ISSN 1870-4069

Instrument Recognition Using Spectral Features and SVM

María Fernanda Arámburo-Castell, Juan Pintor-Michimani

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, Pue., México
{maria.aramburoc,juan.pintorm}@alumno.buap.mx

Abstract. The identification of musical instruments is important in audio analysis this allows musical information to be retrieved and identified. For this purpose, it is important to consider the processing of audio signals. In this study, we propose to use spectral parameters such as Mel Frequency Cepstral Coefficients (MFCC), Mel Spectrogram, Chromagram, Harmonic Percussive Index and spectral contrast to capture instrument characteristics. These features model the timbre, harmonic content and energy distribution essential for differentiation. Accurate extraction and processing of these features is essential, as errors can compromise classification performance in complex, polyphonic soundscapes.

Keywords: Instrument Recognition, MFCC, Spectral Features, SVM.

1 Introduction

Instrument recognition is an important concern in audio analysis, especially in music information retrieval, sound source separation and automatic music transcription [15]. The issue is complicated by the fact that most audio signals are polyphonic, i.e. they are a combination of two or more sources, in this case instruments. For this reason, the majority of research divides the monophonic and polyphonic cases into two distinct problems, with the principal objective being to ascertain the instrument or the predominant instrument, respectively.

This is of particular importance given that the predominant instrument is a powerful description of musical data that can be used to identify songs. The application's utility in identifying musical genres is noteworthy, particularly given that certain instruments are often used as distinctive characteristics [4]. This is a subject on which a substantial amount of research has been conducted.

In the case of polyphonic music, there is research by Han et al. [8], who developed a deep CNN for instrument recognition based on Mel spectrogram inputs and aggregation of multiple sliding window outputs on the audio data.

Another interesting research is [7] where the researchers proposed a method for automatic recognition of predominant instruments using SVM (Support Vector Machine) classifiers trained with features extracted from real musical audio signals. Similarly, in [2], an approach is proposed to automatically identify all instruments present in an audio signal using sets of individual convolutional neural networks (CNNs) per tested instrument, which is a similar approach to that

used by Avramidis K. et al. [1] who use RNN (recurrent neural networks), CNN (convolutional neural networks) and CRNN (convolutional recurrent neural networks) to find the dominant instrument, similar to the work of Pons J. et al. [14] who do the same using CNNs with MFCC.

In regard to the selection of parameters, the most commonly utilized are the MFCC (Mel-frequency cepstral coefficient). However, as evidenced in [5, 6], alternative options exist, including the MEL spectrogram, the Chromagram, the Harmonic Percussive Index (HPI), and Spectral Contrast. In this study, the aforementioned parameters are employed to identify the most significant parameters for the developed model, which in this case will be a Support Vector Machine (SVM).This is accomplished by employing specific algorithms, such as Recursive Feature Elimination, or statistical measures like the ANOVA F-value, to identify the most relevant parameters.

2 Methodology

This paper proposes a methodology to train an SVM for both monophonic and polyphonic audio signals. This can be separated in three steps, the pre processing of the signal, extraction of characteristics and the training of model. Each step is relevant, because depending on the chosen parameters the performance of the model can change significantly. In order to expose that, we separate each step in different sub process, this can be seen in Fig. 1, where the three main steps mentioned above are enclosed in a frame and there are the mentioned sub process. Each of them is described in detail below.

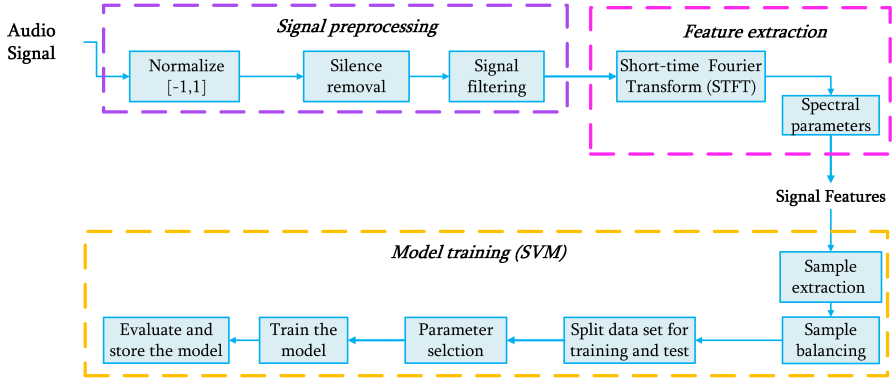


Fig. 1. Proposed methodology.

2.1 Signal Pre-processing

Signal pre-processing is a particularly important step as it prepares the signal to extract its characteristics. In this step, the input signal is in its raw state, which

usually means that it contains noise, has different scales, etc., so it is necessary to normalize and filter the signal before analyzing it. This procedure makes possible to observe the characteristics of the signals and to determine some properties for their subsequent analysis, such as the selection of the amplitude of the windows in which the analysis will be carried out.

Normalize and Silence removal. Normalization of the signal involves modifying its range to align it with the desired range, which in this case is $[-1, 1]$. This can be done through the expression (1):

$$y_{nom} = \frac{y}{max(|y|)}. \tag{1}$$

where y is the original signal, $max(|y|)$ is the maximum absolute value of the signal and y_{nom} is the normalized signal.

Once the signal has been normalized, the next step is to trim the samples in which the signal works with less than $20dB$. In Fig. 2 you can see the normalized signal and the signal after trimming the silences. This Fig. shows how the audio signal of an instrument, in this case viola, changes when we remove the silence spaces, which are especially noticeable at the beginning and at the end of the original signal.

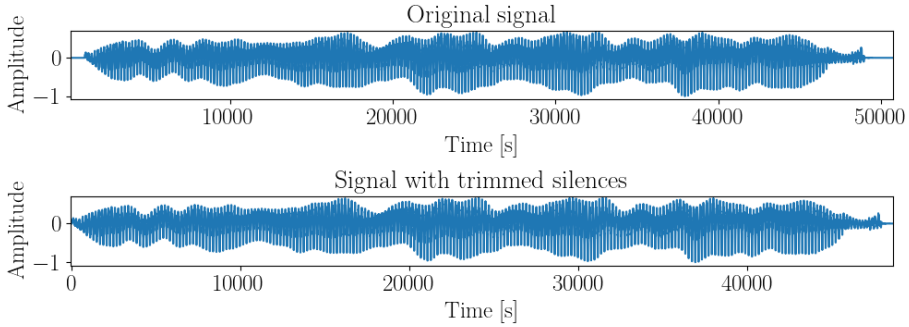


Fig. 2. Original signal and signal resulting from the removal of silences.

Window and Filter. The selection of the window length is typically contingent upon the filter selected from the signal, as both are employed to determine the requisite frequency range for analyzing the interest patron. In this particular instance, it is necessary to select a uniform window for all the instruments, given that a specific pass band filter is proposed for each instrument, with consideration given to their respective work frequency range. Table 1 illustrates the range of each instrument and the period from the slowest frequency, which must be no greater than half of the selected window.

As can be observed, the majority of the values are less than 15 ms, although in certain instances, such as those pertaining to the viola, acoustic guitar, and piano, the values exceed 30 ms. In order to accommodate these values, a window of at least 100 ms is necessary. However, utilizing a longer window may result in the loss of information from the other instruments. Consequently, the recommendation proposed in reference to this matter is to set the window length at 40 ms, with the range from the filter for each instrument corresponding to its respective range of operation.

Table 1. Frequency ranges of instruments [9, 11].

Instrument	F. Min (Hz)	F. Max (Hz)	Signal period (ms)
Cello	65	1000	15.3846
Clarinet	125	2000	8.0000
Flute	250	3500	4.0000
Acoustic Guitar	20	5000	50.0000
Electric Guitar	80	500	12.5000
Piano	27	5000	37.0370
Saxophone	110	2000	9.0909
Trumpet	165	1200	6.0606
Violin	196	3000	5.1020
Viola	130	1000	7.6923
Percussion	30	5000	33.3333

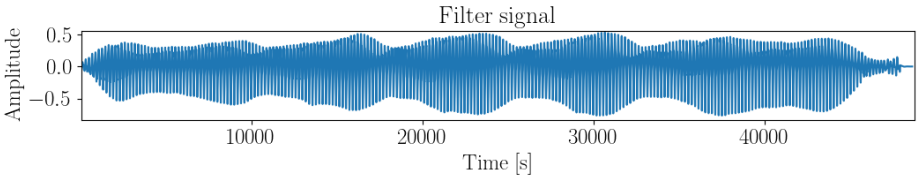


Fig. 3. After-filtered viola audio signal.

Fig. 3 presents an illustration of the signal subsequent to filtration, which correlates with the signal depicted in Fig. 2. Upon examination of both figures, the impact of the filtration process on the audio signal of the instrument becomes evident.

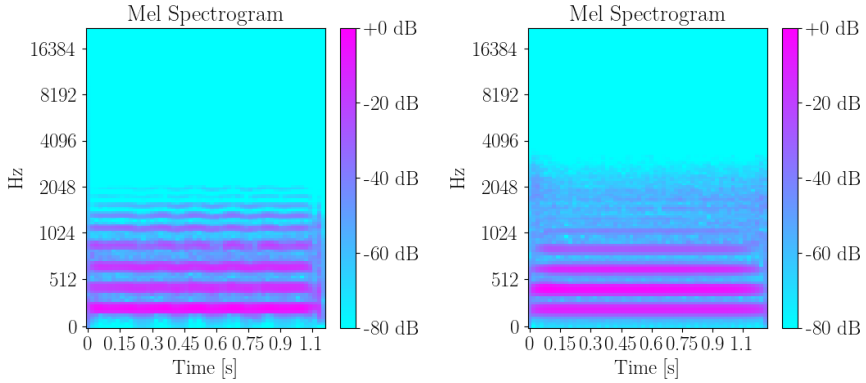
2.2 Feature Extraction

In the audio signal analysis is usual to use the short-time Fourier Transform, which main idea is to consider the changes in frequency in small periods of

time [12], to this it is needed to select the length of those periods, for this case are 40ms and a window function which is multiply for the filter signal to obtain the frequency information. This window shifts across the time and compute the Fourier transform for each resulting window [12] . Part of this process is to select the overlap, which means how much advance the window in the time, for example, the first window is $[0ms, 40ms]$ and the second will be $[20ms, 60ms]$ with overlap of 50% and with overlap of 75% the second window will be $[10ms, 50ms]$.

For each time frame, it is possible to obtain a spectral vector with coefficients associated with a time position. This allows a two-dimensional representation of the squared magnitude of the STFT call spectogram to be plotted, where the horizontal axis represents time and the vertical axis represents frequency [12].

Log-Mel Spectrogram. The Log-Mel spectrogram is a representation that condenses timbre and pitch information computed from the above spectrogram by grouping STFT bins into overlapping frequency bands that approximate human pitch perception [10]. The number of bands is significantly less than STFT, which is also an advantage when selecting them as parameters.



(a) Mel Spectrogram for viola. **(b)** Mel Spectrogram for saxophone.

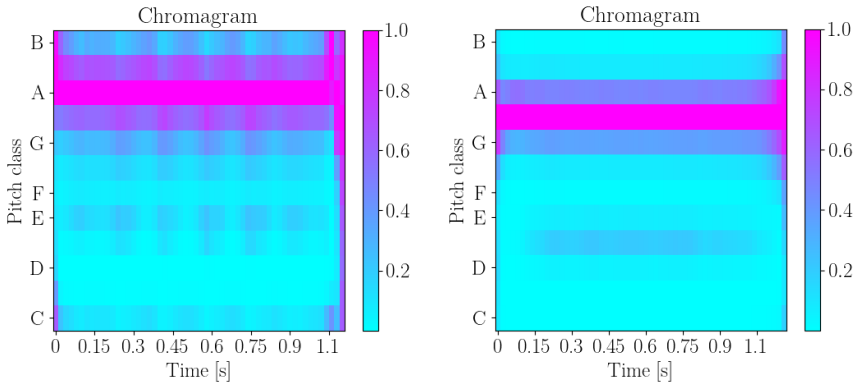
Fig. 4. Mel spectrograms for viola and saxophone for the note A3.

As a result, these parameters give figures as can be seen in Figures 4a and 4b, which are the Mel spectograms for the note A3 for viola and saxophone. As can be seen, they are quite similar, but there are some notable differences in the frequencies from 1024 Hz to 4096 Hz.

Mel Frequency Cepstral Coefficients (MFCCs). The Mel Frequency Cepstral Coefficients are a compact representation of the shape and spectral envelope

of an audio signal, calculated from the Mel spectogram by taking the logarithm of the magnitude of each resulting band and calculating the discrete cosine transform over the resulting band. The resulting real part is similar to the real part of the Fourier transform. The fascinating thing about this parameter is that a small subset contains the most important information, so usually between thirteen and twenty parameters are considered. In this work, it is used twenty parameters.

Chromagram. The Chromagram is a magnitude spectogram similar to the MEL spectogram, the main difference being that the MEL spectogram considers the frequency range, as the Chromagram defines twelve different pitch classes, where each corresponding to a particular frequency range [12]. For example, note A5 has a range of [427.47, 452.89]. Figures 5a and 5b show the chromagram for note A3 on viola and saxophone. It can be seen that both are quite similar, although there are some differences on the note B in the case of the viola.

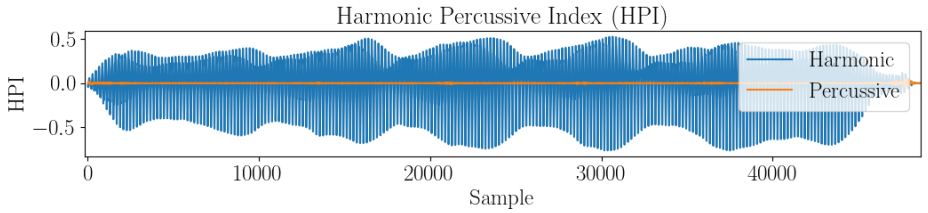


(a) Chromagram for viola. **(b)** Chromagram for saxophone.

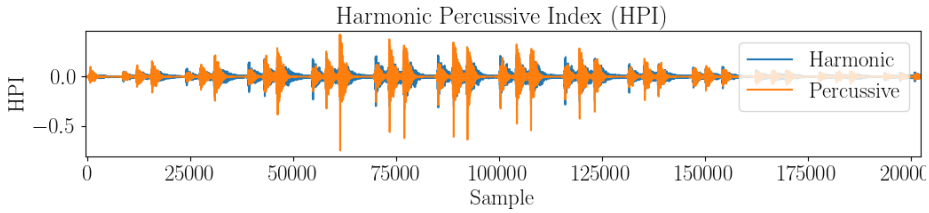
Fig. 5. Chromagram for note A3 for viola and saxophone.

Harmonic Percussive Index (HPI). Musical instruments can be divided into percussive and melodic instruments, the former being characterized by the fact that they can generate vibrations on their own, whereas harmonics require a string or wind to vibrate. This characteristic gives rise to the harmonic and percussive index, which is calculated by separating the harmonic and percussive parts for each window [5]. Figures 6a and 6b show the separation of the harmonic and percussive parts of a viola and a drum. It can be seen that the drum has a lot of percussive energy, while the viola has almost zero.

Spectral Contrast. Spectral contrast characteristics are an important parameter because they provide a representation of the spectral characteristics of the



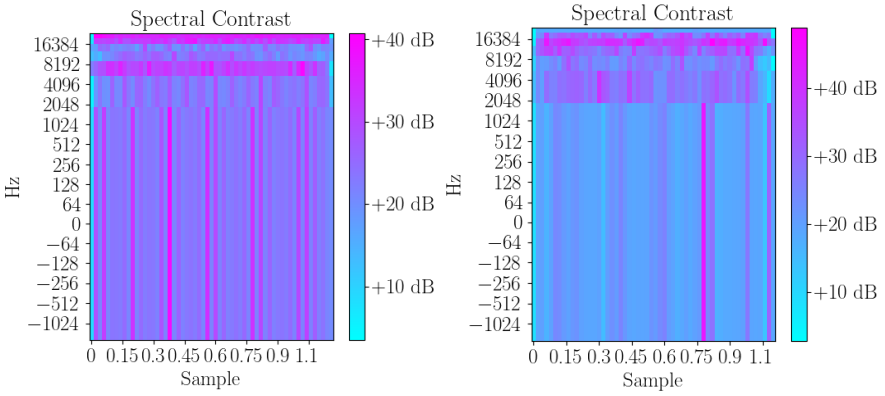
(a) HPI for viola.



(b) HPI for drums.

Fig. 6. Harmonic percussive index for viola and drums.

sound by highlighting the differences between peak and valley energies in different frequency bands. This method emphasizes the relative distribution of spectral energy, which can vary significantly between different types of musical instrument [5]. Figures 7a and 7b show the spectral contrast for note A3 on viola and saxophone. It can be seen that both are quite similar, although there are some differences, specially considering that the time scale are different.



(a) Spectral Contrast for viola. **(b)** Spectral Contrast for saxophone.

Fig. 7. Spectral Contrast for note A3 for viola and saxophone.

Once all the parameters have been calculated, it is necessary to organize them into a dataset in order to be able to use this information in any model. For each window calculated for each audio signal, a vector is generated with the MFCC vector, the MEL spectogram, the chromogram and the spectral contrast as concatenated vectors and the HPI, so that for each case there is a vector of 169 values.

2.3 Support Vector Machines

The problem of instrument recognition is studied with different approaches as RNN, CNN, RNN, CRNN and SVM. For this paper, the selected model was SVM because this allows to quickly train a diversity of models with some variations, like filter or not the signal, change the type of model, the size of data set, etc. Some of these variations are considering in Fig. 8, this scheme mentions some considerations to take in count:

- **Sample extraction.** The datasets contain a lot of information, which is useful for training robust models with deep learning, but for this particular case, it is necessary to take a sample of all the data in order to optimize the training. In this case, two samples are taken, one with 10% and the other with 2.5%.
- **Sample balancing.** In most cases, the datasets do not have the same number of cases from each class, which can lead to poor results.For this, it is necessary to consider oversampling or under sampling in order to have the same number of cases for each class. In this case, the dateset is under sampled. For this, the number of cases for each class is the same as the percentage selected for the sample of the smallest class.
- **Split data.** The data set used to train the model is split into two sets, one for training the model and one for testing it. Some common percentages are 80% and 20%. Since the number of data in this work, the chosen percentages are 85% and 15%.
- **Parameter selection.** The dataset extracted by this methodology have 169 parameters, which is a high number and increase the time to tain the model, for that is necessary to use some technique that reduce the number of parameters on option is Recursive Elimination Feature, but as the name says, it probes the parameters in the model recursively, which leads a hight computational cost. Another option available is the SelectKBest function in Python, which can select the best parameters using statistical techniques such as ANOVA or Mutual Information (MI).
- **Kernel for training the model.** SVM can be trained considering different kernel functions whose are related with how to measure the distance and the expected behaviour of the data. In this case, with linear data, the linear kernel is the best option, while with complex data, such as audio, the radial basis function kernel is a great option.
- **Evaluation of the model.** A train model needs to be evaluated to see how close it is to the test data. To do this, the confusion matrix is a good option because it gives information about the accuracy for each class.

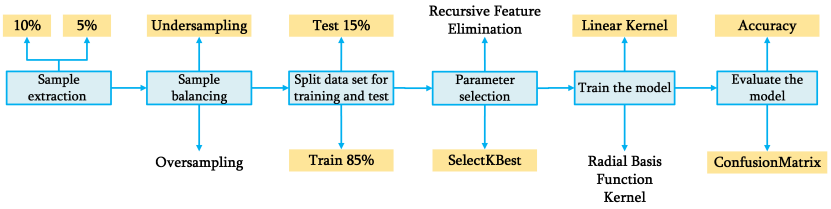


Fig. 8. Proposed methodology to train the SVM.

After considering these factors, different models are trained and evaluated using selected combinations of features. This study investigates model performance under various configurations of kernel, feature selector, and the application of filtering techniques.

3 Results

The proposed methodology is implemented for two datasets, the first one is the dataset IRMAS, which is a polyphonic dataset with eleven class, for this work only nine were selected: cello (cel), clarinet (cla), flute (flu), acoustic guitar (gac), electric guitar (gel), piano (pia), saxophone (sax), trumpet (tru), violin (vio) [3]. The second is the Philarmonia dataset, a monophonic dataset that has twenty classes, only nine of which are used in this work: cello (cel), clarinet (cla), flute (flu), acoustic guitar (gac), percussion (gel), saxophone (sax), trumpet (tru), violin (vio) and viola (viola) [13]. For each data set, all audio signals of each class were analyzed and their features were extracted with a window length of 40 ms and an overlap of 50%, i.e., 20 ms. The sizes of the datasets were 784,802 $\times$ 169 for the IRMAS dataset and 552,051 $\times$ 169 for the Philharmonia dataset. As mentioned above, the data sets are huge and the computational cost of training the model is high. For this, after sampling the dataset with 10% and 2.5% and under sampling the classes, the remaining datasets are shown in Table 2.

Table 2. Samples for balanced datasets.

Dataset	**2.5 %**	**10 %**
IRMAS	13077	52308
Philharmonia	2673	10710

It is now feasible to train models with reduced data sets, which enables the evaluation of various factors such as the kernel type, feature selector, and the number of parameters.

As shown in Table 3, a comprehensive overview of different configurations is provided. The results indicate that, in both datasets, using only 2.5% of the data,

the optimal kernel is the linear one, and the most effective feature selector is f classif. This trend is especially evident in the IRMAS dataset, where a significant drop in accuracy is observed when switching the feature selector. In contrast, under the same conditions, the Philarmonia dataset yields similar results when using the RBF kernel, indicating a more stable behavior with respect to kernel changes.

Moreover, as illustrated in Table 3, the implementation of filters results in a substantial enhancement in performance. The employment of the optimal kernel (linear) and feature selector (f class) for both data sets has been demonstrated to result in enhanced accuracy when utilizing the filters. The Irmas dataset demonstrated an accuracy of 0.84% for the 2.5% sample, while the Philharmonia Dataset exhibited an accuracy of 0.75% for the same percentage. Conversely, when filters are not utilized, the accuracies decrease to 0.21% and 0.65%, respectively, indicating a substantial decline in performance.

Table 3. Model results with different data configurations by changing the kernel, feature selector and number of features (NF).

Dataset	%	Filters	Accuracy	Kernel	Feature Selector	NF
IRMAS	10	✓	0.72	Lineal	f classif	10
IRMAS	2.5	✓	0.75	Lineal	f classif	10
IRMAS	2.5	✓	0.21	Lineal	mutual info classif	10
IRMAS	2.5		0.26	Lineal	f classif	10
IRMAS	2.5	✓	0.23	RBF	mutual info classif	10
IRMAS	2.5	✓	0.21	RBF	mutual info classif	5
PHIL	10	✓	0.83	Lineal	f classif	10
PHIL	2.5	✓	0.84	Lineal	f classif	10
PHIL	2.5	✓	0.68	Lineal	mutual info classif	10
PHIL	2.5		0.65	Lineal	f classif	10
PHIL	2.5	✓	0.68	RBF	mutual info classif	10
PHIL	2.5	✓	0.70	RBF	mutual info classif	5

Another interesting comparison is the effect that dataset size has on model accuracy. As shown, the accuracy is 0.72 for the IRMAS dataset and 0.83 for the Philharmonia dataset when 10% of the data is used. With only 2.5% of the data, the accuracies are 0.75% and 0.84%, respectively. The model trained with filtered signals, lineal kernel and f classif feature selector, has performed well, but to prove that this is true in other circumstances, another sample of 5% is taken from each complete dataset, this to prepare a cross validation with different percentages from the new datasets. Table 4 shows the results from samples of 10%, 5% and 1% of the news datasets. It can be seen that the IRMAS dataset has less accuracy with less data, since the Philarmonia dataset has similar values.

Table 4. Samples for balanced datasets.

Dataset	Sample	Train	10%	5%	1%
IRMAS	10	0.72	0.72	0.71	0.73
IRMAS	2.5	0.75	0.75	0.7	0.65
PHIL	10	0.83	0.84	0.86	0.84
PHIL	2.5	0.84	0.83	0.83	0.83

Table 5 shows a compendium of other test models that have been computed, where it can be seen that the parameters used are MFCC or part of the MEL spectrum with some cases of spectral contrast. None of these use the chromatic diagram and only one uses the HPI.

Table 5. Results of models with different data configurations The selected characteristics are listed.

Dataset	IRMAS	IRMAS	IRMAS	PHIL	PHIL	PHIL
Percentage	10	2.5	2.5	10	2.5	2.5
Accuracy	0.72	0.75	0.26	0.83	0.84	0.65
Filters	✓	✓		✓	✓	
Features	MFCC 3	MFCC 3	MFCC 1	MFCC_3	MFCC_3	MFCC 3
	MFCC 5	MFCC 4	MFCC 3	MFCC_4	MFCC_4	MFCC 4
	MEL 1	MFCC 5	MFCC 4	MFCC_5	MFCC_5	MEL 1
	MEL 2	MEL 1	MEL 71	MFCC_6	MFCC_6	MEL 2
	MEL 67	MEL 2	MEL 73	MFCC_7	MFCC_7	MEL 3
	MEL 69	MEL 67	MEL 75	MEL 1	MEL 1	MEL 127
	MEL 71	MEL 69	MEL 76	MEL 2	MEL 2	MEL 128
	MEL 72	MEL 71	MEL 77	MEL 3	MEL 3	CONTR 1
	MEL 73	MEL 72	MEL 78	MEL 4	MEL 4	CONTR 7
	CONTR 7	CONTR 7	MEL 79	CONTR 7	CONTR 7	HPI

4 Conclusions

The findings presented above, along with the data in Table 5, highlight the critical role of filtering in achieving optimal results and reducing training time. The application of filtering techniques consistently improves model performance. In the absence of such filtering, the resulting data quality is significantly compromised.

Among the features evaluated, MEL spectrograms and MFCCs proved to be the most effective, enhancing classification accuracy across a range of models. These parameters consistently delivered strong results. In contrast, the chromagram and Harmonic Percussive Index (HPI) were found to be less impactful. The HPI was excluded in most models, though an interesting exception occurred with a support vector machine (SVM) trained on unfiltered data from the Philharmonia dataset, where the HPI contributed to an accuracy of 65%. This suggests that while the HPI is generally less useful, it may still offer value in specific contexts. Overall, the results underscore the importance of careful feature selection and the significant benefits of preprocessing in improving model performance.

References

1. Avramidis, K., Kratimenos, A., Garoufis, C., Zlatintsi, A., Maragos, P.: Deep convolutional and recurrent networks for polyphonic instrument classification from monophonic raw audio waveforms. In: Proceedings of the 46th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021). pp. 3010–3014 (2021)
2. Blaszke, M., Kostek, B.: Musical instrument identification using deep learning approach. Sensors (Basel) **22**(8), 3033 (2022)
3. Bosch, J., Bogdanov, D., Gómez, E., Herrera, P.: Irmas: A dataset for instrument recognition in musical audio signals. `https://www.upf.edu/web/mtg/irmas`, accessed: 2024-11-19
4. Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR). pp. 559–564 (2012)
5. Chulev, J.: Improving musical instrument classification with advanced machine learning techniques. arxiv (2024). https://doi.org/10.48550/arXiv.2411.00275
6. Ding, E., Sharma, E.: Musical instrument identification using machine learning. Journal of Student Research **13**(2), 1–10 (2024)
7. Fuhrmann, F., Herrera, P.: Polyphonic instrument recognition for exploring semantic similarities in music. In: Proceedings of the 13th International Conference on Digital Audio Effects (DAFx10). pp. 1–8 (2010)
8. Han, Y., Kim, J., Lee, K.: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**, 208–221 (2017). https://doi.org/10.1109/TASLP.2016.2632307
9. Hispasonic: Tabla de rango de frecuencias de instrumentos musicales. `https://www.hispasonic.com/reportajes/tabla-rango-frecuencias-instrumentos-musicales/39`, accessed: 2024-11-19
10. Lerch, A.: An Introduction to Audio Content Analysis: Music Information Retrieval Tasks and Applications. John Wiley & Sons (2023)
11. Meza, F.J.A., Mella, C.A.A.: Módulo web para ecualización de sonido. Informe final de proyecto de título, Pontificia Universidad Católica de Valparaíso, Facultad de Ingeniería, Escuela de Ingeniería Informática (December 2012)
12. Müller, M.: Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer Cham (2015)
13. Orchestra, P.: Philharmonia orchestra sound samples. `https://philharmonia.co.uk/resources/sound-samples/`, accessed: 2024-11-19
14. Pons, J., Slizovskaia, O., Gong, R., Gómez, E., Serra, X.: Timbre analysis of music audio signals with convolutional neural networks. In: Proceedings of the 25th European Signal Processing Conference (EUSIPCO). pp. 2744–2748 (2017)
15. Yu, D., Duan, H., Fang, J., Zeng, B.: Predominant instrument recognition based on deep neural network with auxiliary classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 852–861 (2020)