Automatic Music Genre Classification in Small and Ethnic Datasets

Tiago Fernandes Tavares[1] * and Juliano Henrique Foleiss[1]

School of Electric and Computer Engineering – University of Campinas – Brazil
tavares@dca.fee.unicamp.br

Abstract. Automatic music genre classification commonly relies on a large amount of well-recorded data for model fitting. These conditions are frequently not met in ethnic music collections due to low media availability and ill recording environments. In this paper, we propose an automatic genre classification technique especially designed for small, noisy datasets. The proposed technique uses handcrafted features and a vote-based aggregation process. Its performance was evaluated over a Brazilian ethnic music dataset, showing that using the proposed technique produces higher F1 measures than using traditional data augmentation methods and state-of-the-art, Deep Learning-based methods. Therefore, our method can be used in automatic classification processes for small datasets, which can be helpful in the organization of ethnic music collections.

Keywords: Computational Ethnomusicology, Automatic Music Genre Classification, Music Information Retrieval

1 Introduction

Automatic music genre classification (AMGC) is a process that associates a digitized audio signal to a label corresponding to its musical genre. This can be pursued by estimating a vector representation from each audio track and then performing a data-driven classification process. An adequate AMGC system can be helpful for the organization of musical collections.

One of the first methods for AMGC was designed by Tzanetakis and Cook [15]. It relies on the assumption that the auditory content of a recorded sound depends on its spectral shape [13], which means that features that describe the shape of short time spectra produce a vector representation of their content. In this representation, vectors that are close are related to audio excerpts that sound similar and, conversely, distant vectors are related to audio excerpts that sound very different.

The method proposed by Tzanetakis and Cook [15] begins by dividing a digital music track into short (46ms) frames. Each frame has a set of features estimated. After that, the mean and variance of each feature in 1s-long blocks is

* thanks FAPESP for funding this project.

calculated. Last, the mean and variance of the blockwise statistics are calculated, generating a vector that describes each audio track with four dimensions (mean of means, mean of variances, variance of means, variance of variances) for each estimated feature. This vector representation yielded to further classification steps based on machine learning.

Further proposals for improvements in AMGC involved using different feature sets (such was wavelet histograms [6] estimated from the audio signal). These features were manually developed aiming at highlighting aspects of audio that are known to be relevant. For this reason, they are called *handcrafted features*.

More recently, advances in deep learning (DL) techniques enabled using deep neural networks (DNNs) [12] to perform AMGC directly using spectrograms. In this case, the neural network's hidden layers emerge adequate features during the training stage. This phenomenon has shown to be useful in optical character recognition, and work by Costa et al. [3] has shown that they can outperform other techniques in AMGC.

Both using handcrafted features and DL techniques depend on using data driven optimization processes for parameter estimation. Such data is easily available for popular music datasets, because audio files are broadly available and there are datasets [15, 5] specifically aimed at research use. However, obtaining consistent data in ethnic music datasets is hard, as it often implies in performing field recordings, which are costly.

For this reason, social media networks can be a useful tool for building ethnic music datasets. However, this data can consist of low-quality recordings (using hand cameras and with a significant amount of crowd noise). Also, some genres are clearly populated by only a few artists and producers. This can become a problem because, as noted by Sturm [14], such unbalance can force classifiers to learn artist-specific features, rather than genre-specific features. Therefore, maintaining artist balance implies in building very small datasets.

As a consequence of these difficulties, ethnic music datasets can easily contain less than one hundred tracks. Such a small number of tracks implies in problems related to the dimensionality of the vectors employed to describe them (handcrafted features vectors usually contain hundreds of dimensions) and the number of parameters that must be optimized (DL models often contain thousands of parameters). Therefore, it is necessary to use adequate techniques to mitigate these problems and avoid model overfitting.

One possibility is to use data augmentation techniques. In this case, each track is digitally processed to generate a similar, yet different, track. As a consequence, the original dataset size is multiplied by the number of augmentation proposals. Data augmentation has been shown to slightly improve classification results in popular music datasets [7].

Another possibility is to use feature selection techniques. These techniques aim at reducing the dimensionality of the feature vector by removing non-informative features. Although there are many techniques for such, feature selection using a Maximum-Relevance-Minimum-Redundancy criterion [9] has shown to improve classification results in previous work [1].

Last, it is possible to employ classification models that naturally mitigate the dataset size problem. One of these models is a Hidden-Markov Model (HMM) classifier [10]. This classifier, similarly to an isolated-word recognition system, receives framewise features as inputs and models each genre as an isolated HMM. In the prediction stage, it yields the genre with a higher likelihood related to the input sequence. The HMM mitigates the dataset size problem because each track is simultaneously represented by all of its frames, instead of a single vector.

This paper proposes a bag-of-frames [16, 2, 4] variation to this problem. In our approach, we calculate feature tracks and aggregate them up to the texture window level, as described by Tzanetakis and Cook [15]. Then, we select a number of texture windows and use them to simultaneously represent each track. In the prediction stage, all selected texture windows are classified and the algorithm yields the most frequently predicted genre.

The proposed approach relies on the assumption that a short texture window (a few seconds long) can be representative of the whole track's content [6], as long as the track is uniform. By selecting several texture windows, we increase the amount of data related to each track, thus mitigating the problem of dimensionality. Simultaneously, we expect to prevent errors due to choosing texture windows containing crowd noise or other recording artifacts.

The algorithm is described in detail in Section 2. The experiments and results, discussed in Section 3, show that the proposed method outperforms others in a small, ethnic music dataset. This paper is concluded in Section 4.

2 Proposed Method

The genre classification method proposed in this paper relies on mapping each track to several points in a $\mathbb{R}^N$ vector space. Within this vector space, the similarity between two sonic textures is represented by a small Euclidean distance between their corresponding vector mappings. The mapping process, as shown in Figure 1, consists of calculating statistics of the spectral shape descriptors described in Table 1.

The mapping process begins by dividing an audio track into short-time (46ms) frames with 50% overlap between two consecutive frames. Then, a set of features, described in Table 1, is estimated for each frame using the methods and expressions presented by Tzanetakis and Cook [15]. The mean and variance of each feature is calculated over windows of 3s, yielding vectors that describe texture windows.

After that, a set of K different texture windows, linearly distributed in time, are chosen to represent each track. Thus, the feature vectors representing each of the chosen windows are yielded to a classifier algorithm. This algorithm could be any vector classification algorithm, but in this work we evaluated the K Nearest Neighbors (KNN) algorithm and the Support Vector Machines (SVM), which have been successfully used in previous work [15, 6].

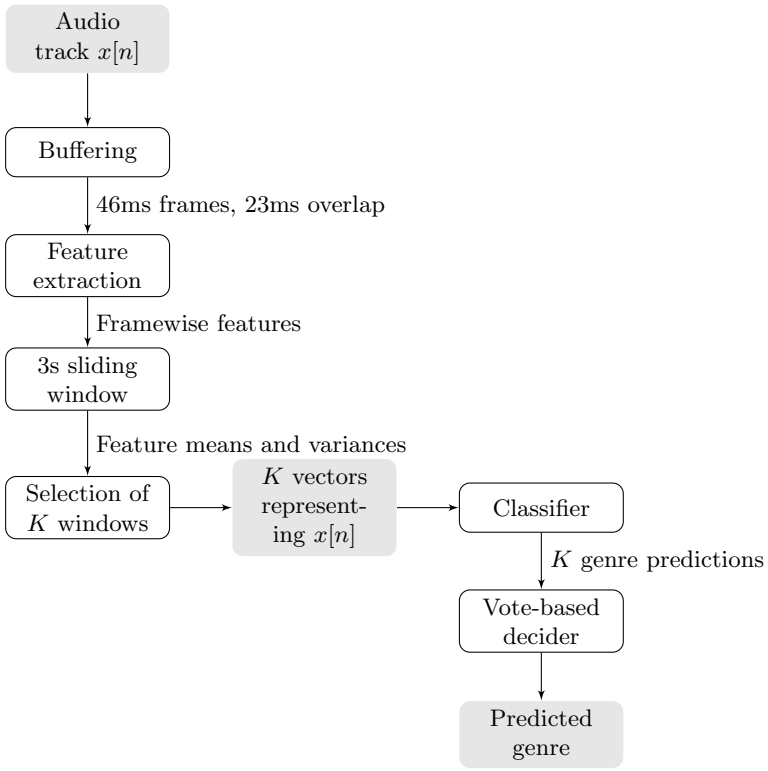

Fig. 1. Overview of the proposed method.

Feature	Description
Energy	Energy of a frame, that is, the squared sum of its samples.
Spectral centroid	Centroid of the frame's spectrum.
Spectral roll-off	Frequency under which 95% of the frame energy is located.
Spectral flux	Sum of the positive difference between the spectra of two consecutive frames. Indicates percussive
Zero crossings	Number of times the time-domain signal within a frame crosses zero.
MFCCs	30 mel-frequency cepstral coefficients, which provide a rough representation of the timbre within a frame.

Table 1. Features estimated during the mapping process.

In the training stage, all texture window representations are simultaneously yielded to the classification model. They are all labeled according to the track's genre.

During testing, all texture windows representing a specific track are yielded to the classifier. Then, the classifier predicts genre labels for all texture tracks and the most frequent label is chosen as the track's label. This prevents errors that

might occur due to localized recording problems, such as crowd noise, speech, and pauses between songs within the same track.

The proposed method was tested in two datasets: the freely-available Seyerlehner [11] dataset and a new dataset, developed for this work, comprising audio extracted from online videos of Brazilian folk music genres. Its performance was compared to previous classifier proposals. The testing procedure, the datasets and the previous classifiers are described in the next section.

3 Evaluation and Results

This section reports the experimental procedures and discussions carried out in this work. Subsection 3.1 describes the datasets employed in evaluation. The classification process in these datasets was also evaluated using methods available in the literature, as discussed in Subsection 3.2. Last, Subsection 3.3 shows the results and brings further discussions.

3.1 Datasets

The proposed method had its performance evaluated over two datasets. The first dataset, YTBR, was especially built for this work and comprises 53 audio tracks extracted from online videos, divided into 7 genres of Brazilian folk music. Table 2 briefly describes this content.

Genre	Tracks	Description
Capoeira	8	A dance and fighting style developed by african slaves during Brazil's colonial era. Frequently exhibited by practitioners on open streets .
Fandango	7	A ballroom dance style from Southern Brazil, derived from European traditions.
Forró	8	A ballroom dance style derived from both European and African music, commonly associated with the Northeast of Brazil.
Frevo	7	A dance style typically present in carnival celebrations in some cities of the Brazilian North and Northeast.
Maracatu	7	A rhythmic style linked to African and Portuguese musical and religious traditions, played in drum ensembles.
Repente	8	A voice and guitar improvisation style in which two musicians develop a game involving rhyme and storytelling.
Toada	8	A solo voice (no accompaniment) style, commonly associated to cattle farmers from semi-arid inlands.
Total	53	A media collection containing Brazilian folk music genres.

Table 2. Content description of the YTBR dataset.

The YTBR dataset was carefully built by selecting musical genres that are typically present in folk musical manifestations. Videos representing each musical

genre were found online, and their audio tracks were extracted. The dataset was built taking care of not using more than one track by each artist, thus avoiding the possibility of training systems to recognize artists, instead of genres [14]. Also, it was built maintaining balance between the number of tracks representing each genre. With both of these restrictions, the resulting dataset was much smaller than the typical datasets used in Music Information Retrieval, comprising only 53 tracks, divided in 7 genres.

The second dataset, Seynherlaner [11], comprises 190 complete, studio-recorded tracks divided into 19 genres of popular Western music. It was used to further corroborate the results obtained in the YTBR dataset and evaluate their generalization in other small datasets.

The classification results for both datasets were evaluated using the proposed method and the reference methods described in the next section.

3.2 Reference methods

The first reference method (GTZAN) is the one proposed by Tzanetakis and Cook [15]. It involves mapping tracks to a vector space short (texture-level) and long (track-level) statistics of low-level handcrafted features. After mapping, the resulting vectors are classified using a Support Vector Machine. This method was reimplemented and validated using the instructions provided in the original paper.

Using a bandpass filterbank, it is possible to build artificial variations of each track that are still recognized as belonging to the same genre. They represent audio tracks as if they were listened to through specific radio equipment. The audio bands were chosen arbitrarily, as there was no reason to choose specific bands for the filters. Dataset augmentation causes the number of samples to exceed their dimension, thus enabling the use of MRMR for feature selection. Combinations of dataset augmentation (AUG) and feature selection (MRMR) were also tested, using the method by Tzanetakis and Cook as basis. We evaluated the result variation by selecting different numbers of features with MRMR [9], and only the best results are reported.

Also, we implemented a HMM-based classifier. It works similarly to the isolated word recognition system described by Rabiner [10]. This system classifies sequences of framewise MFCCs using the Viterbi algorithm. Therefore, it takes the timewise organization of textures into account, which makes sense when classifying music. In our implementation, this classifier models each genre using an ergodic HMM with a diagonal covariance matrix and Gaussian emissions. We tested models comprising different numbers of states (6, 10, 15, 20) and only the results for the best-performing model are reported.

Last, we reimplemented the music genre classifier proposed by Nanni et al. [8]. This classifier is based on convolutional neural networks (CNNs). It interprets the track spectrogram as a sequence of images (each representing a few seconds of audio) and develops image filters that are able to highlight genre-specific characteristics of these images. The classification of each image (that is, each few seconds of audio) is decided using a SoftMax output layer, and the classification

for the whole track is calculated by summing the results of all image-wise output layers. This method was reimplemented using the instructions provided in the original publication, and validated using the same datasets used by the original authors. Several variations of the training parameters were evaluated, and only the best results were reported.

The results regarding these experiments, as well as further discussions on their impact, are presented in the next section.

3.3 Results and discussion

In all experiments, we conducted a K-fold (stratified) cross validation protocol. To evaluate each experiment, we calculated the Recall and Precision, as shown in Expressions 1 and 2, for each class. Then, we calculated the F1-score, as shown in Expression 3, for each class. The mean and standard deviation of the class-wise F1-score across the four test attempts is used as reference for evaluation and further discussion.

$$\text{Recall} = \frac{\text{\# of correct classifications}}{\text{\# of elements known to belong to class}}. \tag{1}$$

$$\text{Precision} = \frac{\text{\# of correct classifications}}{\text{\# of elements predicted to belong to class}}. \tag{2}$$

$$\text{F1-Score} = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{3}$$

Our experiments were first conducted in the YTBR dataset. The experiments in this dataset used 4 folds for cross validation. This number of folds was chosen instead of the usual 10 because the number of data elements is too small.

In the first experiment, we evaluated several variations of the proposed method. This experiment allowed evaluating the impact of the number of texture windows used to represent each track, as well as the impact of using KNN and SVM as the classification model. The results for this experiment are shown in Figure 2.

Figure 2 shows that the average F1-Score is usually higher for SVM-based classifiers than for KNN-basd classifiers. Also, it is possible to see that the standard deviation for SVM-based classifiers tends to be smaller. Last, results show a tendency of increasing the average F1-score with the increase of the number of texture windows, but this increase ceases in values larger than 25. The proposed method variation using 125 texture windows and SVM classifier was used in further tests.

The same evaluation protocol was conducted using all reference methods described in Section 3.2. The results, shown in Figure 3, highlight that the average F1-score achieved by the proposed method is higher than that obtained by all other methods.

The results displayed in Figure 3 show that data augmentation generates only a small performance gain to the original GTZAN method, whereas MRMR provides a slightly higher improvement. It is evident that HMM, CNN and the

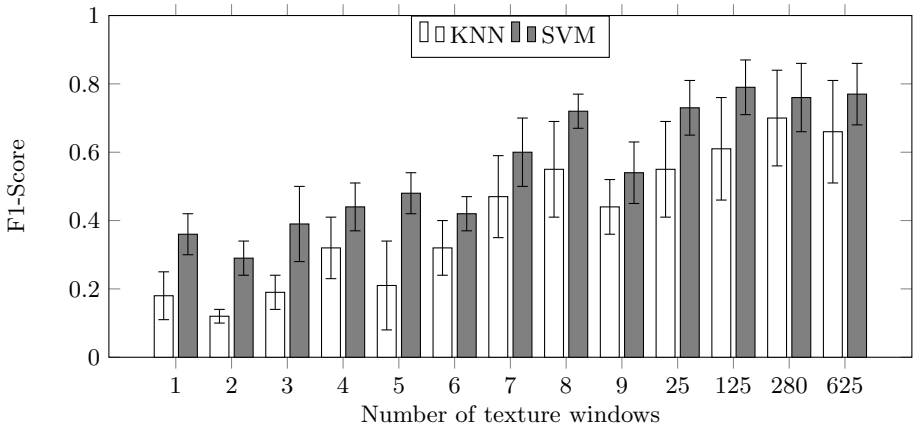

Fig. 2. Classification results in the YTBR dataset for different numbers of texture windows.

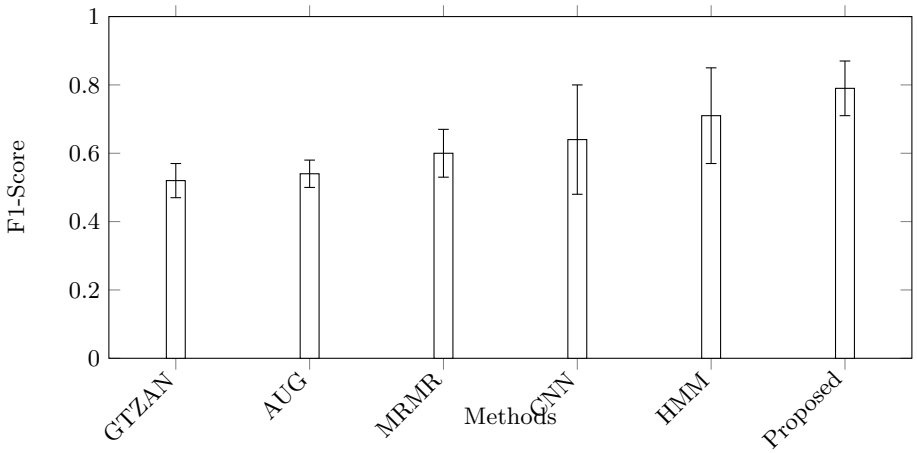

Fig. 3. Classification results in the YTBR dataset for different classification methods. Best results with MRMR were obtained using a 30 feature selection and best results with HMM were obtained using 15-state models for each genre.

proposed method outperform the other three proposals on average. Additionally, it is possible to see that the proposed method obtained a smaller standard deviation when compared to HMM and CNN.

Interestingly, the standard deviation related to both HMM and CNN results are visibly higher than the others. This can indicate insufficient or ill-conditioned data for adequate parameter fitting. However, the proposed method seems to be more resilient to that conditions, as its obtained standard deviation is smaller than the others.

These results show that the proposed AMGC outperforms the other evaluated methods in one small dataset. However, it is necessary to evaluate if these results are consistent in other datasets. For such, we conducted similar experiments in the Seyerlehner dataset [11]. In these experiments, we used a 10-fold cross validation schema. Following the same experimental procedure executed before, we first evaluated the impact of changing the number of texture windows and the classification algorithm in our method. The results are shown in Figure 4.

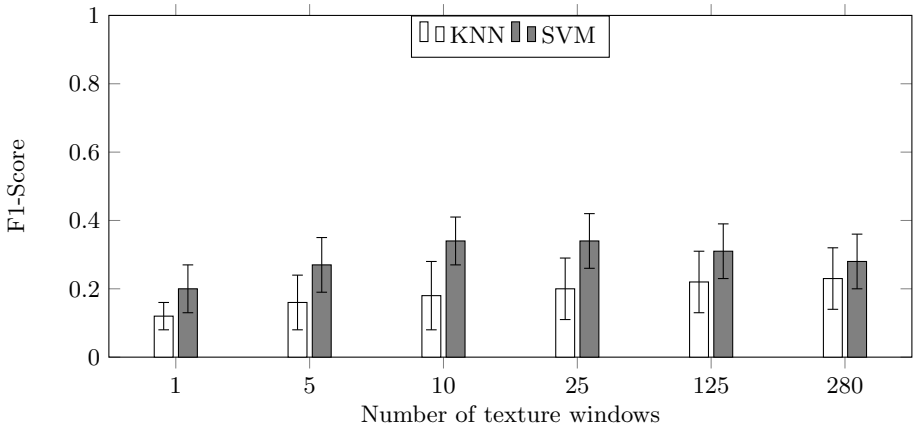

Fig. 4. Classification results in the Seyerlehner dataset for different numbers of texture windows.

Figure 4 shows some aspects that are similar to those seen in Figure 2. SVM-based classification seems to outperform KNN-based classification, and the average performance raises with the increase of the number of analyzed texture windows. However, in this case, the best results were achieved using between 10 and 25 texture windows. This is a smaller number than that required in the YTBR dataset. In the YTBR dataset, however, there is a significant amount of audio data that contains crowd noises and speech, while this is only a minor issue in the Seyerlehner dataset. As a consequence, a smaller number of texture windows can be enough to provide a meaningful representation for such a dataset. The next tests were conducted using the proposed method version with a SVM classifier and 25 texture windows.

The test protocol was applied to the reference methods. GTZAN was used to provide a baseline performance measure. AUG and MRMR were not used in this setup because they were clearly less effective than the other methods. HMM and CNN were evaluated normally. Figure 5 shows the F1-Score estimated in each experiment.

As it can be seen, GTZAN and CNN had similar performances in this dataset, whereas HMM presented a small (less than one standard deviation) improvement. The best results were achieved by the proposed method. These results are

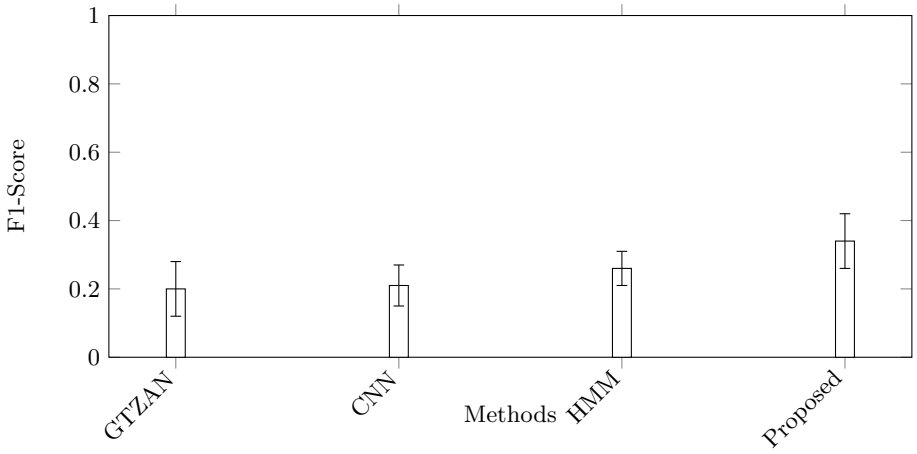

Fig. 5. Classification results in the Seyerlehner dataset for different classification methods.

consistent to those shown in Figure 3, hence it is possible to conclude that the proposed method consistently outperforms the other evaluated methods in small datasets.

Although CNNs have shown to achieve good results in larger datasets, they did not perform well in the small-data scenario. This is probably linked to their large number of parameters, which need a larger amount of data to be adequately optimized. Hence, it is possible that transfer learning techniques, that is, pre-training models in larger datasets before their application to the small dataset, can mitigate this issue and allow to employ the full potential of deep-learning for this scenario.

Nevertheless, our method – using handcrafted features – has outperformed other proposals for the proposed scenario. It uses algorithms that require significantly less computational power than HMMs and CNNs. Genre prediction, in special, only requires processing a fraction of the whole track, which reduces the computational load related to digital signal processing. Therefore, it is more suitable for mobile devices, in which it is relevant to organize personal (often small) media collections.

Next section presents conclusive remarks.

4 Conclusion

This works presents a novel method for automatic music genre classification (AMGC). The method is based on calculating framewise, low-level features from an audio signal and then calculating their statistics within a running texture window of 3 seconds. A fixed number of texture windows is simultaneously used

to represent each audio track, and prediction ambiguities are solved using a simple voting mechanism.

The proposed method has shown to outperform both baseline and state-of-the-art methods. The tests were conducted in two datasets. One of them contains audio tracks for Brazilian folk music genres downloaded from a social network and the other contains Western popular music. The results are consistent in both datasets.

The Brazilian folk music dataset can be freely downloaded. Instructions for download are available in the author's website. Therefore, it can be used in further research on AMGC.

These results show that the proposed method can be used in the organization of both ethnic music collections, which are prone to contain noisy recordings, and personal music collections, which can contain a very small amount of data. Also, the proposed method uses significantly less computational resources than deep-learning or HMM-based methods, thus being more suitable for use in mobile devices.

All results obtained in this work regard small datasets, which is a specific case for AMGC. Although the proposed method was inspired in small, ill-recorded datasets, previous work has shown that similar approaches can yield good results in larger datasets. Tests regarding such datasets comprise a clear direction for future work.

Acknowledgements

The authors thank FAPESP for financial support. The datasets and source code for the proposed method can be downloaded from the author's website (`http://www.dca.fee.unicamp.br/~tavares`).

References

1. Baniya, B.K., Lee, J., Li, Z.N.: Audio feature reduction and analysis for automatic music genre classification. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 457–462 (Oct 2014)
2. Barbedo, J.G.A., Lopes, A.: Automatic genre classification of musical signals. EURASIP J. Adv. Sig. Proc. 2007 (2007), `http://dx.doi.org/10.1155/2007/64960`
3. Costa, Y.M., Oliveira, L.S., Jr., C.N.S.: An evaluation of convolutional neural networks for music classification using spectrograms. Applied Soft Computing 52, 28 – 38 (2017)
4. Jang, D., Jin, M., Yoo, C.D.: Music genre classification using novel features and a weighted voting method. In: 2008 IEEE International Conference on Multimedia and Expo. pp. 1377–1380 (June 2008)
5. Jr., C.N.S., Koerich, A.L., Kaestner, C.A.A.: Feature selection in automatic music genre classification. In: 2008 Tenth IEEE International Symposium on Multimedia. pp. 39–44 (Dec 2008)

6. Li, T., Ogihara, M.: Content-based music similarity search and emotion detection, vol. 5 (2004)
7. McFee, B., Humphrey, E.J., Bello, J.P.: A software framework for musical data augmentation. In: Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015. pp. 248–254 (2015)
8. Nanni, L., Costa, Y.M.G., Lumini, A., Kim, M.Y., Baek, S.: Combining visual and acoustic features for music genre classification. Expert Syst. Appl. 45, 108–117 (2016), http://dx.doi.org/10.1016/j.eswa.2015.09.018
9. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (Aug 2005)
10. Rabiner, L.R.: Readings in speech recognition. chap. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990), http://dl.acm.org/citation.cfm?id=108235.108253
11. Seyerlehner, K.: Annotated seyerlehner genre dataset. http://www.seyerlehner.info/index.php?p=1_3_Download (2012)
12. Sigtia, S., Dixon, S.: Improved music feature learning with deep neural networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6959–6963 (May 2014)
13. Smalley, D.: Spectromorphology: Explaining sound-shapes. Org. Sound 2(2), 107–126 (Aug 1997), http://dx.doi.org/10.1017/S1355771897009059
14. Sturm, B.L.: Classification accuracy is not enough - on the evaluation of music genre recognition systems. J. Intell. Inf. Syst. 41(3), 371–406 (2013), http://dx.doi.org/10.1007/s10844-013-0250-y
15. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10(5), 293–302 (Jul 2002)
16. West, K., Cox, S.: Features and classifiers for the automatic classification of musical audio signals. In: Proceedins of ISMIR 2004 (2004)