Fusion of Multiple Audio Representations for Music Genre Classification

Diego Furtado Silva[1] **, Micael Valterlânio da Silva**[1] **,**
Ricardo Szram Filho[1] **, Angelo Cesar Mendes da Silva**[2]

[1]Departamento de Computação – Universidade Federal de São Carlos, Brazil
Rod. Washington Luís, km 235, Jardim Guanabara – 13565-905 São Carlos, SP

[2]Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, Brazil
Av Trabalhador São-carlense, 400, Centro – 13566-590 São Carlos, SP

diegofs@ufscar.br, micael.sax@hotmail.com, ricardo.szram@estudante.ufscar.br, angelo.mendes@usp.br

Abstract. *Music classification is one of the most studied tasks in music information retrieval. Notably, one of the targets with high interest in this task is the music genre. In this scenario, the use of deep neural networks has led to the current state-of-the-art results. Research endeavors in this knowledge domain focus on a single feature to represent the audio in the input for the classification model. Due to this task's nature, researchers usually rely on time-frequency-based features, especially those designed to make timbre more explicit. However, the audio processing literature presents many strategies to build representations that reveal diverse characteristics of music, such as key and tempo, which may contribute with relevant information for the classification of genres. We showed an exploratory study on different neural network model fusion techniques for music genre classification with multiple features as input. Our results demonstrate that Multi-Feature Fusion Networks consistently improve the classification accuracy for suitable choices of input representations.*

1. Introduction

With the active growth of digital music distribution, mainly due to streaming platforms' success[1], organizing and retrieving musical information through computational methods has become an increasingly relevant task in both academia and industry. While the volume of data associated with music collections becomes more abundant and diverse, providing an excellent digital music experience becomes a more complex task.

Looking at improving this experience, researchers have proposed many methods for different tasks of Music Information Retrieval. Notably, machine learning has gained significant attention in this area, especially in classification tasks. The genre represents one of the most common ways of labeling music recordings. With this information, online music platforms can better organize artists and songs with similar characteristics. This organization has implications for efficient information retrieval in recommendation systems, among other tasks [1, 2, 3, 4, 5].

Most of the research on machine learning in the music domain relies on music content data, i.e., it only considers the audio recording as the input for model learning [6, 7, 8]. Usually, research in this knowledge domain is limited to one input feature. For example, the literature on genre classification is mainly based on the use of representations that highlight timbre or rhythm characteristics [9, 10].

However, the music genre's subjectivity makes its definition quite difficult, even for humans [11]. Music from different genres may sound similar in many ways. For example, rock and blues are performed by bands with similar compositions, with the same musical instruments, and may use similar melodic constructions.

Researchers have attempted to mitigate this obstacle in recent years by using alternative data sources related to music to be categorized. One example is the multimodal classification, where different data modalities are combined, such as audio, image, and text, to further improve the result obtained [4, 12, 13, 14]. In addition, when using different input data, the results obtained are usually more accurate than those obtained from a single input [15].

Despite being a good approach and having a wide range of practical applications, different data modalities are not always available for the same song. Furthermore, improving models based on audio can also assist in the multimodal classification task's final result.

Unlike ensemble-based models, such as voting and stacking, model fusions are designed to work as an end-to-end homogeneous method to deal with multiple inputs or different transformations of the same data. Tasks that rely on multimodal learning, such as multimedia information retrieval, are good examples of fusion-based model applications [16, 17].

The two primary techniques for fusing models are early and late fusion. The main difference between them is the stage the models are fused. Early fusion techniques first extract characteristics from each input, aggregate them into a single vector, and then use a classifier to represent the combined features. In late fusion methods, each input is processed and presented to a classification model, which outputs scores for each class. Next, the outputs of those classifiers are concatenated and then used as input features to another classifier. Figure 1 illustrates the difference between these approaches.

[1]IFPI issues Global Music Report 2021 - IFPI. Available in https://www.ifpi.org/ifpi-issues-annual-global-music-report-2021

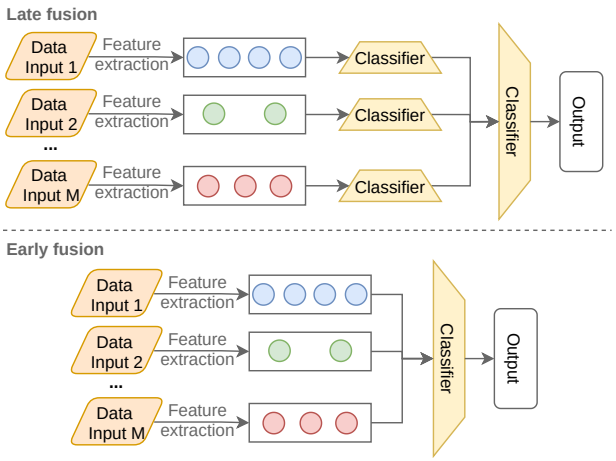

Figure 1: Illustration of late and early fusion techniques for M data inputs.

Although we can use fusion methods of machine learning techniques, deep neural networks form a convenient set of tools in this context since deep neural networks implicitly extract features based on the input data. In the case of music content, these inputs may be different audio representations of the same recording. The intuition behind using diverse representations relies on the fact that some audio processing may "hide" some interesting characteristics in a feature set that may appear in another.

In this work, we investigate ways to improve the content-based classification of music genres by combining models. For this, we use the fusion of classifiers trained on different audio representations, such as melspectrogram, constant-Q spectrum, and chroma. All models used in this work are deep learning-based networks, which means that the proposed techniques can be easily adapted to other tasks, given they extract new rich features according to the input data. Therefore, we named the general technique explored in this works as Multi-Feature Fusion Networks (MUFFN, pronounced as "muffin").

The obtained results show that MUFFN constantly leads to more accurate results than individual models. However, some fusion models reduce accuracy, depending on the representations used and how they are combined. Therefore, this work explores different possibilities for constructing MUFFNs and presents discussions to help other researchers examine similar techniques.

2. Related Work

The literature on content-based music genre classification is considerably vast [18, 19, 7]. According to [20], the evolution of audio feature manipulation for music classification is evident when we analyze the related literature. For example, in the first efforts in this area, the genre classification was usually performed by extracting a single set of features from the whole recording and using them as input for traditional machine learning algorithms [21]. Afterward, researchers found that dividing the recordings into small frames, classify each frame individually, and provide the final answer according to the most commonly observed label could lead us to more exciting results [22]. However, [20] approached this task using algorithms capable of aggregating frame-level features to make a global decision' for the classification.

The notion that the better the feature manipulation, the better the classification results led to the application of deep learning techniques [23], which defines the state-of-the-art of music classification [9, 24]. A convolutional neural network, for instance, is capable of learning rich features from unstructured data, such as music. A usual approach uses a time-frequency-based representation of the audio as the input for a deep neural network.

However, some researchers have noted that using a single representation may not be the most appropriate way to solve some Music Information Retrieval (MIR) tasks. This observation is not limited to classification. For instance, the fusion of different features was successfully used in cover song identification [25]. On the other hand, in genre classification, some authors used traditional heterogeneous [26, 27] and homogeneous [28] ensemble techniques to accomplish the task.

Outside the task of content-based music classification, fusion models are prevalent in the multimodal classification literature [16], including music genre classification. [4] used model fusion to classify albums' genre by their tracks (content/audio-based), its cover (image-based), and reviews on Amazon.com (text-based).

3. Multi-Feature Fusion Networks

The purpose of this work is not to create a new algorithm or neural network architecture but to discuss the multiples audio features and model relevance when used as a complementary way to discriminate a song. Despite recent work, studies like this are underexplored in the MIR community. Besides, we construct the first model for music genre classification based on this concept. Specifically, we propose Multi-Feature Fusion Networks (MUFFN) applied in music classification problems. For this, we investigate different representations and deep neural network architectures for the classification of musical genres and model fusion techniques, aiming to improve accuracy compared to individual models.

4. Experimental Setup

Since MUFFN is a general framework for creating classification models, this work's experimental setup needs to cover various procedures. This section defines the procedures adopted to guide the search for a suitable MUFFN model for classifying musical genres. In other words, we present the datasets used, the parameters of the training phase, the decisions behind the base neural network's choice – including a brief ablation study –, decisions on audio representations for the input, and details of the model's fusion.

For the sake of reproducibility, we created a code repository for this work[2], containing source code and links

[2]https://github.com/diegofurts/muffn

to all datasets. The repository also comprises additional results and details of our proposal.

4.1. Training Procedure

We established a simple but definite training phase procedure to fairly compare models trained in a single representation and MUFFN-based models. First, we trained all the models using 5-fold cross-validation. Then, for each training fold, we separated 10% of the data for validation. We used fixed seeds for random steps in all cases, so the examples in each fold keep the same for every trained model.

We trained each model using at most 50 epochs, saving the one with the lowest loss in the validation set to test later. Besides, we applied early stopping and reducing the learning rate in plateaus to monitor the validation loss. The early stopping was set with the patience of 20 epochs (to avoid abandoning the training in the first epoch).

We defined the batch size according to memory limitations[3]. Specifically, we used batches of 16 examples for the two smaller datasets and only 8 for the two largest ones.

4.2. Datasets

To assess the investigated models, we used four datasets widely used in music genre recognition research: GTZAN[4], Music Audio Benchmark Dataset (Homburg) [30], Extended Ballroom [31], and the small subset of FMA [32]. These datasets have different dimensions and a different number of labels. Table 1 summarizes the datasets.

Table 1: Summarized description of the datasets used in this work.

	Dataset			
	GTZAN	Homburg	E. Ballroom	FMA
# Recordings	1000	1886	4180	8000
# Genres	10	9	13	8
Audio duration (s)	30	10	30	30
Min. tracks/genre	100	47	23	1000
Max. tracks/genre	100	504	529	1000

We note that we applied the same experimental procedure in all datasets. Although FMA has a default train/test partition, we decided to use the same cross-validation technique for empirical consistency and better analyze the algorithms' stability in slightly different data distributions.

4.3. Audio Representations

We considered a wide range of features designed to reveal the recordings' distinct features to assess the fusion of models trained on different audio representations. However, we acknowledge that many datasets can be composed of copyright-protected songs and, therefore, only distribute a set of features extracted from the original data [33]. Thus, we did not consider using representations used for the exact reconstruction of the original audio.

To achieve these requirements, we used the Python library librosa [34] to extract melspectrograms, mel-frequency cepstrum coefficients (MFCCs), constant-Q transformed spectrograms (CQT), tempograms, tonnetz, and the energy at the harmonics of the spectrum. Besides, we assumed that structure also performs a significant role in the music genre (e.g., pop music is usually more repetitive than jazz), so we also applied self-similarity matrices on the chromagrams.

We kept most of the librosa's default parameters for the feature extraction process. However, for the sake of dimensionality reduction and consequent better memory efficiency, we set the hop length to 1024 (the default is 512). For the same reason, we set the size of the onset auto-correlation window for the tempogram as 128. Finally, after extracting the melspectrograms, we applied a log transformation.

4.4. Deep Neural Models

Although learning algorithms based on deep neural networks present relevant results, we notice that proposing a new neural network for music genre classification is not in this work's scope. Therefore, we decided to rely on papers that presented deep learning-based methods for music classification. Initially, we experimented with models pretrained on other domains, such as VGG16. Additionally, we considered the Convolutional Recurrent Neural Network (CRNN). However, we would need to modify the networks to deal with different representations and datasets in both cases because these architectures rely on several dimensionality reduction layers (max pooling). Thus, in some scenarios, the data dimensionality does not apply to these neural networks without losing the focus on the evaluating of different representations and fusion techniques.

Therefore, we need an architecture that is less sensitive to differences in the data dimensions. The Bottom-up Broadcast Network (BBNN) meets this requirement [35]. Besides, we experimented with CRNN, VGG16, and BBNN on the GTZAN dataset, using the melspectrogram as an input feature. BBNN achieved the best results in this experiment. Considering this, we proceed using the BBNN exclusively.

Figure 2 illustrates the BBNN. It starts with a simple convolution network, which the authors call Shallow Feature Extraction Layer (SFEL). It follows with three inception blocks, being that the output of SFEL is concatenated with each inception's output before serving as input for the next layer. These layers were named Broadcast Module (BM). Next, the authors use more convolutional and pooling layers, comprising the Transition (TL) and Decision (DL) layers. The global average pooling in the DL creates a 32-dimensional feature set used by the softmax layer to deliver a final decision. We refer the reader interested in a detailed description of the BBNN to the paper that proposes this architecture [8]. Besides, we make a

[3] We built our models using Keras [29], and we used Google Colab to execute our experiments because of the low resources available. For the same reason, we do not provide a runtime experiment.

[4] `http://marsyas.info/downloads/datasets.html`

better description of the network available in the paper's repository.

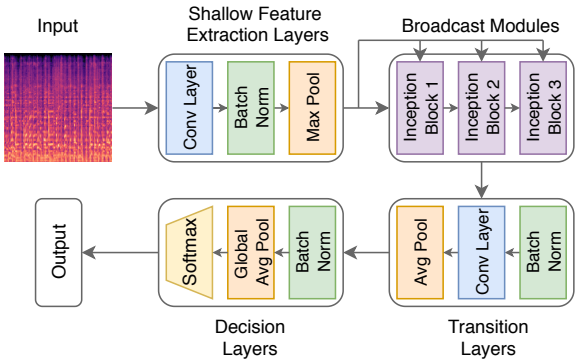

Figure 2: Simplified architecture of the Bottom-up Broadcast Network.

4.5. Ablation Study

Although finding the best neural network is not the focus of this paper, we consider it adequate to perform a short ablation study on the selected architecture. To avoid a significant overhead of this phase, we only conducted this study on GTZAN with MFCCs – given the good results on individual classifiers (c.f. Section 5) and the fact that this is a low-dimensional representation – as input. Also, we used the 60-20-20 split for training, validation, and test sets. Note that it implies a slightly smaller training set than the other experiments. Therefore, the training phase is marginally faster, and we can better understand the effects on validation and test.

For this, we realized experiments using one, two, and all three inception blocks of BBNN. Besides, we note that these blocks have four parallel sequences of layers (c.f. [8] for details). Therefore, we experimented with the network with all inception blocks but removed one of these sequences, composing four new networks.

We note that the BBNN is notably prone to overfitting in all experiments performed in this paper. Achieving perfect accuracy in the first 20 training epochs is common. On the other hand, the model achieves much lower results in the validation and test sets. However, the loss function usually keeps decreasing in both training and validation.

In all ablation experiments, this effect was reduced. As a result, the epoch that achieves the perfect score for the training set occurs later. However, it does not reflect on better generalization and, consequently, better accuracy. In all cases, the complete network achieves the best result.

For example, while the complete network's accuracy is 0.799, removing one or two inception blocks leads to 0.784 and 0.776, respectively. In addition, the accuracy obtained when we removed one of the inception paths varied from 0.773 to 0.793. Although these differences are not significant, we decided to keep using the complete architecture.

4.6. Early and Late Fusion

Once we have defined the neural network to use, we need to define which points to concatenate to build a single model. In the case of late fusion, each input must pass through its entire BBNN. The output of each input's softmax layer is concatenated and then used as an input for another softmax layer for the final decision. Figure 3 illustrates this approach.

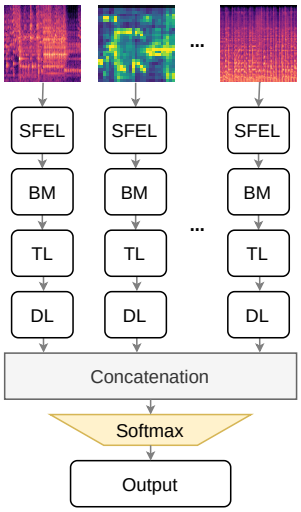

Figure 3: Late fusion scheme when using BBNN-based MUFFN.

We must define a layer that represents the features extracted by the neural network for early fusion models. These features can be obtained after any set of layers illustrated in Figure 2 or after each inception block. However, as shown in the ablation study, we noticed that using all layers of BBNN leads to better results. Therefore, we use the features aggregated by the global average pooling layer. Figure 4 illustrates the approach to building models using early fusion.

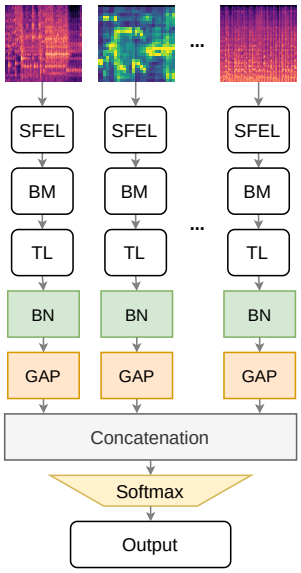

Figure 4: Early fusion scheme when using BBNN-based MUFFN.

It is important to note that we first train individual models and store their weights to train the fusion models. From that, we use these pre-trained models to build MUFFN models. We then evaluate data fusion methods with fixed weights and with fine-tuning the weights through model retraining. In the latter case, we only allow the training procedure to adjust the weights of the transition and decision layers.

5. Results

This section presents the classification results, following the experimental setup introduced in Section 4.

5.1. Individual Classifiers

We started the presentation of results by the individual classifiers. We note that, as mentioned in Section 4.3, we discarded some of the representations considered in this work for preliminary experiments. For example, the accuracy obtained using tonnetz was 0.113 in the GTZAN dataset. On the other hand, the use of self-similarity matrices led to 0.310 of accuracy in the Homburg dataset, probably due to the recordings' short duration.

Table 2 shows the results obtained by using different representations in all datasets.

Table 2: Accuracy and standard deviation obtained by individual classifiers.

Feature	GTZAN	Homburg	E. Ballroom	FMA
Chroma	0.696±0.046	0.536±0.043	0.878±0.012	0.483±0.039
CQT	0.785±0.013	0.639±0.033	0.932±0.006	0.643±0.010
MFCC	0.774±0.037	0.585±0.029	0.926±0.013	0.619±0.024
Melspec	**0.851±0.022**	**0.644±0.031**	**0.941±0.006**	**0.654±0.016**
Tempo	0.478±0.028	0.453±0.026	0.886±0.088	0.432±0.011

We note that the melspectrogram was the best representation in all datasets. In some cases, especially in the GTZAN dataset, this difference is substantial compared to the second-best result. This observation is aligned with the most common assumption in the literature that this is the best representation for the content-based music classification task. Moreover, we observe that using the constant-Q transform consistently leads to the second-best results, and the MFCC achieves accuracy rates close to those obtained by CQT.

5.2. Fusion Models

In this section, we gradually built the base to arrive at the final MUFFN models. For every decision, we first evaluate if it seems promising in the smaller datasets, i.e., GTZAN and Homburg. Then, if it presented good results, we replicated the decision to the other datasets.

We started our decision by assuming that the melspectrogram should be part of all assessed subsets of representations. Then, we used this assumption to create two double-input fusion networks: melspectrogram and MFCC; and melspectrogram and chroma. In the first case, we evaluate if two representations based on the same premise of applying the mel scale improve timbre representation. In the latter case, we assess if tonality- and timbre-related features may complement each other. Besides, we added the tempogram as a third input to the second approach, creating a model that fuses tonality, timbre, and tempo features. We used these three combinations in the first batch of experiments.

We note that we evaluated early and late fusion schemes and training with and without fine-tuning, as described in Section 4.6. In all cases being assessed, early fusion methods achieved better results than late fusion. This phenomenon is more evident when we allow fine-tuning. In some cases, fine-tuning decreases the accuracy of late fusion models. On the other hand, it constantly improves the results of early fused networks. For this reason, we continue our analysis considering only early-fused and fine-tuned models.

Figure 5 illustrates this fact in the specific case of GTZAN with MFCC and melspectrogram as inputs.

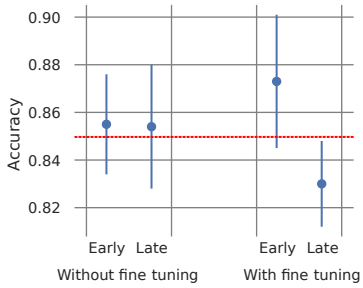

Figure 5: Comparison between early and late fusion, with and without fine tuning, of melspectrogram and MFCC in the GTZAN dataset. The red dashed line shows the accuracy of the best individual model.

Note that the figure results are mostly better than the individual classifier, which also happens on Homburg data. For this reason, we also evaluated this fusion in the remaining datasets. However, the same observations do not apply to the other described fusion. The only exception is the fusion of melspectrogram, chroma features, and tempogram with Homburg data. In this case, the MUFFN achieved 0.002 of accuracy improvement. On the other hand, on GTZAN, the same architecture caused an accuracy reduction of 0.023.

We realized that merging the CQT spectrum caused improved results from the successful merger in the previously described step. Therefore, for this new merger, we evaluated the effects of adding the chromagram or the tempogram. Finally, we evaluate adding these last two representations simultaneously, creating a fusion model of five audio representations.

For a better presentation and interpretation, Table 3 summarizes these fusions and attributes short identifiers to each of them.

Figure 6 shows the accuracy obtained for these models and the best individual one, obtained with melspec-

Table 3: Subsets of audio representations used to create MUFFN models.

Identification	Subset of input representations
MM	Melspectrogram + MFCC
MMC	MM + CQT spectrum
MMCC	MMC + chromagram
MMCT	MMC + tempogram
MMCCT	MMC + chromagram + tempogram

trogram.

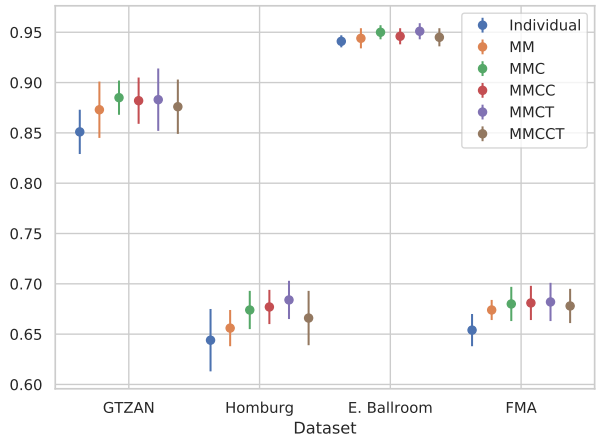

Figure 6: Classification results using individual and fusion models.

For further interpretation and analysis, Table 4 presents the same results, but numerically.

Table 4: Accuracy and standard deviation obtained by the best individual and fusion-based classifiers.

Model	GTZAN	Homburg	E. Ballroom	FMA
Individual	0.851±0.022	0.644±0.031	0.941±0.006	0.654±0.016
MM	0.873±0.028	0.656±0.018	0.944±0.010	0.674±0.010
MMC	**0.885±0.017**	0.674±0.019	0.950±0.007	0.680±0.017
MMCC	0.882±0.023	0.677±0.017	0.946±0.008	0.681±0.017
MMCT	0.883±0.031	**0.684±0.019**	**0.951±0.008**	**0.682±0.019***
MMCCT	0.876±0.027	0.666±0.027	0.945±0.009	0.678±0.017

We note that, in general, adding other representations can improve the results. In contrast, the two representations that add chroma features achieved lower accuracy in most cases than the respective version without such a representation. It is easily observed by comparing MMC against MMCC and, especially, MMCT against MMCCT.

6. Discussion

The results presented show that there is great potential for improving content-based music classification. We recall exploring different neural network architectures with a single representation to find a robust model in this scenario. It means that our experiments showed that the use of MUFFN could improve the classification even compared to a powerful competitor.

Unfortunately, that gain comes at a cost. Aggregating different neural networks increases the number of parameters to be trained, generating a higher computational cost. An advantage in this sense that we observed in our experiments is that we achieved good results without training networks from scratch. The fine-tuning strategy used does not extend across the entire network but only to a reduced set of final layers of each model.

The number of parameters that BBNN needs to train for a single-representation model is $185,642$. Besides, it has $5,184$ non-trainable parameters. For example, the number of parameters in MUFFN with four distinct inputs and an early fusion approach is $763,274$. However, only $54,922$ of them are trainable. Therefore, if we intended to train the model from scratch, we would need to adjust $742,538$ of these parameters.

One way to reduce overhead for the design and training of a MUFFN is to choose the appropriate inputs. We have not yet carried out an in-depth study in this regard due to its scope. However, it was possible to observe some patterns. For example, when adding chromagram as one of the inputs, the accuracy tendency is to maintain or decrease. For instance, the accuracy obtained by MMCCT is consistently lower than MMCT. However, the same does not happen with the insertion of tempograms. This aspect probably appears because the tempogram adds complementary information to representations based on the mel scale, which does not occur for chroma features.

With these results, we believe that using MUFFN can significantly improve the results of specific tasks. For example, consider the task of hierarchical classification. It is intuitive to think that representations associated with timbre are sufficient at a higher hierarchy level. The instruments used in classical and popular music tend to be quite different, for example, in a melspectrogram. However, at a level that we need to differentiate between genres that use similar instruments and techniques, other representations can be used to "disambiguate" decisions.

For these reasons, we believe that the obtained results are motivating for future work in this direction.

7. Final Remarks

As we pointed out a few times during this work, our work presents a general framework that can be modified in several aspects. Furthermore, we demonstrate that MUFFN models can improve music genre classification accuracy in all assessed datasets with a few options evaluated. Therefore, we hope that our work will seed other research endeavors to create models based on MUFFN.

We note that we have not studied the contribution of each input in this work. Thus, despite discussing some clues, we cannot provide clear interpretations to guide the choice of representation inputs. Therefore, we intend to conduct an in-depth study to understand this aspect better and then provide more precise guidelines for the design of MUFFN for MIR tasks.

Furthermore, we acknowledge that neural networks may automatically learn features from the melspectrogram, such as MFCC or CQT, for a sufficiently large dataset. However, it depends on many factors, including the network architecture. As MUFFN is a framework, we intend to evaluate how these observations stand in different scenarios.

As a future direction, we intend to evaluate other neural network architectures as the basis for the fusion model. We limited ourselves to only one architecture in this work to avoid missing the central proposal's focus. However, we believe that different architectures can achieve better results for each diverse input representation and, consequently, for the fused model. Furthermore, as we acknowledge our proposal adds complexity to the models, we intend to focus on more efficient architectures and training strategies.

Finally, we intend to use MUFFN in other MIR tasks, such as tagging, emotion recognition, content-based similarity, and genre identification in multi-label and hierarchical classification scenarios.

8. Acknowledgments

This work was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code PROEX-12049601/D, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Finance Code 2019/08653-0.

References

[1] Paul Lamere. Social tagging and music information retrieval. *Journal of new music research*, 37(2):101–114, 2008.

[2] Peter Knees and Markus Schedl. Music retrieval and recommendation: A tutorial overview. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136. ACM, 2015.

[3] Yan Wan. *Deep Learning for Music Classification*. PhD thesis, Hong Kong University of Science and Technology, 2016.

[4] Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval*, 1(1):4–21, 2018.

[5] Sandy Manolios, Alan Hanjalic, and Cynthia CS Liem. The influence of personal values on music taste: towards value-based music recommendations. In *ACM Conference on Recommender Systems*, pages 501–505, 2019.

[6] Markus Schedl, Emilia Gómez, Julián Urbano, et al. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3):127–261, 2014.

[7] YV Murthy and Shashidhar G Koolagudi. Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Computing Surveys*, 51(3):45, 2018.

[8] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification. *arXiv preprint arXiv:1901.08928*, 2019.

[9] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017.

[10] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.

[11] Klaus Seyerlehner. *Content-based music recommender systems: Beyond simple frame-level audio similarity*. PhD thesis, Johannes Kepler Universität Linz, 2010.

[12] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. Multimodal music information processing and retrieval: Survey and future challenges. In *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 10–18. IEEE, 2019.

[13] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

[14] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):478–493, Mar 2020.

[15] Jaehun Kim, Julián Urbano, Cynthia C. S. Liem, and Alan Hanjalic. One deep music representation to rule them all? a comparative analysis of different representation learning strategies. *Neural Computing and Applications*, 32:1067–1093, 2020.

[16] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[17] Wenwu Zhu, Xin Wang, and Hongzhi Li. Multi-modal deep analysis for multimedia. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[18] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010.

[19] Bob L Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, 2014.

[20] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

[21] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[22] Kris West and Stephen Cox. Finding an optimal segmentation for audio genre classification. In *International Society for Music Information Retrieval Conference*, pages 680–685, 2005.

[23] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6959–6963. IEEE, 2014.

[24] Jaehun Kim, Julián Urbano, Cynthia CS Liem, and Alan Hanjalic. One deep music representation to rule them all? a comparative analysis of different representation learning strategies. *Neural Computing and Applications*, 32(4):1067–1093, 2020.

[25] Christopher J Tralie. Early MFCC and HPCP fusion for robust cover song identification. In *International Society for Music Information Retrieval Conference*, pages 294–301, 2017.

[26] Loris Nanni, Yandre MG Costa, Rafael L Aguiar, Carlos N Silla Jr, and Sheryl Brahnam. Ensemble of deep learning, visual and acoustic features for music genre classification. *Journal of New Music Research*, 47(4):383–397, 2018.

[27] Rodolfo M Pereira, Yandre MG Costa, Rafael L Aguiar, Alceu S Britto, Luiz ES Oliveira, and Carlos N Silla. Representation learning vs. handcrafted features for music genre classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[28] Deepanway Ghosal and Maheshkumar H Kolekar. Music genre recognition using deep neural networks and transfer learning. In *Interspeech*, volume 2018, pages 2087–2091, 2018.

[29] Francois Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

[30] Helge Homburg, Ingo Mierswa, Bulent Moller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *International Society for Music Information Retrieval Conference*, pages 528–531, 2005.

[31] Ugo Marchand and Geoffroy Peeters. The extended ballroom dataset, 2016.

[32] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. In *International Society for Music Information Retrieval Conference*, pages 316–323, 2017.

[33] Wenqin Chen, Jessica Keast, Jordan Moody, Corinne Moriarty, Felicia Villalobos, Virtue Winter, Xueqi Zhang, Xuanqi Lyu, Elizabeth Freeman, Jessie Wang, Sherry Kai, and Katherine M. Kinnaird. Data usage in mir: History & future recommendations. In *International Society for Music Information Retrieval Conference*, pages 25–32, 2019.

[34] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

[35] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification. *Multim. Tools Appl.*, 80:7313–7331, 2021.