ISSN 1870-4069

Enhancing Music Genre Classification Using Tonnetz and Active Learning

Omar Velázquez-López[1], José Luis Oropeza-Rodríguez[1],
Gibran Fuentes-Pineda[2]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Mexico

{ovelazquezl2018, joropeza}@cic.ipn.mx,
{gibranfp}@unam.mx

Abstract. The task of music genre recognition (MGR) has significant applications in the music industry, including copyright control and music recommendations. Traditional feature engineering approaches in classical machine learning (ML) have utilized spectral features like Mel Frequency Cepstral Coefficients (MFCC), alongside time and frequency domain features such as zero-crossing rate, spectral centroid, and spectral rolloff. This study investigates the impact of harmonic, tonal, and rhythmic features, specifically tonnetz, chroma, and tempo, on the performance of ML models. We implemented Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost, both in their traditional forms and with the integration of active learning, to classify genres in the GTZAN dataset. Our results demonstrate that active learning significantly improves model accuracy, with the highest accuracy achieved by the active SVM model at 80.3% using a combination of tonnetz and chroma features. This study underscores the importance of tonal and rhythmic features, particularly tonnetz, in optimizing MGR models. Future work includes expanding the dataset using the developer API from a music streaming platform and exploring alternative feature representations such as auditory filter banks.

Keywords: music genre classification, active learning, tonnetz, chroma, tempo, support vector machine, k-nearest neighbors, xgboost, GTZAN dataset, music streaming platform API.

1 Introduction

The music genre recognition (MGR) has been a fundamental task in the field of music recognition and information retrieval since the early 21st century, with works by authors such as Pachet and Cazaly [1] and Tzanetakis and Cook [2]. Currently, this task has significant applications in the music industry, such as copyright control, genre classification, and music recommendations, driven by the rise of music streaming platforms [3]. In classical machine learning (ML), feature engineering plays a crucial role in measuring the results and performance of models, differing from deep learning algorithms where features are typically extracted automatically during the training process.

To address the problem of MGR, spectral features such as Mel Frequency Cepstral Coefficients (MFCC), along with time and frequency domain features like zero-crossing rate, spectral centroid, and spectral rolloff, have traditionally been used.

This article proposes that harmonic, tonal, and rhythmic features also significantly influence the representation of the musical signal and, consequently, the model's performance. We use tonnetz as a harmonic feature, chromagram as a tonal feature, and tempo (beats per second) as a rhythmic feature. The tonnetz is a visual representation of the harmonic relationships between chords and notes in music, while the chroma features represents the energy of the 12 musical notes over time. It is important to mention that some of the most significant works in chord recognition have used chroma [4] and tonnetz [5] features.

This study focuses on exploring and demonstrating the relevance of tonnetz in MGR using classical machine learning models. We hypothesize that tonnetz is the prominent musical feature in genre classification, as chord progressions tend to vary more between genres than chromatic tones or song tempos. For example, a jazz song and a rock song may share similar chromatic tones and tempos but will have distinctive chord progressions. To this end, we implement Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost, both in their traditional form and by adapting the active learning method proposed by Deng and Ko [6], to classify genres in the GTZAN dataset.

2 Related Work

In the state-of-the-art of musical genre classification (MGR), traditional audio features such as Mel Frequency Cepstral Coefficients (MFCC) and time and frequency domain scalar features, such as zero-crossing rate, spectral centroid, and spectral rolloff, have been widely used in various studies. These audio features have proven effective in representing the acoustic information of music signals.

For instance, Qi et al. [7] include features such as spectral centroid, spectral rolloff, spectral flux, zero-crossing rate, and low energy in their analysis. Ghildiyal et al. [8] use zero-crossing rate, Root Mean Square Energy (RMSE), MFCC, spectral centroid, and spectral rolloff. Elbir et al. [3] employ zero-crossing rate, spectral centroid, spectral contrast, spectral bandwidth, and spectral rolloff. However, Qi et al. [7] and Ghildiyal et al. [8] also mention using musical features. Qi et al. [7] mention features related to texture, timbre, and instrumentation without specifying exactly which features, while Ghildiyal et al. [8] specify including chroma features. Table 1 provides a summary of the features used in each study according to their category.

Both Qi et al. [7] and Ghildiyal et al. [8] calculate two statistical measures for each feature, while Elbir et al. [3] add the standard deviation in addition to these two measures, doubling or tripling the feature size. This is the reason why the feature vectors in each work reach those approximately sizes.

Table 1. Features used in musical genre classification.

Categ.	**Feature**	**Elbir et al. [3]**	**Qi et al. [7]**	**Ghildiyal et al. [8]**
Traditional	MFCC	yes	yes	yes
	Zero Crossing Rate	yes	yes	yes
	Spectral Centroid	yes	yes	yes
	Spectral Rolloff	yes	yes	yes
	Spectral Contrast	yes	-	-
	Spectral Bandwidth	yes	-	-
	Flux	-	yes	-
	Low Energy	-	yes	-
	RMSE	-	-	yes
Musical	Chroma	-	not specified	yes
	Tonnetz	-	not specified	-
	Tempo	-	not specified	-
	Vector size (approx.)	93	60	72

Table 2. Some results for GTZAN in state of art.

Reference	**Model**	**Accuracy %**
Qi et al. [7]	KNN	90
Ghildiyal et al. [8]	Decision Tree	74.3
Elbir et al. [3]	SVM	72.7

Regarding machine learning models, techniques ranging from classical ML, such as k-nearest neighbors (KNN) and support vector machines (SVM), to deep learning (DL) models like convolutional neural networks (CNN) have been applied in MGR.

In the work by Elbir et al. [3], ML models from KNN to SVM and a DL model (CNN) trained with other spectrogram features were trained and evaluated. On the other hand, Ghildiyal et al. [8] compared several ML models, including Decision Tree, Random Forest (RF), and KNN. Both studies trained and evaluated their models' using samples from the GTZAN dataset. Table 2 shows the best-performing model from each work cited in this section. We have only considered results where ML algorithms were evaluated with the GTZAN dataset.

In Table 2 it is also important to note that the high performance of Qi et al.'s [7] KNN model is likely due to the fact that their work not only trained the model with samples from GTZAN, as Ghildiyal and Elbir did, but also added nearly 18,000 samples obtained from the Spotify developer API.

Furthermore, it is important to mention that Ghildiyal et al. [8] achieved even higher performance using their DL model. However, for the purposes of this work, which focuses on ML algorithms, it is not comparable. A similar situation occurs with Deng's work [6], which, although their contribution of active learning during model training is valuable to this work, evaluates their model with a different dataset.

3 Methodology

3.1 Dataset

The GTZAN dataset is widely used for music genre classification (MGC). It consists of 1000 audio tracks, each 30 seconds long, divided into ten genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, and Rock [3]. Its small size aligns well with the focus of this paper, as applying ML algorithms does not require the large amount of data needed by DL algorithms.

3.2 Preprocessing and Features

Data preprocessing includes several stages:

- MFCC Extraction: 20 MFCC coefficients were extracted from each audio sample to capture frequency characteristics with half-second windows.
- Time and Frequency Domain Features: zero-crossing rate, spectral centroid, and spectral rolloff were calculated to obtain additional information from the audio.
- Musical Features: Features such as tonnetz, chroma, and tempo were added to evaluate their impact on genre classification.

Considering that the features shared by all works in Table 1 are MFCC, zero-crossing rate, spectral centroid, and spectral rolloff, these were extracted and feature vectors were created as shown in Table 3. In this work, we decided to call them base features, as they will be used as the minimum length of the feature vector. On the other hand, Table 4 shows the respective size for each musical feature.

The size of the feature vector in each experiment will depend on which musical feature is added. Thus, it can range from a length of 46 to a maximum of 62.

To illustrate the musical features, we generated tonnetz spectrograms, chromagram, and tempo (Beats) plots for representative examples of the jazz and rock genres. Figure 1 shows these plots for the track 'jazz.00004', while Figure 2 presents the same features for the track 'rock.00009'. The tonnetz spectrograms provide a visual representation of the harmonic relationships between chords and notes in music, the chromagram represents the energy of the 12 musical notes over time, and the tempo plots show the onset strength and the distribution of beats over time.

Table 3. Base features.

Feature	Statistical measure	# of features
MFCC (20 coeff)	Mean and variance	40
Zero Crossing Rate		2
Spectral Centroid		2
Spectral Rolloff		2
Vector size		46

Table 4. Musical features.

Feature	Statistical measure	# of features
Tonnetz (6 relationships)	Mean and variance	12
Chroma (12 tones)		24
Tempo (unique value)		2
Vector size		16

The visualizations in Figures 1 and 2 demonstrate how chord progressions and tonal and rhythmic features vary between musical genres. In the tonnetz spectrogram, we can observe significant differences in the harmonic relationships between the jazz and rock tracks, highlighting the tonnetz's ability to capture the distinctive harmonic structure of each genre.

Although the chromagram appears to reveal significant differences in this instance, this is likely because the two tracks are in different musical scales. If both tracks were in the same scale, despite being different genres, they might risk appearing similar, as would be the case with the execution speed, i.e., the tempo.

These differences in tonal and rhythmic features provide a basis for assuming they can aid in music genre classification. However, this exercise is purely intuitive, so it will be necessary to measure the mentioned differences using ML.

3.3 Algorithms and Experiments

In this section, we detail the training procedure for both traditional and active learning approaches using ML algorithms. For all these models, we used numpy files containing manually extracted features from the audio signals. The data was pre-scaled.

The dataset was divided into training and testing sets, with a 70% training and 30% testing split. For each model, hyperparameter optimization was performed through grid search, selecting the best parameters using scikit-learn, except for XGBoost, which was built using the xgboost library.

For each model presented in this paper, hyperparameter optimization was performed through grid search, selecting the best parameters using scikit-learn, except for XGBoost, which was built using the xgboost library.

a) SVM: The optimization of this model consisted of finding the best hyperparameters C and gamma. We fixed an RBF kernel due to its ability to

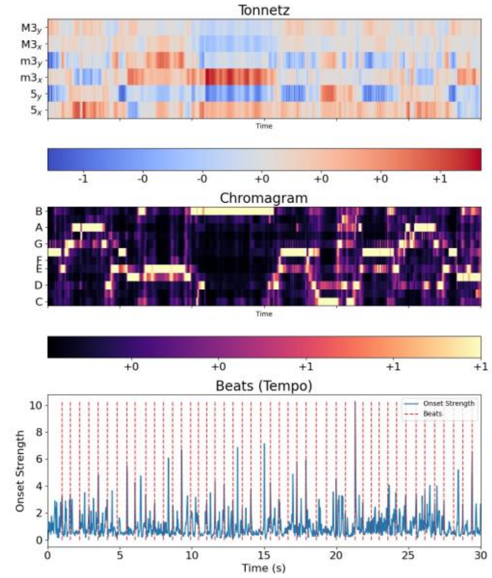

Fig. 1. Tonnetz spectrogram, chromagram and tempo of a jazz track.

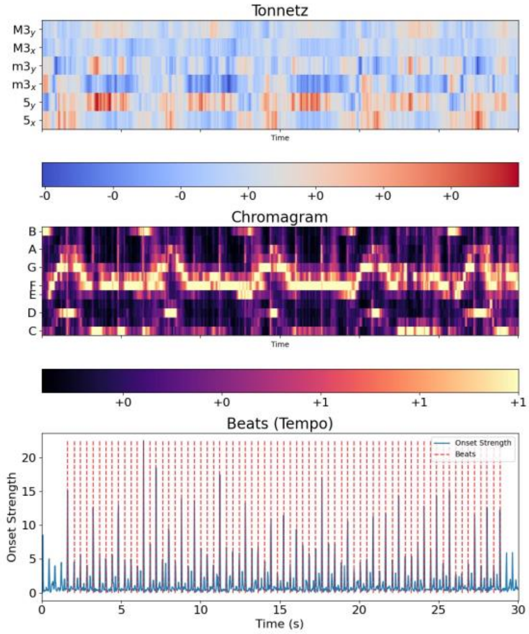

Fig. 2. Tonnetz spectrogram, chromagram and tempo of a rock track.

handle nonlinear problems. Using the base features from Table 3, the best hyperparameters found were a value of 10 for C and 0.01 for gamma.

b) KNN: The best values for n_neighbors, weights, and metric were selected through grid search. Using the base features from Table 3, the best hyperparameters found were 7 for n_neighbors, 'distance' for weights, and 'euclidean' for metric.

c) XGBoost: In this model, the RandomizedSearchCV function from the xgboost library was used to find the best hyperparameters n_estimators, max_depth, learning_rate, subsample, and colsample_bytree. Using the base features from Table 3, the best hyperparameters found were 200 for n_estimators, 6 for max_depth, 0.1 for learning_rate, 0.8 for subsample, and 0.7 for colsample_bytree.

We also conducted experiments integrating Deng's active learning method [6] during the training of each model. We chose Deng's active learning method because of its proven effectiveness in various domains, including text and image classification. Its ability to iteratively improve model performance by selecting the most informative samples aligns well with our goal of enhancing music genre classification accuracy.

For this, an active learning function was used, which iteratively selects the samples with the highest uncertainty from the pool dataset to add them to the training set. In each iteration, the model is trained with the expanded training set and its performance is evaluated. This process is repeated until the defined number of iterations is completed.

The active learning method allows the model to improve its performance by incorporating informative samples in a controlled manner. A block diagram of the method is shown in Figure 3.

We used a value of 10 for the parameter m, which represents the number of iterations in the active learning process. This value was chosen based on preliminary experiments that balanced computational efficiency and performance improvement. In each iteration, the model is trained with the expanded training set, incorporating samples from the pool set with the highest uncertainty. We divided the data into 20% for the initial set and 80% for the pool set.

After training and evaluating the models in their traditional form and also by adapting the active learning method, using the base features from Table 3, we used 30% of the total data for the traditional approach and 30% of the pool set for the active learning method. The results obtained are shown in the second column of Table 5. These show that the use of active learning significantly improves the accuracy of the models.

In the case of SVM, accuracy increases from 66% to 75%, representing a considerable improvement. Similarly, KNN improves from 60% to 69.7%, and XGBoost from 63.67% to 74%. These improvements suggest that active learning allows the models to focus on the most informative samples, resulting in better generalization and performance.

In comparison, traditional models without active learning present lower accuracies, indicating the effectiveness of Deng's active learning approach [6] in MGC problem. However, as mentioned in the introduction of this paper, we are also interested in demonstrating that the musical feature tonnetz influences the results. For this reason, a series of experiments were conducted to evaluate the effect of independently adding each musical feature: tonnetz, chroma, and tempo. The results of these experiments are also presented in Table 5, spanning the third to fifth columns.

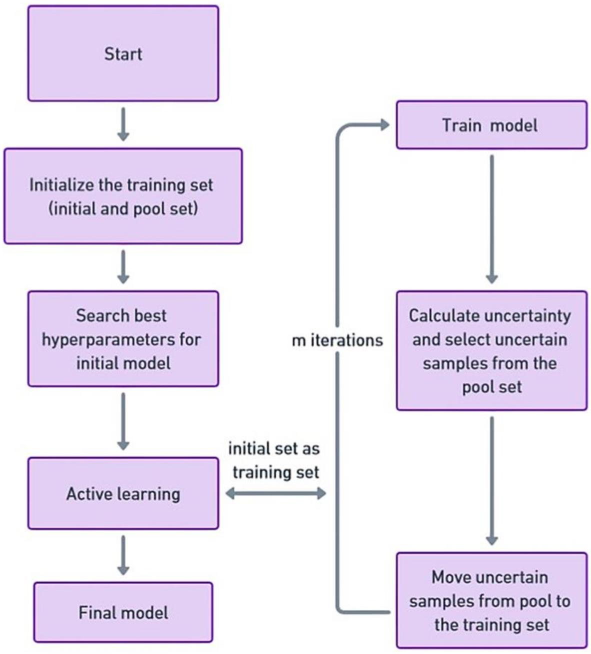
Start
Initialize the training set (initial and pool set)
Search best hyperparameters for initial model
Active learning
Final model
initial set as training set
m iterations
Train model
Calculate uncertainty and select uncertain samples from the pool set
Move uncertain samples from pool to the training set

Fig. 3. Block diagram of active learning.

Reviewing the results of the effect of independently adding each musical feature, it is noticeable that the accuracy of the SVM model improves significantly by adding the tonnetz feature, reaching 73.3%, and with active learning, this accuracy increases to 78.1%. For the KNN model, the addition of chroma shows a significant improvement, especially with active learning, where accuracy reaches 73%. In the case of XGBoost, the inclusion of tonnetz increases accuracy to 66%, and with active learning, it rises to 75%. These results highlight the importance of tonal and rhythmic features in music genre classification, particularly emphasizing the influence of tonnetz.

After verifying our first hypothesis, we conducted a series of comprehensive experiments by combining the musical features to maximize performance in each model. The best and only combination reported was Tonnetz + Chroma for SVM active. The result of this experiment is presented in the sixth and last column of Table 5.

The results show that combining features can lead to significant improvements in model accuracy. The SVM active model achieved the highest accuracy with a combination of tonnetz and chroma, reaching 80.3%. Its confusion matrix is shown in Figure 4.

These results highlight that combining musical features can be beneficial for optimizing model performance. However, there remains a noticeable trend in the relevance of the tonnetz feature over chroma and tempo.

For the statistical analysis, techniques such as paired t-tests could be conducted to compare the performance differences between traditional and active learning methods.

Table 5. Accuracy of our different models.

Model	Base features (%)	Adding tonnetz (%)	Adding chroma (%)	Adding tempo (%)	Adding tonnetz and chroma (%)
SVM	66	**73.3**	66.6	68	-
KNN	60	**69**	65.1	61.2	-
XGBoost	63.67	**66**	64.2	63.6	-
SVM active	75	78.1	77.2	76.3	**80.3**
KNN active	69.7	66.3	**73**	67.8	-
XGBoost active	74	**75**	73	72.3	-

The results would likely indicate that the performance improvements are statistically significant, reinforcing the effectiveness of the active learning approach.

As a final comparison, Table 6 includes the best performance obtained from the experiments in this work compared to state-of-the-art works that have evaluated their models with the GTZAN dataset.

Table 6 shows that our proposed model, SVM with active learning, achieved an accuracy of 80.3%, surpassing the results of Ghildiyal et al. [8] and Elbir et al. [3], who reached 74.3% with Decision Tree and 72.7% with SVM, respectively. Although Qi et al.'s [7] KNN model achieved an accuracy of 90%, significantly higher than the other models, as previously mentioned, this high performance is likely influenced by their use of nearly 18,000 additional samples from the Spotify developer API during training, beyond the GTZAN dataset. These results demonstrate the effectiveness of our approach and highlight the relevance of active learning and musical features, particularly tonnetz, in music genre classification.

4 Conclusion and Future Work

The results of this study highlight the significant impact of active learning and tonal features on the accuracy of music genre classification models. The active learning approach proved to be highly effective, allowing the models to focus on the most informative samples, which in turn resulted in better generalization and overall performance. Specifically, the SVM model with active learning achieved an accuracy of 80.3%, surpassing traditional models and showcasing the efficacy of integrating active learning with carefully selected musical features.

Our experiments demonstrated that the tonnetz feature plays a crucial role in enhancing model performance, particularly when combined with chroma features. This finding underscores the importance of incorporating harmonic, tonal, and rhythmic features to improve the classification accuracy of ML models for MGR. The results suggest that these features capture essential aspects of musical structure that are not fully leveraged by traditional spectral features alone.

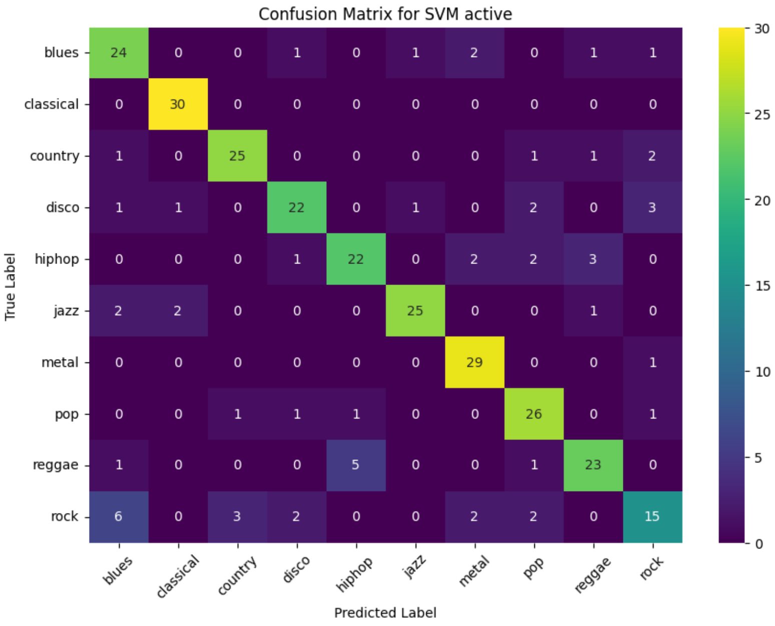

Fig. 4. GTZAN dataset confusion matrix for SVM active using Tonnetz and chroma.

Table 6. Comparative analysis with models of state of art.

Reference	Model	Accuracy %
Qi et al. [7]	KNN	90
This Work	SVM active	80.3
Ghildiyal et al. [8]	Decision Tree	74.3
Elbir et al. [3]	SVM	72.7

However, while the results are promising, the study's scalability to larger and more diverse datasets needs further exploration. The GTZAN dataset, though widely used, is relatively small and homogeneous. To ensure the robustness and generalizability of the proposed approach, it will be necessary to validate the models on larger datasets that reflect the diversity of real-world music streaming platforms. Incorporating additional samples from such platforms, as done by Qi et al. [7], could help address this limitation and provide more comprehensive insights into the model's performance in varied musical contexts.

For future work, we plan to expand the dataset by incorporating additional samples from the developer API of a music streaming platform, as mentioned earlier. This expansion will allow us to test the scalability and robustness of our approach on a broader range of genres and musical styles. Additionally, we aim to explore alternative feature representations, such as auditory filter banks proposed by the authors in [10], to evaluate their effectiveness in MGR. Another avenue for future research is the exploration of convolutional neural network (CNN) models, similar to those employed by Ghildiyal et al. [8], to assess their potential in further improving genre classification accuracy.

By addressing these areas, we hope to develop more robust and accurate models for music genre classification, leveraging advanced feature extraction techniques and larger, more diverse datasets. Such efforts will contribute to advancing the field of MGR and improving the practical applications of genre classification in the music industry.

References

1. Pachet, F., Cazaly, D.: A classification of musical genre. In: Proceedings of RIAO Content-Based Multimedia Information Access Conference (2000)
2. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on speech and audio processing, vol. 10, no. 5, pp. 293–302 (2002) doi:10.1109/ TSA.2002.800560
3. Elbir, A., Çam, H. B., İyican, M. E., Öztürk, B., Aydın, N.: Music genre classification and recommendation by using ML techniques. In: 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), IEEE, pp. 1–5 (2018)
4. Korzeniowski, F., Widmer, G.: Feature learning for chord recognition: The deep chroma extractor. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, p. 37–43 (2016) doi: 10.48550/arXiv.1612.05065
5. Humphrey, E. J., Cho, T., Bello, J. P.: Learning a robust tonnetz-space transform for automatic chord recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 453–456 (2012) doi: 10.1109/ICASSP.2012.6287914.
6. Deng, G., Ko, Y. C.: Active learning music genre classification based on support vector machine. Advances in Multimedia, vol. 2022, no. 1, p. 4705272 (2022) doi: 10.1155/20 22/4705272
7. Qi, Z., Rahouti, M., Jasim, M. A., Siasi, N.: Music genre classification and feature comparison using ml. In: Proceedings of the 2022 7th International Conference on Machine Learning Technologies, pp. 42–50 (2022) doi: 10.1145/3529399.352940
8. Ghildiyal, A., Singh, K., Sharma, S.: Music genre classification using ML. In: Fourth International Conference on Electronics, Communication and Aerospace Technology IEEE, pp. 1368–1372 (2020)
9. Sturm, B. L.: An analysis of the GTZAN music genre dataset. In: Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, pp. 7–12 (2012) doi: 10.1145/2390848.239085
10. Velazquez-Lopez, O., Oropeza-Rodriguez, J. L., Suarez-Guerra, S.: Application of auditory filter-banks in polyphonic music transcription. Computación y Sistemas, vol. 26, no. 4, pp. 1421–1428 (2022) doi:10.13053/CyS-26-4-4271