Musical Genre Classification Using Mel-Spectrograms and Mel-Scalograms

Luiz Alberto G. Viana[1], Beatriz M. dos Santos[2], Antonio C. L. Fernandes Júnior[1] e Eduardo F. de Simas Filho[1]
[1]Programa de Pós-Graduação em Engenharia Elétrica
[1]Departamento de Engenharia Elétrica e de Computação, [2]Instituto de Humanidades, Artes e Ciências
Universidade Federal da Bahia
Salvador, Bahia
E-mails: luiz.guimaraes@ufba.br, b.marques@ufba.br, antonio.lopes@ufba.br, eduardo.simas@ufba.br.

***Abstract*— Musical genre is a category that groups songs with similar characteristics in terms of style, form, instrumentation, rhythm, harmony, lyrics, or cultural function. The task of Musical Genre Classification is extensively studied in the field of Music Information Retrieval (MIR), and several deep learning techniques have been explored to address it. In this work, we propose the study of mel-scalograms as an alternative representation for the task of music genre classification. We present a systematic comparison between mel-spectrograms and mel-scalograms by evaluating different CNN architectures: MobileNetV2, EfficientNetB0, ResNet50, and VGG16. The experiments were conducted using the GTZAN dataset, which is widely used in the literature, applying a complete training pipeline that includes data augmentation, transfer learning, fine-tuning, and 5-fold cross-validation. The results showed that mel-spectrograms had a slight advantage in average validation accuracy, while mel-scalograms presented a lower standard deviation across training runs. The best-performing models were EfficientNetB0 and ResNet50, achieving up to 86% accuracy. Our findings suggest that both representations are viable, with consistent results across different network architectures and evaluation folds.**

***Keywords*— Musical Genre Classification, Musical Information Retrieval, MIR, Wavelet, Scalogram, Mel-Scalogram, Mel-Spectrogram, Convolutional Neural Network, Data Augmentation, Transfer Learning, Fine-tuning.**

I. Introduction

Musical genre is a category that groups songs with similar characteristics in terms of style, form, instrumentation, rhythm, harmony, lyrics, or cultural function. However, these labels often do not have clearly defined boundaries [1]. The task of Musical Genre Classification is extensively studied in the field of Music Information Retrieval (MIR), and the inherent subjectivity of musical genres makes it a complex challenge [2]. Its applications are not limited to recommendation systems, market analysis, and automatic music tagging, but can also be part of more complex processes such as music generation.

The last decade has seen dramatic improvements in a wide range of such music classification tasks due to the increasing use of deep learning [3]. Architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been successfully applied to different MIR tasks. In the context of Musical Genre Classification, several techniques have already been explored. Recently, [4] combined CNNs with a transformer-based model, fusing audio and lyrics features. Similarly, [5] proposed a multimodal approach using the VGG16 architecture. In contrast, [6] explored a variety of models, including Feedforward Neural Networks (FNN), Long Short-Term Memory (LSTM), Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN). Furthermore, [7] focused on the classification of Brazilian musical genres using Support Vector Machines (SVM), highlighting the applicability of traditional machine learning techniques to culturally specific datasets.

One of the most critical design choices when developing deep learning models for audio is how to represent the audio signals. Some studies have used raw audio directly for training [8]. Time-frequency representations such as spectrograms [9], mel-spectrograms [4], MFCCs (Mel-Frequency Cepstral Coefficients) [6], tempograms [10], scalograms [11], and mel-scalograms [12] are widely used because they allow visual models, like CNNs, to process audio as images. The motivation behind this technique lies in the fact that the problem of image classification has been extensively studied over the past decade, with several network architectures developed specifically for this domain. Observing the error rates in competitions such as the ILSVRC Challenge [13] is a good way to measure this progress.

In this work, we propose the study of mel-scalograms as an alternative representation for the task of automatic music genre classification. The mel-scalogram was proposed by [12] for training CNNs in the task of Musical Tempo Estimation and achieved good results as a complement to the mel-spectrogram. In this study, we present a systematic comparison between mel-spectrograms and mel-scalograms by evaluating different CNN architectures on the widely used GTZAN dataset [1]. We aim to investigate the classification performance gains brought by this new representation and analyze which characteristics are better captured by each method. Additionally, we propose a complete training pipeline including data augmentation, transfer learning, fine-tuning, and 5-fold cross-validation to ensure a robust evaluation of the models.

II. Proposed Model

The proposed model for musical genre classification consists of generating an image representation of the audio signal

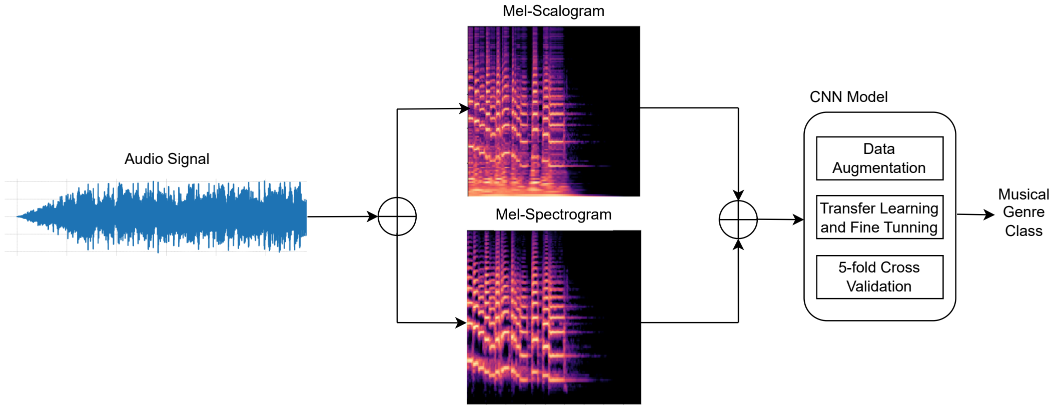


Fig. 1: Simplified proposed model. The audio signal is transformed into either a mel-spectrogram or a wavelet-based mel-scalogram. The resulting image is then used to train the convolutional neural network model. This training process incorporates data augmentation, transfer learning, and fine-tuning techniques, and is evaluated using 5-fold cross-validation. Finally, a musical genre label is assigned to each audio signal.

from a musical piece labeled with its corresponding genre. In some experiments, the audio signal was converted into a mel-spectrogram, while in others it was transformed into a mel-scalogram. The objective is to perform supervised learning by training a convolutional neural network with the generated images, as illustrated in Figure 1.

A. Dataset

In this work, we used the GTZAN Genres dataset, one of the most widely adopted benchmarks in the literature for Musical Genre Classification. The dataset was introduced by Tzanetakis and Cook in their seminal 2002 study on audio-based genre classification [1], and it remains a reference point for evaluating new methods in Music Information Retrieval (MIR).

The GTZAN dataset contains 1,000 audio excerpts of 30 seconds each, distributed evenly across 10 musical genres: classical, country, disco, hip-hop, jazz, rock, blues, reggae, pop, and metal. Each genre class contains 100 audio clips, making the dataset balanced and well-suited for supervised learning tasks. The excerpts were collected from various sources including CDs, radio, and MP3 files, and were standardized to mono, 16-bit, 22,050 Hz WAV format.

Despite its age, the GTZAN dataset remains popular due to its accessibility, simplicity, and the wide range of musical styles it encompasses. Many recent studies continue to use GTZAN as a benchmark for testing new feature extraction techniques, deep learning architectures, and classification strategies [6], [4], [5].

It is important to note that the dataset has known limitations, such as potential duplicates and lack of metadata, which have been discussed in the literature. Nevertheless, it provides a consistent baseline for comparing performance across different musical genre classification models.

B. Audio Signal Representation as Image

In tasks involving Musical Genre Classification, signal representation plays a fundamental role, particularly when using convolutional neural networks (CNNs), which require two-dimensional inputs. In this work, each audio clip was processed without discarding any portion of the signal, preserving the full 30 seconds as available in the original dataset.

All audio files were first converted to mono using a down-mixing strategy, combining the left and right channels into a single audio stream. The sampling rate was kept at 22050 Hz, consistent with the original configuration of the GTZAN dataset. This sampling rate is widely adopted in Music Information Retrieval (MIR) tasks, as it provides sufficient resolution for capturing both low and high-frequency content relevant for genre identification.

Time–frequency representations were then computed to transform the one-dimensional waveform into an image format. These representations, such as mel-spectrograms and mel-scalograms, allow CNNs to capture both temporal and spectral patterns in a way similar to image classification tasks. No temporal cropping or segmentation was applied, and the entire duration of each audio file was used to generate the input representations.

1) Mel-Spectrogram: The mel-spectrogram combines the Short-Time Fourier Transform (STFT) with a frequency-to-Mel scale conversion, creating a more perceptually relevant representation of audio. This method was chosen for its widespread application in visual audio representation, providing a basis for comparative analysis. The parameters for the mel STFT, such as frame length, hop size, and the use of 128 Mel bands, directly influence the time and frequency resolution of the spectrogram [14]. In this study, a frame length of 2048 samples and a hop size of 512 samples were employed. This configuration ensures a balance between time and frequency resolution, suitable for the task of genre classification. Figure 2a illustrates a mel-spectrogram generated from the processed audio signal.

2) Mel-Scalogram: Wavelets have been widely used in previous studies, such as [1], and have proven to be an effective technique for detecting events over time. The scalogram is a visual representation of wavelet coefficients. As an example of its application in Music Information Retrieval (MIR), [15] investigated different methods for generating scalograms and achieved promising results in the task of musical tempo estimation.

The wavelet scalogram may be generated from the preprocessed audio signal vector by initially applying the Continuous Wavelet Transform (CWT). Given a signal $f(t)$, its CWT is defined as follows:

$$\mathcal{W}_f^{\psi}(a,\tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-\tau}{a} \right) dt \tag{1}$$

where the parameter a (>0) refers to the scale, and τ represents the translation or location of the mother wavelet function $\psi(t)$. Both a and $\tau \in \mathbb{R}$. The parameter a controls the dilation/contraction of the mother wavelet function. The superscript asterisk in $\psi^* (\cdot)$ denotes the complex conjugate of the function $\psi(\cdot)$, and $\mathcal{W}_f^{\psi}(a,\tau)$ is known as the wavelet coefficient [16]. In our implementation, the continuous wavelet transform was computed in its discrete form by applying the FFT to both the input signal and the wavelet, performing pointwise multiplication in the frequency domain, and then using the inverse FFT to return the result to the time domain.

Several mother wavelet functions can be used in signal analysis. It is still unknown which continuous wavelet function performs better for the musical genre classification problem. In this study, the complex Morlet wavelet was chosen after visually analyzing the mel-scalogram generated by different wavelet functions. The complex Morlet wavelet is defined as:

$$\psi(t) = \frac{1}{\sqrt{\pi B}} e^{-\frac{t^2}{B}} e^{j2\pi C t} \tag{2}$$

with $B = 6$ and $C = 6$, where B is the bandwidth and C is the center frequency. These values were chosen based on a visual inspection of the resulting scalograms, aiming to produce images that closely resembled the structure and clarity of mel-spectrograms. The wavelet scale parameter a was selected to represent the Mel frequencies, using 128 bands to maintain consistency with the mel-spectrogram generated. A mel-scalogram generated using this approach can be observed in Figure 2b. This scalogram generation methodology was originally proposed by [12] for the task of musical tempo estimation.

III. Convolutional Neural Network Training Procedure

In this work, we use classical image classification architectures that have shown strong performance in various applications. These include MobileNetV2 [25], EfficientNetB0 [26], ResNet50 [27], and VGG16 [28]. Our goal is to compare the performance of these different architectures on the task of Musical Genre Classification and determine which model achieves the best results.

We defined eight experiments in total — one for each architecture using mel-spectrograms and one for each architecture using mel-scalograms. For all experiments, we used 5-fold cross-validation, splitting the dataset into 80% for training and 20% for validation. This split was chosen due to the relatively small number of examples in the dataset. We implemented our models in Python using TensorFlow and Keras, with all training and evaluation performed in the cloud on Google Colab Pro.

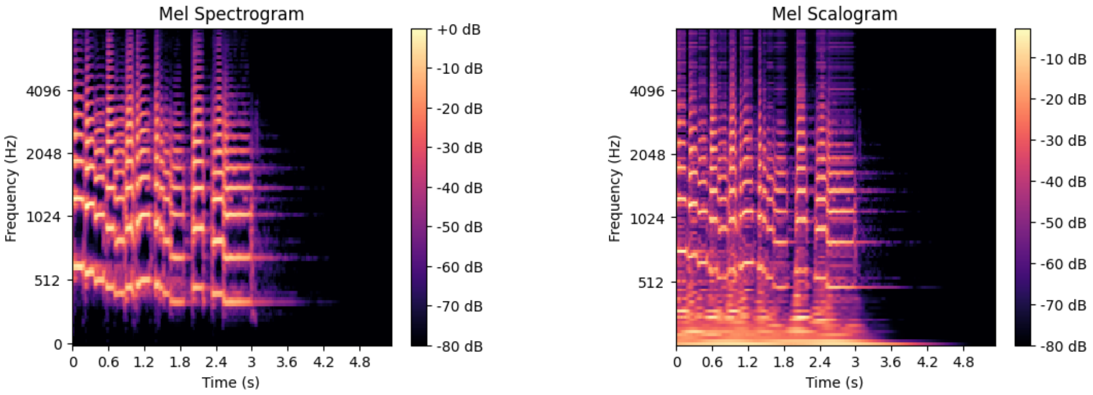
For each architecture, we added a custom top layer specific to our classification task, consisting of a Global Average Pooling layer, a dense layer with 128 neurons and ReLU activation, a Dropout layer with a rate of 30%, and a final dense layer with 10 neurons and softmax activation.

A. *MobileNetV2*

The MobileNetV2 architecture [25] was developed to work efficiently on mobile applications (in consideration of the high computational resources required by state of the art networks). This network is based on MobileNetV1 and its main innovation is the use of blocks with inverted bottlenecks. That novel layer module significantly reduce the memory footprint and the need for main memory access. The model architecture consists of a basic building block which is a bottleneck depth-separable convolution with residuals - that substitutes the traditional convolutions with a combination of a depth-wise convolution (which applies one filter in each input channel) and a pointwise convolution 1x1 (combine the channels to generate new representations. It has an initial fully convolution layer with 32 filters, 19 residual bottleneck layers, kernel size 3x3, dropout and batch normalization during training, with the exception of the first layer. Because ReLU zeroes out negative activations and thus can discard information, MobileNetV2 instead uses a linear bottleneck: it first expands the channels with a 1×1 convolution, applies a depthwise 3×3 convolution, and then linearly projects back into the lower-dimensional space before adding the residual shortcut. The output of the block is added to the original input, forming a residual shortcut. (It has an constant use of expansion rate throughout the network except for the first layer and the width multiplier was applied in all layers except the last convolutional layer).

B. *EfficientNetB0*

In an attempt to understand if there is an ideal way to scale up ConvNets for better accuracy and efficiency, a new scaling approach called compound scaling is proposed, which aims to uniformly increase the input scales, balancing the dimensions (width, depth and resolution of the input) by scaling them in a constant ratio. Obtained through architecture search, the base architecture named EfficientNetB0 [26] is similar to MnasNet[29], except that EfficientNet is larger. It consists of a sequence of MBConv (Mobile Inverted Bottleneck) blocks with Squeeze-and-Excitation for channel enhancement. From B0, larger variants are generated by systematically scaling the three main dimensions mentioned above, with the scaling being controlled by a single chosen coefficient, which allows these dimensions to be adjusted in a coordinated manner. The main advantage of this method is that the optimal parameters can be determined based on a search performed once on



(a) (b)

Fig. 2: (a) Mel-Spectrogram and (b) Mel-Scalogram generated from the "trumpet"audio example provided by [17].

the small baseline network without the search cost becoming expensive. This method made it possible to scale a basic model (EfficientNetB0) up to larger models ranging from EfficientNetB1 to B7, with good gains in accuracy and efficiency, fewer parameters and FLOPs compared to models such as ResNet or Inception [30].

C. ResNet50

In an attempt to solve the degradation problem present in very deep neural networks, ResNet [27] introduces a deep residual learning framework. Instead of learning direct mappings, ResNet blocks learn residual functions, reformulating the output as a sum between the input and the transformation performed by the convolutional layers. This structure is implemented with "shortcut connections", of the identity mapping type, which directly sum the input with the output of the residual subnetwork, without adding extra parameters or computational complexity. For comparison, a Plain Network was made based on the VGG architectures but lighter, with 3x3 filters - considering the same number of filters and layers in case of same output size of feature map and doubled number of filters if the size is halved - and convolutions with stride 2 for downsampling. Then, for the ResNets, there is the addition of shortcuts, which can be used in two ways: If the input and output dimensions match, an identity shortcut is used; otherwise, a projection shortcut implemented with a 1×1 convolution is applied to align dimensions, most commonly at downsampling stages. The basic architecture consists of stacking residual blocks, for deeper versions (50+ layers) bottleneck blocks with 3 layers are used (1x1 → 3x3 → 1x1). When the blocks are added to the convolutions, the sum is done channel by channel between feature maps.

D. VGG16

This architecture focuses on the depth of ConvNets working with more convolutional layers, proposing a simple and deep convolutional neural network architecture [28], composed exclusively of small 3×3 convolutions and 2×2 max-pooling, stacked on top of each other. Instead of using large filters such as 7×7 or 11×11, several 3×3 layers in a row can achieve the same larger receptive field effect, but with fewer parameters and more non-linearities, increasing the expressive power of the network. The architecture is configured as follows: a stack of convolutional layers with 3x3 filters - it also uses 1x1 followed by non-linearity - with stride of 1 and padding for preserving the resolution; five max pooling layers for some of the convolutional layers (with 2x2 windows and stride of 2 to reduce the spatial resolution between the blocks). After the stack of convolutional layers it has 3 fully connected layers (being two with 4096 channels and the last one with 1000 channels considering each output class) and finishes with the softmax layer;ReLU is used as activation function after each convolution in all hidden layers and no use of Local Response Normalization since it didn't improve performance in the experiments. This is the basic of the architecture, for the variants tested, the configurations differ in depth (from A to E, in which network A has 11 weight layers and network E has 19 weight layers, it increases the depth progressively), with each pooling, the number of filters doubles (64 → 128 → 256 → 512). The study argues that the division made (for example, using 3x3 receptive fields throughout the network instead of using a large one only once) makes the network present a more discriminative decision function and also allows it to have a decrease in the number of parameters.

E. Transfer Learning and Fine-tuning

Transfer learning is a powerful strategy in deep learning that allows the reuse of knowledge acquired in a source domain to improve performance in a target domain. Particularly in visual tasks, pretrained convolutional neural networks (CNNs) on large-scale datasets such as ImageNet have demonstrated great effectiveness when adapted to new classification tasks [24]. This approach is especially valuable when the available data for the target task are limited, enabling faster convergence, improved generalization, and reduced computational costs.

In this work, we employed transfer learning across all evaluated architectures by initializing the models with ImageNet-pretrained weights. Initially, we froze all convolutional layers and trained only the new classification head for 100 epochs.

Following this, we applied a fine-tuning procedure in which all layers were unfrozen and the entire network was trained for an additional 20 epochs with a reduced learning rate. This two-phase strategy allows the model to gradually adapt both low- and high-level features to the specific characteristics of the musical genre classification task, while minimizing the risk of overfitting.

F. Data Augmentation

Data augmentation is a widely adopted strategy in deep learning to increase the diversity of the training set by generating realistic variations of the original data. This approach helps reduce overfitting and is commonly viewed as a form of regularization [23].

When dealing with image data, common augmentation techniques include rotation, translation, flipping, and color adjustment. However, such operations are not applicable to scalograms, since they are generated from audio signals and encode precise time-frequency information. Applying arbitrary geometric transformations could distort the underlying audio characteristics encoded in the scalogram.

To address this, we adopted a data augmentation technique directly applied to the time axis of the audio signal before scalogram generation. Specifically, we performed time stretching by expanding the signal (i.e., decreasing the playback speed) using randomly selected stretching factors. Unlike previous work [15], which applied both compression and expansion to long-duration audio excerpts, this study focused exclusively on expansion. This decision was motivated by the fixed length of the GTZAN dataset samples (30 seconds). Applying compression would result in reduced signal length and require artificial padding, which could introduce undesirable artifacts.

The augmentation process involves applying a stretching factor fa selected randomly from the set $\{1.0, 1.1, 1.2, 1.3\}$. After stretching, a random 30-second window is extracted from the resulting signal, preserving the original input duration. Figure 3 shows a scalogram with dimensions of 300×300 pixels, followed by the same scalogram after stretching with a factor of $fa = 1.3$. The green window represents the region of pixels that will be used during network training. This ensures that the network is exposed to meaningful variations in temporal structure while maintaining consistent input dimensions. The scalograms are then generated from these modified audio excerpts using the same parameters as the original ones.

This strategy allows the model to learn from variations in temporal resolution and rhythmic patterns without altering the frequency structure, providing a richer and more robust representation space for musical genre classification.

IV. Results

After training all architectures, a similar behavior was observed across the different experiments. Table I presents the results of the experiments. The first column lists the four evaluated architectures, separated by the input type used — either mel-spectrogram or mel-scalogram. The second and third columns show the training and validation results in terms of accuracy and standard deviation, based on 5-fold cross-validation.

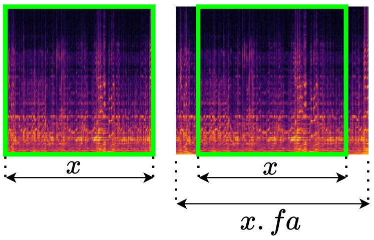


Fig. 3: Data augmentation applied during the training process. The first mel-scalogram has an original horizontal dimension of x. The green window represents the image used for training. The second mel-scalogram shows the same example after being stretched by a factor fa, and the green window indicates the portion of the image used for training.

Table I: Comparison of Accuracy between Mel-Scalograms and Mel-Spectrograms

	Training (%)	**Validation (%)**
Mel-Spectrogram		
MobileNetV2	99.6 ± 0.2	76.7 ± 4.9
EfficientNetB0	99.9 ± 0.1	83.5 ± 2.2
ResNet50	99.9 ± 0.1	83.3 ± 1.5
VGG16	98.5 ± 0.3	74.3 ± 3.6
Mel-Scalogram		
MobileNetV2	99.4 ± 0.5	74.0 ± 2.5
EfficientNetB0	99.6 ± 0.2	81.5 ± 1.5
ResNet50	99.9 ± 0.1	81.3 ± 1.1
VGG16	98.9 ± 0.1	72.6 ± 2.6

The results show that all architectures behaved similarly when the input representation was changed, with a slight advantage for the mel-spectrogram. Architectures such as MobileNetV2 and VGG16, which presented higher standard deviations, showed consistent behavior across both mel-spectrogram and mel-scalogram inputs. Interestingly, the standard deviation was lower for the experiments using mel-scalograms.

EfficientNetB0 and ResNet50 achieved the highest average validation accuracy across both input types, with ResNet50 also presenting the lowest standard deviation. For each experiment, five independent training runs were conducted. Figure 4 shows the training and validation accuracy curves for fold $k = 3$ using the ResNet50 architecture with mel-scalogram as input. The fine-tuning phase can be clearly observed starting at epoch 100, where the accuracy fluctuations are reduced and a slight improvement in the final results is achieved. This behavior was consistent across all training runs.

We also present the confusion matrix for the best-performing experiment in Figure 5. This result was obtained with EfficientNetB0 on fold $k = 2$, with 86% of accuracy, using the mel-spectrogram as input. The model correctly classified all examples of the genres metal, pop, and classical. On the other hand, rock yielded the lowest performance, often being confused with genres such as blues, country, disco, and reggae. This is understandable, as rock shares strong historical and instrumental roots with both blues and country, which

Table II: Comparison with State-of-the-Art in Music Genre Classification

Study	**Year**	**Model**	**Dataset(s)**	**Accuracy (%)**
This study	2025	CNN-EfficientNetB0	GTZAN	83.5
Ahmed et al. [6]	2024	CNN	GTZAN, ISMIR	92.7
Paghdal et al. [18]	2024	Vision Transformer	GTZAN	91.6
Ashraf et al. [19]	2023	CNN+Bi-GRU	GTZAN	89.3
Kostrzewa et al. [20]	2021	Ensemble-1 Vote	FMA	53.4
Ndou et al. [21]	2021	SVM	GTZAN, BMD	79.7
Karunakaran et al. [22]	2018	Hybrid Classifier on Spark	GTZAN, FMA	82.4

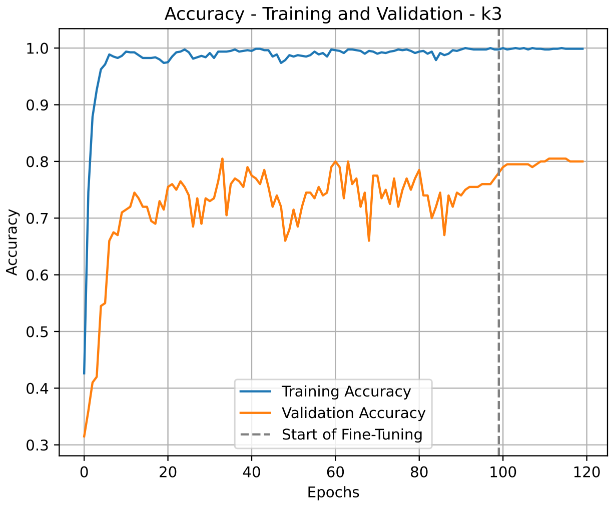
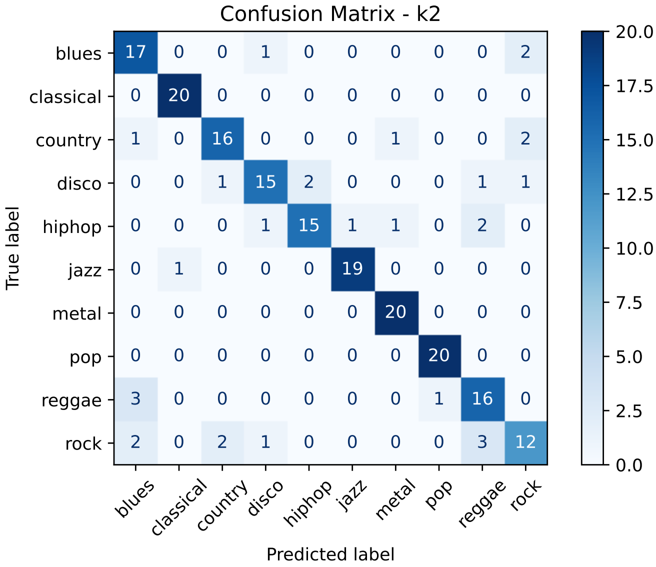


Fig. 4: Training and validation accuracy curves for ResNet50 with mel-scalogram input, fold $k = 3$. The fine-tuning phase begins at epoch 100.

likely contributed to the misclassifications by the CNN.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	17	0	0	1	0	0	0	0	0	2
classical	0	20	0	0	0	0	0	0	0	0
country	1	0	16	0	0	0	1	0	0	2
disco	0	0	1	15	2	0	0	0	1	1
hiphop	0	0	0	1	15	1	1	0	2	0
jazz	0	1	0	0	0	19	0	0	0	0
metal	0	0	0	0	0	0	20	0	0	0
pop	0	0	0	0	0	0	0	20	0	0
reggae	3	0	0	0	0	0	0	1	16	0
rock	2	0	2	1	0	0	0	0	3	12

Fig. 5: Confusion matrix for the best-performing model: EfficientNetB0 using mel-spectrogram input, fold $k = 2$.

Table II presents a comparison between our method and recent state-of-the-art approaches for musical genre classification. Although direct numerical comparison is difficult due to methodological differences, we include this table to contextualize our results in the broader literature.

Several works achieve accuracy values higher than ours, but caution must be taken when interpreting these results. For instance, [6] and [18] employ deep CNNs or transformer-based architectures trained on combinations of datasets and often rely on fixed train/test splits. In particular, [18] segment each 30-second track into six 5-second clips, increasing the size of the dataset but potentially introducing data leakage if care is not taken to ensure that segments from the same track do not appear in both training and testing sets.

A similar issue is observed in [19], where each track is split into 3-second clips. While this strategy often improves classification performance, it makes generalization harder to assess due to potential overlap of content between splits. Moreover, most of these studies do not report using cross-validation, which further limits the reliability of accuracy metrics when compared to our methodology.

In contrast, our work uses a EfficientNetB0 architecture, applied to full-length tracks, with no artificial segmentation. Although this results in a lower accuracy (83.5%), we argue that it reflects a more realistic evaluation of the model's ability to generalize to unseen musical content.

V. Conclusion

In this work, we performed a comparison between two types of input representations — mel-spectrograms and mel-scalograms — using classical image classification architectures for the task of musical genre classification. The GTZAN dataset, which is widely used in the literature, was adopted in the experiments, and techniques such as data augmentation, transfer learning, fine-tuning, and 5-fold cross-validation were applied.

The results showed that mel-spectrograms had a slight advantage in terms of average validation accuracy. However, mel-scalograms presented a lower standard deviation across training runs. The best-performing models were EfficientNetB0 and ResNet50, which reached up to 86% validation accuracy in some of the folds. From a musical perspective, the models performed well in classifying classical, metal, and pop music, while showing more difficulty in correctly classifying rock. Our results were competitive with the state of the art, even though we did not apply cropping techniques to the input audio and avoided simple training procedures without cross-validation.

As suggestions for future work, one possibility would be to combine mel-spectrogram and mel-scalogram representations into a single input, to investigate whether mel-scalograms can complement the features extracted from mel-spectrograms. Another direction would be to explore different architectures in order to identify an optimal model for the task of musical genre classification.

Acknowledgments

The authors express their gratitude to CNPq and CAPES for the financial support.

References

[1] Tzanetakis, G. and Cook, P., "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. doi: https://doi.org/10.1109/TSA.2002.80056010.1109/TSA.2002.800560.

[2] Moysis, L., Iliadis, L. A., Sotiroudis, S. P., Boursianis, A. D., Papadopoulou, M. S., Kokkinidis, K.-I. D., Volos, C., Sarigiannidis, P., Nikolaidis, S., and Goudos, S. K., "Music Deep Learning: Deep Learning Methods for Music Signal Processing—A Review of the State-of-the-Art," *IEEE Access*, vol. 11, pp. 17031–17052, 2023. doi: https://doi.org/10.1109/ACCESS.2023.324462010.1109/ACCESS.2023.3244620.

[3] Y. Ding and A. Lerch, "Audio Embeddings as Teachers for Music Classification," in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

[4] B. Jandoubi and M. A. Akhloufi, "Deep Multimodal Classification of Musical Genres," in *Proceedings of the IEEE SoutheastCon 2025*, pp. 384–389, 2025. doi: https://doi.org/10.1109/SOUTHEASTCON56624.2025.1097147710.1109/SOUTHEASTCON56624.2025.10971477.

[5] O. E. Oguike and M. Primus, "Multimodal Music Genre Classification of Sotho-Tswana Musical Videos," *IEEE Access*, vol. 13, pp. 28799–28808, 2025. doi: https://doi.org/10.1109/ACCESS.2025.353602610.1109/ACCESS.2025.3536026.

[6] M. Ahmed, U. Rozario, M. M. Kabir, Z. Aung, J. Shin, and M. F. Mridha, "Musical Genre Classification Using Advanced Audio Analysis and Deep Learning Techniques," *IEEE Open Journal of Computer Society*, vol. 5, pp. 457–467, 2024. doi: https://doi.org/10.1109/OJCS.2024.343122910.1109/OJCS.2024.3431229.

[7] E. F. Simas Filho, E. A. Borges Jr., and A. C. L. Fernandes Jr., "Genre Classification for Brazilian Music using Independent and Discriminant Features," *Journal of Communication and Information Systems*, vol. 33, no. 1, 2018. Available: https://doi.org/10.14209/jcis.2018.11

[8] S. Allamy and A. L. Koerich, "1D CNN Architectures for Music Genre Classification," *arXiv preprint arXiv:2105.07302*, 2021.

[9] Q. Kong, X. Feng, and Y. Li, "Music Genre Classification Using Convolutional Neural Network," in *Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2014.

[10] W.-H. Hsu, B.-Y. Chen, and Y.-H. Yang, "Deep Learning Based EDM Subgenre Classification using Mel-Spectrogram and Tempogram Features," *arXiv preprint arXiv:2110.08862*, 2021.

[11] H. Chen, P. Zhang, H. Bai, Q. Yuan, X. Bao, and Y. Yan, "Deep Convolutional Neural Network with Scalogram for Audio Scene Modeling," in *Proceedings of Interspeech 2018*, pp. 3304–3308, 2018. doi: https://doi.org/10.21437/Interspeech.2018-152410.21437/Interspeech.2018-1524.

[12] L. A. G. Viana, "Tempo Estimation Using Combined Mel-Spectrogram and Mel-Scalogram," in *Proc. of the 1st Latin American Music Information Retrieval Workshop (LAMIR)*, 2024, pp. 29–33. Doi: https://doi.org/10.5281/zenodo.1490804010.5281/zenodo.14908040.

[13] *Large Scale Visual Recognition Challenge 2016 (ILSVRC2016)*, UNC Vision Lab, Disponível em: http://image-net.org/challenges/LSVRC/2016/index. Acesso em: 07 março, 2021.

[14] Kah Liang Ong, Chin Poo Lee, Heng Siong Lim, Kian Ming Lim, and Ali Alqahtani. "Mel-MViTv2: Enhanced Speech Emotion Recognition With Mel Spectrogram and Improved Multiscale Vision Transformers". *IEEE Access*, 11, 108571-108579, 2023. doi: 10.1109/ACCESS.2023.3321122

[15] L. A. G. Viana, A. C. L. Fernandes Júnior, and E. F. de Simas Filho, "Estimativa de Andamento Musical Através de Escalogramas Wavelet e Redes Neurais Convolucionais," in *Anais do XVI Congresso Brasileiro de Inteligência Computacional (CBIC'2023)*, Salvador, BA, pp. 1–8, 2023. doi: https://doi.org/10.21528/CBIC2023-147.

[16] M. Domingues, O. Mendes, M. Kaibara, V. Menconi, E. Bernardes. "Exploring the continuous wavelet transform". *Revista Brasileira de Ensino de Física*, 2016. doi: 10.1590/1806-9126-RBEF-2016-0019.

[17] B. McFee, C. Raffel, D. Liang, et al., "librosa: Audio and Music Signal Analysis in Python,"in Proceedings of the 14th Python in Science Conference (SciPy 2015), 2015, pp. 18-24. doi: 10.25080/Majora-7b98e3ed-003.

[18] Paghdal, B., Kushwah, S., Kishor, A., Singh, P. K., and Kumar, A., "Melspectrogram Based Music Genre Classification System Using Vision Transformer," in *Proceedings of the 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT)*, pp. 1–6, 2024. doi: 10.1109/INSPECT63485.2024.10896162.

[19] Ashraf, M., Abid, F., Din, I. U., Rasheed, J., Yesiltepe, M., Yeo, S. F., and Ersoy, M. T., "A Hybrid CNN and RNN Variant Model for Music Classification," *Applied Sciences*, vol. 13, no. 3, 2023, Art. no. 476. doi: 10.3390/app13031476.

[20] Kostrzewa, D., Kaminski, P., and Brzeski, R., "Music Genre Classification: Looking for the Perfect Network," in *Proceedings of the International Conference on Computational Science (ICCS)*, 2021, pp. 55–67. doi: 10.1007/978-3-030-77961-0_6.

[21] Ndou, N., Ajoodha, R., and Jadhav, A., "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches," in *Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–6. doi: 10.1109/IEMTRONICS52119.2021.9422487.

[22] Karunakaran, N., and Arya, A., "A Scalable Hybrid Classifier for Music Genre Classification Using Machine Learning Concepts and Spark," in *Proceedings of the 2018 IEEE International Conference on Intelligent Autonomous Systems (ICoIAS)*, 2018, pp. 128–135. doi: 10.1109/ICoIAS.2018.8494161.

[23] A. Géron. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools and techniques to build intelligent Systems". *O'Reilly Media, Sebastopol*, CA, 2019.

[24] L. Shao, F. Zhu, and X. Li, "Transfer Learning for Visual Categorization: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015. doi: https://doi.org/10.1109/TNNLS.2014.2330900.

[25] Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and Chen, Liang-Chieh, "Mobilenetv2: Inverted residuals and linear bottlenecks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[26] Tan, Mingxing and Le, Quoc., "Efficientnet: Rethinking model scaling for convolutional neural networks", *International conference on machine learning*, pp. 6105–6114, 2019.

[27] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Deep residual learning for image recognition", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[28] Simonyan, Karen and Zisserman, Andrew, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[29] Tan, Mingxing and Chen, Bo and Pang, Ruoming and Vasudevan, Vijay and Sandler, Mark and Howard, Andrew and Le, Quoc V, "Mnasnet: Platform-aware neural architecture search for mobile", *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.

[30] Szegedy, Christian and Liu, Wei and Jia, Yangqing and Sermanet, Pierre and Reed, Scott and Anguelov, Dragomir and Erhan, Dumitru and Vanhoucke, Vincent and Rabinovich, Andrew, "Going deeper with convolutions", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 , 2015.