The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

A Resampling Approach for Imbalanceness on Music Genre Classification Using Spectrograms

Vinicius D. Valerio,[1] **Rodolfo M. Pereira,**[2,4] **Yandre M. G. Costa,**[1]
Diego Bertolini,[3] **Carlos N. Silla Jr.**[2]
[1]Graduate Program in Computer Science, State University of Maringa (UEM), Maringa, PR, Brazil
[2]Computer Music Technology Laboratory, Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil
[3]Academic Department of Computation, Federal University of Technology - Parana (UTFPR), Campo Mourao, PR, Brazil
[4]Federal Institute of Education, Science and Technology of Parana - Campus Pinhais (IFPR), Pinhais, PR, Brazil
E-mail: {vinidiasvalerio, rodolfomp123, yandre.costa, carlos.sillajr, diegobertolini}@gmail.com

Abstract

In real-world problems, modeled as machine learning tasks, the datasets are typically unbalanced, meaning that some classes have much more instances than others. In the Music Information Retrieval field it is not different and songs datasets usually are very unbalanced. Considering this scenario, we propose a novel approach to face the class imbalance problem applied to music genre classification. The proposed method uses vertical sliced spectrograms extracted from the songs' audio signal to apply oversampling and undersampling into the minority and majority classes, respectively. The experimental results for F-Score measure showed that our approach was able to beat the best result of Random Undersampling technique by 0.086, using MultiLayer Perceptrons. Besides, comparing to the baseline results, our approach significantly increased the individual results for all the minority classes.

Introduction

In the Machine Learning (ML) field, in order to build a prediction model capable to correct classify instances from all the range of classes, a balanced dataset is desirable. The most balanced are the classes, the greater are the chances that the classifier will learn patterns from all the instances and construct a representative model. However, a lot of researchers face imbalanced class distribution issues, mainly when working with datasets extracted from real world problems. As shown in Chawla, Japkowicz, and Kolcz (2004), the imbalanceness affects the generalization capability of the classification method, making the classifier focus on the global accuracy and leading it to the correct classification of instances only from the majority classes.

According to Zheng, Cai, and Li (2016), there are two kinds of solutions to unbalance problems: data-level and algorithm-level methods. While the first one is applied over the dataset in order to adjust the distribution of classes, the second is focused in the development of new algorithms or the improvement of existing ones, targeting to increase the recognition ratio of the minority classes. The first solutions, also known as resampling, is the most common and widely used in the research community. The resampling methods balance the dataset by creating new instances for the minority classes (oversampling) and/or removing existent samples from the majority classes (undersampling), using a predefined criteria.

As nowadays music is a prevalent topic in society, it is reasonable that the research field of Music Information Retrieval (MIR), which deals with a wide range of computational tasks related to music, keeps growing. Therefore, the automatic classification of music pieces into genres is a task that deserves attention. However, such as in other real world problems, the number of songs from each genre uses to be very different, i.e, songs datasets are frequently quite unbalanced.

Considering this context, we propose an alternative and novel approach to deal with the class imbalance problem in music genre classification using specific information from the musical domain. In our method, first of all, the songs' audio signal are converted into spectrograms (a short-time Fourier visual representation of the music (Costa et al., 2012)). Next, the songs' spectrograms are divided into vertical slices (each one represents 30 seconds of audio signal) and two steps are applied: an oversampling over the minority classes and an undersampling over the majority classes. While the oversampling step separate and associate each spectrogram slice as a different song piece from the same genre, the undersampling step only considers the spectrogram middle slice, discarding the other slices. Finally, in order to classify the songs, visual features are extracted from the spectrogram slices.

In this work, in order to generate the vectors of characteristics used in the classification, we extracted LBP features (Local Binary Pattern) from the spectrograms. These features are structural approaches to texture descriptors, introduced by Ojala, Pietikainen, and Maenpaa (2002).

This paper is organized as follows: The second section presents a brief theoretical background for this work, being subdivided in Music Information Retrieval, Visual Features and Local Binary Pattern. The third section shows a discussion about related works. While the forth section intends to present the proposed approach, the fifth section shows the experimental evaluation of this method, such as the results

and discussions. Finally, in the sixth section, we show the conclusions and possible future direction for this work.

Theoretical Background

This section intends to give a brief background about concepts used in this work, such as Music Information Retrieval, Visual Features and Local Binary Pattern.

Music Information Retrieval

Music Information Retrieval (MIR) (Downie, 2003) is an interdisciplinary area that aims at analysis and description of musical data, allowing to retrieve information concerning the digital music files. Over the last decade, several researches have been carried out in this area for different purposes. As some examples, we can mention the automatic recognition of music genres (Nanni et al., 2016) and automatic music recommendation (Van den Oord, Dieleman, and Schrauwen, 2013). Thus, automatic classification of musical genres has become an important task within the musical and computational contexts.

Although they do not have an absolute definition, musical genres are considered classes in which the songs share similar features to each other, such as rhythmic patterns, instruments and chords (Costa et al., 2012). Since sometimes not even humans are able to distinguish some musical genres and accurate information about these genres may be vague, implementing an ideal system for automatic classification is a challenging task (Lippens, Martens, and De Mulder, 2004). Furthermore, the classification of musical pieces into genres is a time consuming task, which makes it a perfect job for a computer.

Visual Features

Although most MIR papers use audio-based features to classify the songs, the visual-based features have shown competitive or even better results in the last years (Costa et al., 2011, 2012; Nanni et al., 2017). In this work particularly, we use a visual representation known as spectrogram, a time-frequency image created from the audio signal, i.e., the horizontal axis of the generated image represents the song‘s time, while the vertical axis represent the song‘s frequency. Furthermore, the color intensity of the pixel is the amplitude of the audio signal.

In other words, a spectrogram graphically represents the data relation of an audio signal in time and frequency domains. This kind of visual feature has been successfully used for music classification purposes since 2011 (Costa et al., 2011), and has been recently combined with audio-based features, proving its complementarity (Lucio and Costa, 2016; Nanni et al., 2017). Figure 1 shows an example that represents the process to generate a spectrogram image from an audio signal.

Most of the image processing techniques employed for MIR purposes operates on gray level images, thus it is convenient to convert the spectrogram image to gray levels, such as shown in Figure 1. Moreover, it is also important to state that, for the genre classification purpose, the most important information present in the spectrograms concerns the energy

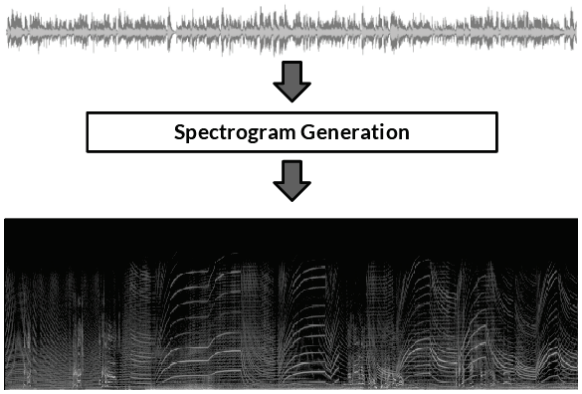


Figure 1: Spectrogram image extracted from an MP3 audio sample. Music title: Crash; artist: Blah Blah Blah; genre label: Rock.

intensity of the audio signal. Thus, when a color scale image is converted to grayscale, this information is often fully preserved (Costa et al., 2012).

Local Binary Pattern (LBP)

The LBP feature was first proposed by Ojala, Pietikainen, and Maenpaa (2002) and is one of the most successful texture descriptors used to describe spectrogram content. This method has a low computational complexity and it allows local analysis of the information (Nanni, Lumini, and Brahnam, 2012). To generate the LBP feature set, first each pixels of the image and its adjacent pixels (neighborhood) are selected, and then the method finds a histogram of the local binary patterns, which can be used as a texture descriptor.

Figure 2 shows that we can create variations of LBP patterns, establishing different amounts of neighboring pixels for their operation. These variations are important for the descriptor to be able to operate on textures of different scales. Thus, such variations are identified as $LBP_{P,R}$, in which P represents the amount of neighboring pixels existing in a circle of R, around the central pixel.

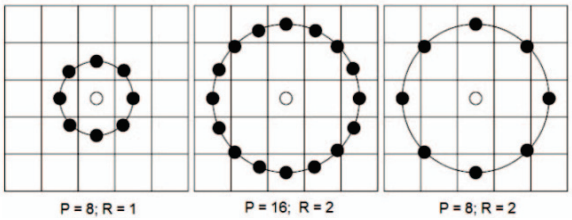


Figure 2: Variations of the LBP descriptor (Ojala, Pietikäinen, and Harwood, 1996).

In this paper we used $LBP_{8,2}$. Using this pattern, the values obtained for each of the eight neighbors are concatenated, generating a binary number that is converted to the decimal base. Next, the value of the central pixel is replaced by the decimal base, which was calculated using its neighborhood. As final result we have a histogram containing 59

values that represents the visual-based feature set for the specific audio signal. These features represent the intensity of the pixels, which describe the texture of the image.

Related Works

In this section we present some related work concerning resampling methods. In the last years, a lot of methods have been proposed. Between the first solutions there are Random Oversamping (ROS) and Random Undersampling (RUS), which randomly create/remove instances from the dataset.

Moreover, Chawla et al. (2002) presented SMOTE (Synthetic Minority Oversampling Technique), a method that creates synthetic samples for the minority classes based on its neighbors. Although SMOTE is the most popular data-level oversampling method and has being widely used by the research community, in Yen and Lee (2009) is shown that it may cause over-generalization, which reinforce the community needs for new resampling methods.

Following this context, the works of Chan and Stolfo (1998) and Liu, Wu, and Zhou (2009) addressed the imbalance problem by combining classifiers built from multiple resampled datasets. In their approach, they created subsets from the majority class containing the same amount of samples from the minority class. Next, they trained a classifier using each one of these subsets and then combined the results.

Furthermore, Tahir, Kittler, and Yan (2012) proposed a novel undersampling method introduced as Inverse Random Undersampling (IRUS). The main idea behind this approach is to severely apply undersampling in the majority class until creating a large number of distinct training sets. Then, for each one of the training sets, it finds a decision boundary which separates the minority class from the majority class. In the end, the multiple designs are fused to improve the classification.

It is important to note that, while the existing resampling methods from the literature are generic (such as the ones stated in the previous paragraphs), to the best of our knowledge, the technique proposed in this paper is the first one to deal with the imbalaceness problem using specific characteristics from the music domain.

Proposed Approach

In this section, we describe the steps towards the creation of the feature vectors, in which we use the proposed resampling method to balance the dataset.

As stated before, our approach is based on a visual representation of the audio signal known as spectrogram. Thus, the first step of the feature extraction process consists into convert the audio signal to a grayscale spectrogram[1]. After the grayscale spectrograms are generated, we propose to combine oversampling and undersampling techniques to balance the classes' distribution. Such as in most works involving unbalanced datasets, in this work the oversampling technique is applied on the minority classes and the undersampling on the majority classes.

In both resampling techniques, before extract the feature vectors from the full-length spectrograms, we used a zoning mechanism that divides the spectrogram images into "vertical slices", in which each of these slices represents 30 seconds of the audio signal.

As shown in Figure 3(a), to generate the new "samples", the oversampling step takes advantage of the vertical slices created in the spectrogram. Considering these slices as different instances, the feature vectors are extracted from each one of them individually. It is important to observe that the final remaining slice (which usually has less than 30 seconds) is discarded.

For the undersampling step, as shown in Figure 3(b), after applying the zoning in the sample, only the central slice of the spectrogram is considered. In other words, this means that even though the zoning produce several slices in the spectrogram, we only consider one vertical slice and discard the other ones.

In the final step, in order to complete the undersampling phase, we randomly discard instances from the majority classes until all classes from the dataset are equally distributed.

It is important to note that in many works from the literature, such as in Costa et al. (2011, 2012); Nanni et al. (2016, 2017), the authors use only a 30 seconds piece from the songs (even when the full-length audio is available), discarding the remaining audio signal. However, our approach proposes to use the remaining audio signal to balance the dataset in the oversampling step.

Experimental Evaluation

In this section we present experimental results for the proposed approach. The experiments were performed using six well consolidated classification algorithms from the literature: C4.5, k-Nearest Neighbors (kNN), Multilayer Perceptron (MLP), Naive Bayes (NB), Random Forest (RF) and Support Vector Machines (SVM). All tests were executed with the WEKA data mining tool version 3.9 (Hall et al., 2009).

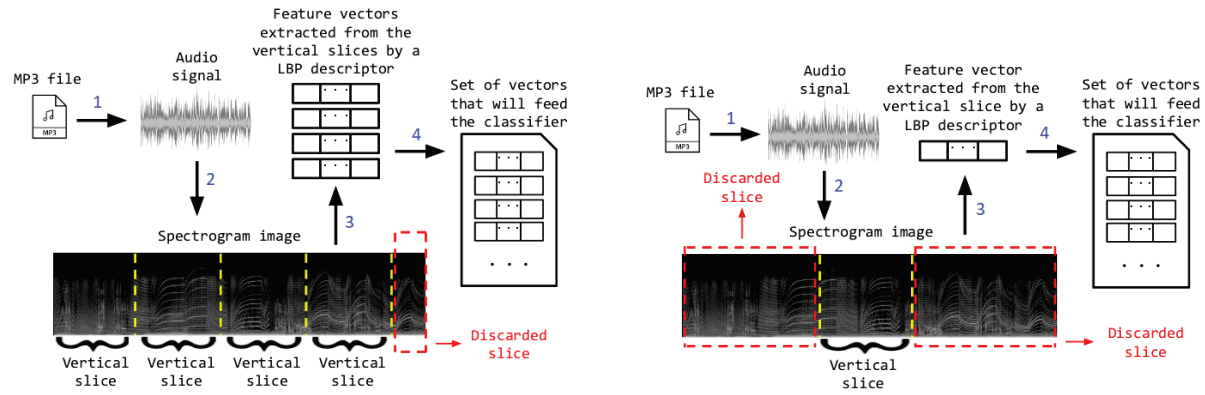
Parameters and Configuration

Table 1 reports the parameter settings used by the algorithms executed in this work. All experiments were conducted using a five-fold cross-validation scheme.

It is important to note that the folds were created using the *artist filter* technique, introduced by Pampalk et al. (2005), in which all songs from the same artist are placed into the same fold, i.e., there will be no songs from the same artist in more than one fold, and not even pieces of the same song. The rationale behind this is that we are trying to create a classifier system able to perform genre classification, instead of artist classification, i.e, if *artist filter* technique is not used there is a possibility that the classifier will learn patterns from artists rather than musical genres.

Its important to observe that, using the oversampling methods from the literature, such as Random Oversampling

[1]The spectrograms were generated with the Sound eXchange (SoX 14.3.0) software, which is available for download at sox.sourceforge.net



(a) Oversampling. (b) Undersampling.

Figure 3: Proposed resampling method.

Table 1: Parameter settings of the algorithms.

Algorithm	Parameters	
C4.5	Confidence Factor	0.25
	Min. Instances per Leaf	2
kNN	Number of Neighbours	1
MLP	Learning Rate	0.3
	Momentum	0.2
	Hidden Layers	(#attr + #class)/2
	Number of Epochs	500
	Validation Threshold	20
NB	Use Kernel Estimator	No
	Use Supervised Discretization	No
RF	Number of Trees	10
	Max Depth	0
	Base Classifier	Bagging
SVM	Kernel	Polynomial
	Complexity Constant	1

(ROS) and SMOTE, its practically impossible to apply the *artist filter* technique to create the folds, since the new samples may not be assigned to a specific artist. Thus, this reinforce the novelty of our proposed resample approach, in which even making a oversampling step we can still use *artist filter*.

The Dataset

The experiments were conducted on a single-label subset extracted from the Free Music Archive (FMA) website. The extraction of songs from this repository was first introduced by Benzi et al. (2017). FMA is an online library[2] of audio and video content that the public, including music fans, webcasters, and podcasters, may listen to, download, or stream for free. The current full dataset version, available for download[3] is composed of 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a multi-label taxonomy of its 161 genres.

[2]www.freemusicarchive.org

[3]github.com/mdeff/fma

Due to the fact that we are concerned specifically with single-label classification in this work, we created a single-label subset from the full FMA dataset. This subset considers only the songs from the full dataset with one genre. Besides, we grouped the sub-genres with the same genre label, i.e, sub-genres such as indie-rock and psych-rock were considered as rock, and so on. The subset is composed of 11 genres with the following number of songs each: Blues (114), Classical (790), Country (123), Electronic (3533), Folk (1876), Hip-Hop (2852), Jazz (432), Old-Time/Historic (523), Pop (1411), Rock (6255) and Soul-RnB (136).

Table 2: Dataset Main Characteristics.

#Songs	18,045
#Genres	11
#Albums	2,987
#Artists	3,680
Minimum Duration	3s
Maximum Duration	10m
Average Duration	03m33s
Total Duration	44 days 11h52m53s

Table 2 shows the main characteristics of the subset. We may observe how challenging is the dataset, since we have 18,045 songs divided into 11 genres. Even considering only the single-label portion of FMA, this subset is one of the biggest and unbalanced musical datasets in the literature so far. In contrast, for example, we can mention GTZAN (Tzanetakis and Cook, 2002), one of the most used musical datasets, in which its only 1,000 songs are equally distributed between 10 genres.

In order to allow experimental comparison by the research community, we made available for download the files used in the experiments[4].

Results and Discussion

All experiments were evaluated using the well-known and consolidate *F-Score* metric. While Table 3 shows the base-

[4]sites.google.com/view/fma-sl-grouped

line results for the dataset, i.e, without the application of resampling methods, Table 4 shows the experimental results after applying Random Undersampling (RUS) and our proposed method in the dataset to balance the classes.

The main observation that can be made is concerning the best average result. Looking at Tables 3 and 4, we may note that the best result (0.622) was achieved using MLP over the baseline dataset, i.e, without any resampling method. This may raise an important question: If the best result was obtained with the original dataset, why should we apply any resampling technique?

First of all, the best result is the weighted average of the results individually obtained from each class. Taking a closer look at Table 3, we may note that for all minority classes (Blues, Country, Jazz and Soul-RnB) MLP deeply failed to classify its instances, with low results (0.074, 0.015, 0.109 and 0.052). Furthermore, the same behavior happens with all algorithms, being highly present in SVM, which scored zero for the minority classes.

On the other hand, the macro-average results, which uses a class-based average instead of an instance-based, shows that the best result was in fact obtained with our proposed technique (0.518). Moreover, looking at Table 4, we may note that our proposed approach not only improved the individual results for the minority classes (Blues: 0.520, Country: 0.599, Jazz: 0.328 and Soul-RnB: 0.523), but also beat RUS method for all algorithms.

Thus, we are now able to answer the previous question: Because the resampling methods, in particular our proposed technique, make the classifiers learn better models to represent and predict all classes, improving the individual results per class.

The second consideration that can be made is related to the Pop class. We may note that Pop was practically not affected by the resampling techniques, since the algorithms provided poor results before and after the resampling. The explanation for this particular case is based in the fact that Pop is a "generic" genre. In terms of genre classification, this means that it is hard for the classifiers to learn a common pattern from Pop songs.

Table 3: Baseline experimental results for F-Score.

Genre	C4.5	kNN	MLP	NB	RF	SVM
Blues	.044	.208	.074	.015	.028	.000
Classical	.673	.732	.739	.490	.750	.685
Country	.062	.246	.015	.037	.099	.000
Electronic	.477	.476	.630	.204	.570	.614
Folk	.404	.475	.539	.272	.516	.519
Hip-Hop	.627	.678	.745	.595	.707	.731
Jazz	.106	.162	.109	.012	.122	.000
Old-Time	.776	.887	.891	.377	.854	.823
Pop	.143	.196	.121	.046	.121	.003
Rock	.642	.679	.731	.592	.708	.724
Soul-RnB	.010	.085	.052	.036	.014	.000
Weig. Avg.	.523	.566	**.622**	.404	.593	.595
Macro Avg.	.360	**.439**	.422	.243	.408	.373

Conclusion and Future Works

In this paper, we have presented an alternative approach to address imbalanceness in songs datasets for Music Classification problems. In our approach, first we convert the audio signal representation into spectrograms, which are then divide into vertical zones. Secondly, oversampling and undersampling steps are applied in the spectrograms in order to create/remove samples pieces from the minority/majority classes, generating a balanced subset.

We show that our approach achieves better results than employing a Random Undersampling technique to arrange the classes distribution. Although our best weighted average result (0.518 with MLP classifier) did not beat the best baseline result (0.622 also with MLP), the proposed method considerably increased the individual results for all the minority classes.

It is important to observe that even though we have tested our proposed approach in a music genre classification context, it may also be used in any type of audio classification task, such as mood, acoustic scene, composer, and so on.

As future work we suggest the use of other visual features to generate the feature vectors from the spectrograms. We also propose the use of fusion approaches (early fusion and late fusion) to combine the visual-base features with audio-base features in order to improve the results. We also intend to develop a new strategy that combines different musical pieces in order to generate synthetic samples for the minority classes.

Acknowledgments

We thank the Brazilian Research Support Agencies: CAPES - Coordination for the Improvement of Higher Education Personnel, CNPq - National Council for Scientific and Technological Development and FA - Araucaria Foundation for their financial support.

References

Benzi, K.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2017. FMA: A dataset for music analysis. In *Proceedings of the International Society on Music Information Retrieval Conference*, 316–323.

Chan, P. K., and Stolfo, S. J. 1998. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 164–168.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16:321–357.

Chawla, N. V.; Japkowicz, N.; and Kolcz, A. 2004. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* 6(1):1–6.

Costa, Y. M. G.; Oliveira, L. E. S.; Koerich, A. L.; and Gouyon, F. 2011. Music genre recognition using spectrograms. In *Proceedings of the International Conference on Systems, Signals and Image Processing*, 151–154.

Table 4: Experimental results for F-Score with resampling techniques.

	RUS Algorithm						Proposed Approach					
Genre	**C4.5**	**kNN**	**MLP**	**NB**	**RF**	**SVM**	**C4.5**	**kNN**	**MLP**	**NB**	**RF**	**SVM**
Blues	.281	.417	.264	.060	.392	.212	.342	.628	.520	.204	.481	.325
Classical	.626	.702	.697	.593	.723	.685	.682	.741	.765	.640	.750	.700
Country	.261	.396	.383	.200	.303	.314	.456	.647	.599	.312	.512	.431
Electronic	.229	.142	.390	.196	.213	.330	.317	.360	.445	.330	.379	.445
Folk	.245	.287	.361	.235	.281	.360	.229	.359	.379	.172	.328	.298
Hip-Hop	.512	.511	.534	.509	.551	.605	.470	.519	.596	.458	.532	.583
Jazz	.228	.265	.282	.017	.256	.307	.214	.313	.328	.026	.309	.236
Old-Time	.835	.915	.881	.645	.835	.830	.794	.872	.894	.561	.837	.811
Pop	.137	.183	.231	.095	.191	.207	.155	.226	.174	.040	.202	.151
Rock	.272	.328	.395	.337	.418	.464	.328	.382	.475	.354	.407	.488
Soul-RnB	.283	.267	.336	.256	.274	.438	.381	.550	.523	.219	.468	.402
Weig. Avg.	.355	.401	**.432**	.286	.403	.432	.397	.509	**.518**	.301	.473	.443
Macro Avg.	.355	.401	**.432**	.286	.403	.432	.397	.509	**.518**	.301	.473	.443

Costa, Y. M. G.; Oliveira, L.; Koerich, A. L.; Gouyon, F.; and Martins, J. 2012. Music genre classification using LBP textural features. *Signal Processing* 92(11):2723–2737.

Downie, J. S. 2003. Music information retrieval. *Annual Review of Information Science and Technology* 37(1):295–340.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11(1):10–18.

Lippens, S.; Martens, J.-P.; and De Mulder, T. 2004. A comparison of human and automatic musical genre classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, 233–236.

Liu, X.-Y.; Wu, J.; and Zhou, Z.-H. 2009. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39(2):539–550.

Lucio, D. R., and Costa, Y. M. G. 2016. Bird species classification using visual and acoustic features extracted from audio signal. In *Proceedings of The International Conference of the Chilean Computer Science Society*, 1–12.

Nanni, L.; Costa, Y. M. G.; Lumini, A.; Kim, M. Y.; and Baek, S. R. 2016. Combining visual and acoustic features for music genre classification. *Expert Systems with Applications* 45:108–117.

Nanni, L.; Costa, Y. M. G.; Lucio, D. R.; Silla Jr., C. N.; and Brahnam, S. 2017. Combining visual and acoustic features for audio classification tasks. *Pattern Recognition Letters* 88:49–56.

Nanni, L.; Lumini, A.; and Brahnam, S. 2012. Survey on LBP based texture descriptors for image classification. *Expert Systems with Applications* 39(3):3634–3641.

Ojala, T.; Pietikäinen, M.; and Harwood, D. 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* 29:51–59.

Ojala, T.; Pietikainen, M.; and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(7):971–987.

Pampalk, E.; Flexer, A.; Widmer, G.; et al. 2005. Improvements of audio-based music similarity and genre classificaton. In *Proceedings of the International Society on Music Information Retrieval Conference*, 634–637.

Tahir, M. A.; Kittler, J.; and Yan, F. 2012. Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition* 45(10):3738–3750.

Tzanetakis, G., and Cook, P. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10(5):293–302.

Van den Oord, A.; Dieleman, S.; and Schrauwen, B. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*, 2643–2651.

Yen, S.-J., and Lee, Y.-S. 2009. Cluster-based undersampling approaches for imbalanced data distributions. *Expert Systems with Applications* 36(3):5718–5727.

Zheng, Z.; Cai, Y.; and Li, Y. 2016. Oversampling method for imbalanced classification. *Journal of Computing and Informatics* 34(5):1017–1037.