Among the different approaches, statistical feature integration achieves the best balance between classification performance and computational complexity [16].

The purpose of the paper is to expand the investigation of a new method for temporal feature integration that was recently introduced in Reference [10], where simple statistics mixed with newly proposed functions capturing the texture window shape are evaluated for their performance in a speech/music/other classification task. The novelty of this work lies in the investigation of the applicability of these new metrics for early temporal integration in the context of environmental sound recognition. The metrics are tested on a database of environmental sounds created by the authors and their effectiveness is compared against the standard temporal integration with simple statistics (mean and variance) and the frame-based method without integration. This work focuses on simple engineering, avoiding complexity, and aspires to propose a robust set of aggregated features that improves performance in environmental sound classification tasks. The proposed technique is also evaluated on an extensively tested public dataset (UrbanSound8K), and compared against some computationally heavier Deep Learning approaches.

The remainder of the paper is structured as follows. In Section 2, the database and the proposed methodology are presented, Section 3 contains a precise description and evaluation of the experimental results, and the conclusions are drawn in Section 4.

2. Materials and Methods

2.1. Database

A reliable comparison of ESR methods presented in different papers is only valid if all the evaluation tests are performed on a common universal database. In the absence of such a database, any performance benchmarking of the various approaches is futile. To address this problem, we decided to create a library of environmental sounds that has the potential to be used as a reference database for future benchmarking. The library, which we called BDlib_2, is an extended version of the library presented in Reference [5] and is publicly available at the following location: research.playcompass.com/files/BDLib-2.zip. The library was created by identifying and isolating 10 s audio segments that represent discrete sound categories from the following sources: BBC Complete Sound Effects Library (bbcsfx.acropolis.org.uk) and freesound.org. Particular care has been taken in the selection of the segments in order to keep the sound samples clean of background noise and prevent overlapping of the sound classes. The library is organized in the following 10 classes: airplanes, alarms, applause, birds, dogs, motorcycles, rain, rivers, sea waves, and thunders. Our intention was to include classes of sounds—encountered either in indoor or outdoor soundscapes—that are extensively used in similar recognition schemes. Each class is equally represented in the database by 18 audio files with great variations between them, which reflect real life situations. All the recordings are uncompressed mono audio files in WAV format, with a sampling rate of 44.1 kHz and 16 bit analysis. The total duration of the collected files is 1800 s, which is generally considered a relatively small database for the specific classification task. In order to overcome the problem of data scarcity, we applied two common data augmentation techniques to the whole dataset that have been reported to increase model accuracy for environmental sound classification tasks [17]: time stretching by the factors 0.85 and 1.15 and pitch shifting by −4 and +4 semitones. The deformation factors were selected so that the deformed data preserve their semantic meaning. The idea is that, if a model is trained on additional deformed data, it can generalize better on unseen data. The augmentations were applied using the command line tool Sound eXchange (SoX) [18]. The resulting duration of the complete database after augmentation is 8820 s, which is comparable to the duration of similar databases that are regarded as reliable. All things considered, the authors believe that this database meets the specifications of a reference environmental sound library and encourage its use for future benchmarking.

2.2. Feature Extraction

The first step of each ESR algorithm consists in extracting useful attributes from raw audio data that can describe the signal in a compact way. In practice, the signal is divided into small frames using a window function and a number of features are extracted from each frame separately. Each vector of features corresponding to a single frame is used as an instance for training and testing (for frame-based methods). Based on the domain where the extraction occurs, audio features can be grouped into three categories: time-domain (or temporal) features, frequency-domain (or spectral) features, and cepstrum-domain (or cepstral) features. It is crucial to choose appropriate features for each application that have the potential to result in effective discrimination between the sound classes. In this paper, the choice of the extracted features was based on the results of a previous study on ESR [5]. Table 1 lists the extracted audio features along with their corresponding dimensions (when not noted, the feature dimension is 1) and their abbreviated symbols that are used in this paper. All features add up to a feature vector of 100 dimensions. At this stage, along with the extraction of features, each instance was annotated with the corresponding sound class label. The extraction was performed with the following open-source software tools: MIRtoolbox (MATLAB)—Version 1.7 [19], Marsyas–version 0.5.0 [20], and jAudio—Version 1.0.4 [21].

As stated in the introduction, our intention is to compare the frame-based method against temporal integration methods. In order to make a fair comparison, the window length of the frame-based method should match the length of the texture window of the temporal integration methods. To that end, the features were extracted with two different window configurations: long windows for the frame-based method and short windows over which the statistical integration is performed, which form a texture window with aggregated features for the temporal integration methods. Based on experience derived from previous work [22], the optimal duration for the short window is a few dozens of milliseconds and, for the long window, about one second. Accordingly, the following setup was selected, using a Hann window for both cases.

- short window: 2048 samples length (duration~46 ms) with 50% overlapping
- long window: 65,536 samples length (duration~1.48 s) with 50% overlapping

Therefore, extracting aggregated features over 64 subsequent short windows forming the texture window with 50% overlapping, we have the same temporal resolution for both the frame-based and the temporal integration methods.

Table 1. List of extracted features.

Feature	Symbol
Zero Crossing Rate	ZCR
Root Mean Square	RMS
Relative Difference Function	RDF
Spectral Centroid	CEN
Spectral Spread	SPR
Spectral Flux	FLU
Spectral Roll-Off	ROL
Spectral Skewness	SKEW
Spectral Kurtosis	KURT
Spectral Entropy	ENTR
Spectral Variability	VAR
Spectral Smoothness	SMO
Spectral Flatness Measure (24)	SFM
Spectral Crest Factor (24)	SCF
Brightness	BRI
Roughness	ROU
Irregularity	IRR
MFCC (13)	MFCC
Delta-MFCC (13)	DMFCC
LPCC (12)	LPCC

2.3. Temporal Feature Integration

As already mentioned, statistical feature integration methods fall into the group of early integration techniques. Several integration functions are applied on baseline features over successive short frames attempting to capture the temporal evolution of these features into aggregated features.

To describe this approach more formally, assume that U baseline features are extracted for each frame and **Z[*k*]** = [$z_1[k]$, $z_2[k]$, … , $z_U[k]$] is the U-dimensional original feature vector for the *k*-th frame. A texture window matrix contains a sequence of L frames from $k{-}L{+}1$ to *k*. Thus, the texture window matrix has a dimension of LxU. A number of Q integration functions [F_1, F_2, … F_Q] are applied to each z_i component over the *n*-th texture window. As depicted in Figure 1, every column of the texture window matrix is used as an input to each F_j function. As a consequence, a total of N = UxQ transformations will occur in each texture window *n*, which results in a new N-dimensional feature vector **W[n]**, whose elements are given by Equation (1).

$$w_p = F_j(z_i(k-L+1),\ldots,\ z_i[k]),\ \text{with}\ p\ =\ (i-1)\ \mathrm{Q}\ +\ j \tag{1}$$

The final feature vector **W[*n*]** = [$w_1[n]$, $w_2[n]$, … , w_{UxQ}[n]] is the integrated feature vector.

The most common statistical measures used in place of the function *F* are the Mean Value (MEA), Standard Deviation (STD), Skewness (SKE), and Kurtosis (KUR). For example, the Mean Value is calculated as follows.

$$X_{MEA}[n] = \frac{1}{L}\sum_{m=k-L+1}^{k} x[m], \tag{2}$$

where $x[m]$ is the original feature component of the *m*-th frame and $X_{MEA}[n]$ is the integrated component over the *n*-th texture window that contains *L* frames.

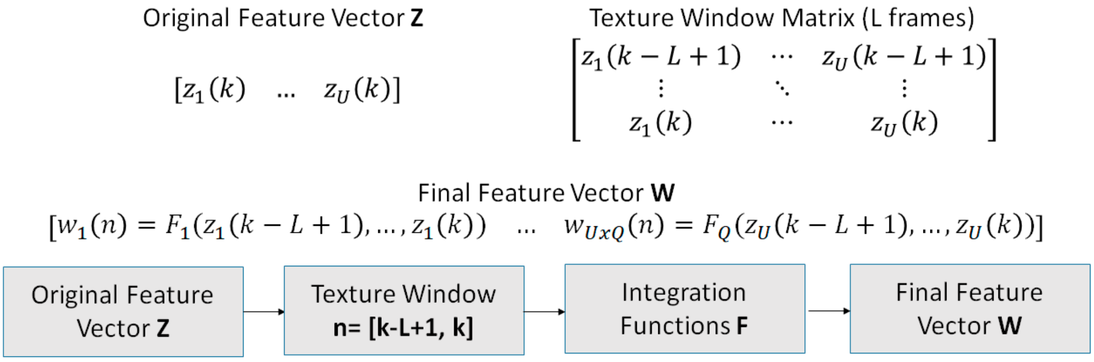
Original Feature Vector **Z**
$[z_1(k) \quad \ldots \quad z_U(k)]$
Texture Window Matrix (L frames)
$\begin{bmatrix} z_1(k-L+1) & \cdots & z_U(k-L+1) \\ \vdots & \ddots & \vdots \\ z_1(k) & \cdots & z_U(k) \end{bmatrix}$
Final Feature Vector **W**
$[w_1(n) = F_1(z_1(k-L+1), \ldots, z_1(k)) \quad \ldots \quad w_{UxQ}(n) = F_Q(z_U(k-L+1), \ldots, z_U(k))]$
Original Feature Vector **Z**
Texture Window **n= [k-L+1, k]**
Integration Functions **F**
Final Feature Vector **W**

Figure 1. Temporal feature integration.

While these measures have been widely used, providing robust performance, they only partly capture the temporal information of successive features. According to Reference [10], more metrics can be used to exploit the information that is hidden inside the time-series of the features. Some of them are used in this paper and presented below.

Mean Sequential Difference (MSD): Like the Standard Deviation (STD) measure, MSD aims to quantify the amount of variation of feature values inside a texture window, taking also into account the frequency of these changes. It is calculated as the Mean value of the summed up absolute differences of successive feature values, as defined in Equation (3).

$$X_{MSD}[n] = \frac{1}{L-2} \sum_{m=k-L+2}^{k} \left| x[m] - x[m-1] \right| \tag{3}$$

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, with respect to their Mean value (Equation (4)).

$$X_{MCR}[n] = \frac{1}{L-1} \sum_{m=k-L+1}^{k} 1_{R_{<1}}(d[m]), \tag{4}$$

where

$$d[m] = [x[m] - X_{MEA}[n]][x[m-1] - X_{MEA}[n]] \tag{5}$$

Flatness (FLA): Like Spectral Flatness, Temporal Flatness is calculated by dividing the geometric mean of the feature values by their arithmetic mean, inside a texture window, according to Equation (6).

$$X_{FLA}[n] = \frac{\sqrt[L]{\prod_{m=k-L+1}^{k} x[m]}}{\frac{\sum_{m=k-L+1}^{k} x[m]}{L}} \tag{6}$$

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

$$X_{CRF}[n] = \frac{MAX(x[k-L+1], \ldots, x[k])}{X_{MEA}[n]} \tag{7}$$

The statistical feature integration functions were implemented in VBA and MATLAB.

2.4. Feature Selection

As we mentioned earlier, the baseline feature vector for the Standard Frame Based (SFb) method has a dimension of 100. The feature vector for the Standard Temporal Integration (STi) method consists of the aggregated features representing the mean, standard deviation, skewness, and kurtosis of the following 12 baseline features: MFCCs, CEN, SPR, SKEW, KURT, ROL, ENTR, BRI, VAR, RMS, ZCR, and RDF, which results in a vector dimension of 96. A decision not to include all the extracted features for calculating the corresponding aggregated feature vector components was taken on the grounds that this would result in a significant increase of the final vector's dimension, while preliminary tests indicated low discriminative power for these attributes. Lastly, the Extended Temporal Integration (ETi) method extends the STi feature set by 96 additional features since it specifies four additional metrics (MSD, MCR, FLA, CRF). The feature vector dimensions for the three methodologies are summarized in Table 2. As we can notice, feature integration leads to high-dimensional vectors, which is an outcome that generally does not necessarily deliver better performance. High-dimensional feature vectors lead to data sparsity, which is a problem known as "the curse of dimensionality". A smaller feature set containing the most salient features, i.e., features that contribute more to the discrimination between the classes, is preferable since it reduces computational complexity, while it can also lead to performance improvements [23,24]. For this purpose, it is common practice to select an optimal subset of features out of the total set of the extracted ones. Several methodologies have been proposed for this purpose. In some approaches, the significance of each feature is estimated individually by calculating certain measures, while, in others, the candidate feature subsets are tested iteratively until classification performance is maximized [25]. In our study, feature ranking was carried out using the following ranking algorithms implemented in the RapidMiner Studio.

1. Information Gain Ratio (IGR): The algorithm calculates weights for each feature corresponding to their relevance to the class attribute by using the information gain ratio. The higher the weight of a feature, the more relevant it is considered.
2. Generalized Linear Model (GLM): The generalized linear model is an extension of ordinary linear regression. This algorithm fits generalized linear models to the data by maximizing the log-likelihood and calculates feature weights with respect to the class of each instance.
3. Support Vector Machine (SVM): The coefficients of a hyperplane calculated by an SVM are set as feature weights signifying the relevance of the class attribute.

Based on the ranking ratings and further experimentation, an optimal subset of features was selected for each method (SFb, STi, and ETi).

Table 2. Feature vector dimension for each method (before feature selection).

Method	SFb	STi	ETi
Dimension	100	96	192

2.5. Classification

The resulting feature sets of each method were tested with the following classification algorithms, utilizing the RapidMiner science platform [26] and Keras software library [27].

1. Logistic Regression (LR)
2. Artificial Neural Network (ANN)
3. Generalized Linear Model (GLM)

LR and GLM were deployed in RapidMiner under a default setup, while *ANN* was developed in Python, utilizing Keras with the following topology: Two hidden layers with (input + output dimensions)/2 neurons each, followed by a dropout layer. The rest of the parameters followed a typical setup: RELU activation function for the intermediate layers, SoftMax for the output layer, Categorical Cross-Entropy as the loss function, and Adam as the optimizer. The learning rate was set to 0.01 and the dropout to 25%, while a validation set was used to identify the parameter setting (epoch) that achieves the highest classification accuracy. We used 10% of the training samples as a validation set for identifying the training epoch that yields the best model parameters.

The performance evaluation was based on the measure of Accuracy. 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. It is reminded that each sound class in the database contains 18 different source files. For the classification, these files were split to 12 files for the training set and six for the testing set. This separation was performed in order to ensure that the algorithms are tested on previously unseen data, and guarantee unbiased and comparable results. If the split was generated randomly, we might have ended up with segments of the same source file being used for both training and testing, which would have led to artificially high classification accuracies. Lastly, in order to prevent overfitting to the test set, the process was repeated three times with different selections for the training and test sets (three-fold cross validation) and the results were averaged over the three iterations to give an estimate of the model's predictive performance.

3. Results and Discussion

In this section, the most significant experimental results are presented and interpreted. Table 3 demonstrates the results of feature ranking using the baseline feature set (SFb method). The features are ranked by the three ranking algorithms according to their discriminative power. Features that appear in the top ranking of all three methods are highlighted with gray, features that appear in two methods are highlighted with lighter gray, while features that appear only in one method are not highlighted. We can see that the results of the three different algorithms are in good agreement, since they all result in almost the same set of salient features.

Table 3. Top 20 features for the SFb method.

Rank Order	Ranking Algorithms		
	IGR	**GLM**	**SVM**
1	MFCC1	MFCC3	MFCC3
2	ENTR	MFCC1	MFCC2
3	BRI	MFCC2	ENTR
4	ROL	SFM15	MFCC1
5	MFCC2	MFCC4	BRI
6	CEN	ZCR	CEN
7	SKEW	VAR	VAR
8	SFM12	SFM18	MFCC4
9	KURT	SFM14	SMO
10	SFM10	ROU	ROL
11	ZCR	ROL	ZCR
12	SMO	SFM17	KURT
13	SFM18	SFM13	MFCC5
14	SFM11	SMO	SFM24
15	MFCC3	SCF7	SFM17
16	SFM13	ENTR	SFM18
17	SFM17	BRI	SFM14
18	VAR	SFM5	RDF
19	SFM19	SFM16	SFM13
20	SFM9	SFM12	SKEW

The same process was repeated for the ETi feature set in order to determine the most salient aggregated features. It is reminded that the ETi feature set is a superset of the STi feature set containing all the aggregated features. The results are shown in Table 4. A standard notation is followed for aggregated features, where subscripts symbolize the integration function and baseline text of the source feature (e.g., CEN_{STD} denotes the Standard Deviation of the Spectral Centroid). The source features of the aggregated features are the same as the baseline features, with some exceptions, like SFM that does not appear at the salient aggregated features, even though it performs well as a baseline feature. On the other hand, SPR, for example, is ranked high at the aggregated features, while it is not present at the salient baseline features. Regarding the integration functions, it is clear that MEA and STD are the top performers, while MSD, FLA, and MCR have very high rankings. It is important to note that SKEW and KURT do not appear at all in the top 20 features, which indicates that the new metrics are more relevant than those two standard statistic measures. These results are a preliminary indication that the ETi method has the potential to outperform the STi method.

Table 4. Top 20 features for temporal integration methods (STi and ETi).

Rank Order	**Ranking Algorithms**		
	IGR	**GLM**	**SVM**
1	MFCC1$_{\mathrm{MEA}}$	MFCC3$_{\mathrm{MEA}}$	MFCC3$_{\mathrm{MEA}}$
2	MFCC2$_{\mathrm{MEA}}$	MFCC2$_{\mathrm{MEA}}$	MFCC2$_{\mathrm{MEA}}$
3	ENTR$_{\mathrm{MEA}}$	MFCC1$_{\mathrm{MEA}}$	MFCC1$_{\mathrm{MEA}}$
4	MFCC10$_{\mathrm{STD}}$	VAR$_{\mathrm{MEA}}$	ENTR$_{\mathrm{MEA}}$
5	MFCC11$_{\mathrm{STD}}$	BRI$_{\mathrm{MEA}}$	BRI$_{\mathrm{MEA}}$
6	BRI$_{\mathrm{MEA}}$	SPR$_{\mathrm{MEA}}$	SPR$_{\mathrm{MSD}}$
7	MFCC12$_{\mathrm{STD}}$	SPR$_{\mathrm{MSD}}$	SPR$_{\mathrm{MEA}}$
8	MFCC9$_{\mathrm{STD}}$	RMS$_{\mathrm{MSD}}$	MFCC4$_{\mathrm{MEA}}$
9	ROL$_{\mathrm{MEA}}$	ENTR$_{\mathrm{MEA}}$	VAR$_{\mathrm{MEA}}$
10	CEN$_{\mathrm{MEA}}$	MFCC4$_{\mathrm{MEA}}$	ZCR$_{\mathrm{MEA}}$
11	MFCC8$_{\mathrm{STD}}$	VAR$_{\mathrm{FLA}}$	RMS$_{\mathrm{MSD}}$
12	MFCC7$_{\mathrm{STD}}$	MFCC1$_{\mathrm{MSD}}$	MFCC5$_{\mathrm{MEA}}$
13	SKEW$_{\mathrm{MEA}}$	ZCR$_{\mathrm{MEA}}$	MFCC1$_{\mathrm{MSD}}$
14	ROL$_{\mathrm{MSD}}$	RDF$_{\mathrm{MEA}}$	ROL$_{\mathrm{MSD}}$
15	KURT$_{\mathrm{STD}}$	ROL$_{\mathrm{MEA}}$	RDF$_{\mathrm{MEA}}$
16	ZCR$_{\mathrm{MEA}}$	ENTR$_{\mathrm{FLA}}$	VAR$_{\mathrm{FLA}}$
17	KUR$_{\mathrm{MEA}}$	MFCC7$_{\mathrm{MEA}}$	CEN$_{\mathrm{MSD}}$
18	MFCC6$_{\mathrm{STD}}$	MFCC12$_{\mathrm{MCR}}$	ZCR$_{\mathrm{MSD}}$
19	MFCC7$_{\mathrm{MEA}}$	SKEW$_{\mathrm{MEA}}$	MFCC6$_{\mathrm{MEA}}$
20	MFCC11$_{\mathrm{MCR}}$	CEN$_{\mathrm{MSD}}$	CEN$_{\mathrm{MEA}}$

Feature ranking allows us to identify which features should be prioritized in the selection of the optimal subset of features. However, we need to perform extra experiments in order to select the optimal dimension for each vector. This was decided after reviewing classification performance considerations with respect to different dimensions. Figure 2 shows the performance of the three feature sets in terms of accuracy, when tested with the ANN algorithm, with respect to the feature vector dimension. We notice a performance improvement from the SFb method to the STi method and another boost in performance, albeit smaller, when the ETi feature set is used. Based on this plot, the dimension of each feature set was selected as the dimension where each curve reaches maximum accuracy. A similar behavior was observed when the same test was performed with the other two classifiers, LR and GLM, which showed a peak in performance at the vector dimension of 55 and 60 features, respectively. Table 5 shows the resulting vector dimensions for each feature set after feature selection, evaluated with each classification algorithm. It can be observed that ANN and GLM require a higher dimension feature vector than LR to reach their peak performance. However, the differences in the resulting dimensions are not significant.

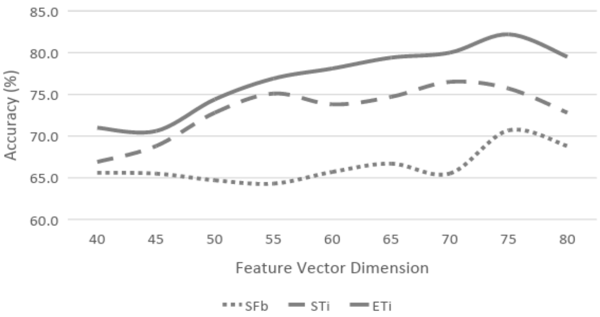

Figure 2. Classification accuracy of the ANN algorithm with respect to the number of features used for the three methods.

Table 5. Feature vector dimension for each method (after feature selection) evaluated with three classifiers.

	LR	ANN	GLM
SFb	40	75	70
STi	60	70	65
ETi	55	75	65

The classification results of the resulting feature sets when tested with the respective classification algorithms are shown in Table 6. Similar conclusions as before can arise. The STi feature set outperforms the SFb feature set, which demonstrates that the process of temporal feature integration with simple statistics can improve performance in environmental sound classification tasks. Furthermore, the ETi feature set, which extends the statistical measures, brings further improvements. This proves the discriminative power of the proposed measures. The top performing classifiers with the ETi feature set are ANN and GLM, with ANN scoring a bit higher, while, for LR, we do not observe a difference in performance between the STi and ETi methods. Nevertheless, time integration brings a considerable boost in performance to this method as well, when compared to the SFB method. The classification accuracy of the best performing setup (ETi combined with ANN) is very promising (81.5%) for environmental sound related applications (sound source identification, indoor/outdoor soundscape semantics etc.). Moreover, the recognition task is evaluated in time frames. In most real world cases of Environmental Sound Recognition tasks, the duration of a soundscape is more than a few seconds. Thus, time integration, which implements a voting scheme across multiple time frames, is expected to improve accuracy results.

Table 6. Classification ratings (Accuracy).

	LR	ANN	GLM
SFb	60.4%	69.8%	69.3%
STi	73.6%	75.2%	75.0%
ETi	74.8%	81.5%	80.4%

In Table 7, we also present the relative difference in performance between the 10 classes of the top performing setup (ETi combined with ANN) in terms of precision and recall. The recognition rates are acceptable for every class with relatively small variations.

Table 7. Classification ratings per class (Precision & Recall).

	Airplane	Alarms	Applause	Birds	Dogs	m/Cycles	Rain	Rivers	Sea Waves	Thunders
Precision	84.8%	97.5%	79.6%	87.8%	83.4%	88.9%	73.0%	90.7%	75.6%	82.0%
Recall	76.7%	86.6%	93.4%	98.9%	79.1%	74.4%	78.0%	75.6%	74.9%	98.5%

In order to compare the performance of the proposed method against previously published approaches, we performed an additional evaluation of our best performing setups (STi and ETi combined with ANN with vector size of 75 features) on the UrbanSound8K dataset [28], which has been utilized by several authors for similar classification purposes. This dataset contains 8732 labeled sound clips taken from field recordings (8.75 h), divided into 10 environmental sound categories: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The methods that are evaluated on the same dataset (without data augmentation) include a dictionary learning approach (spherical k-means, SKM) presented in Reference [29] and two convolutional neural network (CNN) methods (PiczakCNN and SB-CNN) presented in Reference [17,30]. Table 8 shows the classification results of the previously mentioned methods. We can see that the proposed method (ETi-ANN) delivers comparable performance to the state-of-the-art approaches, even though it requires fewer computational resources than the more demanding deep learning methods. It is also evident that the proposed metrics of the ETi method outperform the simple statistical metrics of the STi method.

Table 8. Classification accuracy comparison between the proposed methods ETi-ANN and other methods on the dataset Urbansound8K.

ETi-ANN	72%
STi-ANN	69%
SKM	74%
PiczakCNN	73%
SB-CNN	73%

4. Conclusions and Future Work

A new method for temporal feature integration introducing a set of robust and lightweight measures that supplement the common statistical measures is tested for its effectiveness in environmental sound recognition tasks. The experiments are conducted on a publicly available database of environmental sounds that the authors created. The evaluation methodology is described in detail and the results reveal promising performance for the new metrics. The proposed method ETi combined with the classifiers ANN or GLM results in more than 80% classification accuracy, which significantly outperforms the framed-based approaches and the methods utilizing simple statistical integration. Furthermore, a comparative evaluation against state-of-the-art algorithms and deep learning methodologies was conducted on the dataset Urbansound8k, which reveals that the proposed method achieves similar performance to more computationally expensive approaches. As far as the applicability of the method in real-world scenarios and in the presence of computational constraints is concerned, a trade-off between complexity and performance is intended. Therefore, the results of the evaluation of the method against deep learning techniques are considered satisfactory and suggest that the proposed method can be a good alternative for applications where computational efficiency is a priority.

A future direction of the study could be an exhaustive comparison of all early and late integration methods in order to highlight the best performing approaches or combinations of them. Lastly, taking into consideration that the main objective of this research is to evaluate the advantages of time integration against baseline feature extraction, a reasonable idea would be to further experiment with different baseline features (e.g., wavelets) [31], in order to investigate how they function within the time integration approach. It would be of great interest on Indoor Soundscaping to adapt enhanced temporal integration for aiding wavelet-based approaches that have been successfully applied for utilizing multi-band-/scale and multi-resolution activity detection, accompanied with the extraction of temporal, spectral, and cepstral features [32,33].

Author Contributions: Conceptualization, V.B. Methodology, V.B. and L.V. Investigation, L.V. Data curation, V.B. and N.V. Writing—original draft preparation, V.B. Writing—review and editing, L.V., K.K., and N.V. Supervision, K.K.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chachada, S.; Kuo, C.C. Environmental sound recognition: A survey. In Proceedings of the APSIPA Transactions on Signal and Information Processing, Kaohsiung, Taiwan, 29 October–1 November 2013.
2. Vrysis, L.; Dimoulas, C.; Kalliris, G.; Papanikolaou, G. Mobile Audio Measurements Platform: Toward Audio Semantic Intelligence into Ubiquitous Computing Environments. In *Audio Engineering Society Convention 134*; Audio Engineering Society: New York, NY, USA, 2013.
3. Ntalampiras, S.; Potamitis, I.; Fakotakis, N. Automatic recognition of urban environmental sounds events. In Proceedings of the International Association for Pattern Recognition Workshop on Cognitive Information Processing, Santorini, Greece, 9–10 June 2008; pp. 110–113.
4. Serizel, R.; Bisot, V.; Essid, S.; Richard, G. Acoustic Features for Environmental Sound Analysis. In *Computational Analysis of Sound Scenes and Events*; Springer: Cham, Switzerland, 2018; pp. 71–101.
5. Bountourakis, V.; Vrysis, L.; Papanikolaou, G. Machine learning algorithms for environmental sound recognition: Towards soundscape semantics. In Proceedings of the Audio Mostly 2015 on Interaction with Sound, Thessaloniki, Greece, 7–9 October 2015; p. 5.
6. Chu, S.; Narayanan, S.; Kuo, C.C. Environmental sound recognition with time–frequency audio features. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 1142–1158. [CrossRef]
7. Peeters, G. A Large Set of Audio Features for Sound Description (Similarity and Classification). CUIDADO Project Ircam Technical Report, 2004. Available online: http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf (accessed on 4 April 2019).
8. Cowling, M.; Sitte, R. Comparison of techniques for environmental sound recognition. *Pattern Recognit. Lett.* **2003**, *24*, 2895–2907. [CrossRef]
9. Flocon-Cholet, J.; Faure, J.; Guérin, A.; Scalart, P. An investigation of temporal feature integration for a low-latency classification with application to speech/music/mix classification. In *Audio Engineering Society Convention 137*; Audio Engineering Society: New York, NY, USA, 2014.
10. Vrysis, L.; Tsipas, N.; Dimoulas, C.; Papanikolaou, G. Extending Temporal Feature Integration for Semantic Audio Analysis. In *Audio Engineering Society Convention 142*; Audio Engineering Society: New York, NY, USA, 2017.
11. Meng, A.; Ahrendt, P.; Larsen, J.; Hansen, L.K. Temporal feature integration for music genre classification. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1654–1664. [CrossRef]
12. McKinney, M.; Breebaart, J. Features for audio and music classification. In Proceedings of the International Symposium on Music Information Retrieval, Baltimore, MD, USA, 27–30 October 2003; pp. 151–158.
13. Joder, C.; Essid, S.; Richard, G. Temporal integration for audio classification with application to musical instrument classification. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 174–186. [CrossRef]
14. Ntalampiras, S.; Potamitis, I.; Fakotakis, N. Exploiting temporal feature integration for generalized sound recognition. *EURASIP J. Adv. Signal Process.* **2009**, *2009*, 807162. [CrossRef]
15. Pillos, A.; Alghamidi, K.; Alzamel, N.; Pavlov, V.; Machanavajhala, S. A real-time environmental sound recognition system for the Android OS. In Proceedings of the Detection and Classification of Acoustic Scenes and Events, Budapest, Hungary, 3 September 2016.

16. Tsipas, N.; Zapartas, P.; Vrysis, L.; Dimoulas, C. Augmenting social multimedia semantic interaction through audio-enhanced web-tv services. In Proceedings of the Audio Mostly 2015 on Interaction with Sound, Thessaloniki, Greece, 7–9 October 2015; p. 34.
17. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [CrossRef]
18. SoX—Sound eXchange. Available online: http://www.webcitation.org/74exgvSEq (accessed on 14 December 2018).
19. Lartillot, O.; Toiviainen, P.; Eerola, T. A matlab toolbox for music information retrieval. In *Data Analysis, Machine Learning and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 261–268.
20. Tzanetakis, G.; Cook, P. MARSYAS: A framework for audio analysis. *Organ. Sound* **2000**, *4*, 169–175. [CrossRef]
21. McKay, C.; Fujinaga, I.; Depalle, P. jAudio: A feature extraction library. In Proceedings of the International Conference on Music Information Retrieval, London, UK, 11–15 September 2005; pp. 600–603.
22. Tsipas, N.; Vrysis, L.; Dimoulas, C.; Papanikolaou, G. Mirex 2015: Methods for speech/music detection and classification. In Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX), Malaga, Spain, 26–30 October 2015.
23. Mitrović, D.; Zeppelzauer, M.; Eidenberger, H. On feature selection in environmental sound recognition. In Proceedings of the International Symposium (ELMAR'09), Zadar, Croatia, 28–30 September 2009; pp. 201–204.
24. Kotsakis, R.; Kalliris, G.; Dimoulas, C. Investigation of salient audio-features for pattern-based semantic content analysis of radio productions. In *Audio Engineering Society Convention 132*; Audio Engineering Society: New York, NY, USA, 2012.
25. Schowe, B. Feature selection for high-dimensional data with RapidMiner. In Proceedings of the 2nd RapidMiner Community Meeting and Conference (RCOMM 2011), Dublin, Ireland, 7–10 June 2011.
26. Goyal, V.K. A comparative study of classification methods in data mining using rapidminer studio. *Int. J. Innov. Res. Sci. Eng. ISSN (Online)* **2013**, 2347–3207.
27. Chollet, F. *Deep Learning with Python*; Manning Publications Co.: Shelter Island, NY, USA, 2017.
28. Salamon, J.; Jacoby, C.; Bello, J.P. A dataset and taxonomy for urban sound research. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 1041–1044.
29. Salamon, J.; Bello, J.P. Unsupervised feature learning for urban sound classification. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 171–175.
30. Piczak, K.J. Environmental sound classification with convolutional neural networks. In Proceedings of the 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), Boston, MA, USA, 17–20 September 2015; pp. 1–6.
31. Dimoulas, C.; Kalliris, G. Investigation of wavelet approaches for joint temporal, spectral and cepstral features in audio semantics. In Proceedings of the 134th AES Convention, Rome, Italy, 4–7 May 2013; pp. 509–518.
32. Vegiris, C.; Dimoulas, C.; Papanikolaou, G. Audio Content Annotation, Description and Management Using Joint Audio Detection, Segmentation and Classification Techniques. In Proceedings of the 126th AES Convention, Munich, Germany, 7–10 May 2009. Paper No. 7661.
33. Dimoulas, C.; Vegiris, C.; Avdelidis, K.; Kalliris, G.; Papanikolaou, G. Automated Audio Detection, Segmentation, and Indexing with Application to Postproduction Editing. In Proceedings of the 122nd AES Convention, Vienna, Austria, 5–8 May 2007. Paper No. 7138.