Combining Reduction and Dense Blocks for Music Genre Classification

Charbel El Achkar ✉[1,2], Raphaël Couturier[2], Talar Atéchian[1], and Abdallah Makhoul[2]

[1] TICKET Lab, Antonine University (UA), Baabda, Lebanon
{charbel.elachkar,talar.atechian}@ua.edu.lb
[2] FEMTO-ST Institute, CNRS, Université Bourgogne Franche-Comté (UBFC), Belfort, France
{raphael.couturier,abdallah.makhoul}@univ-fcomte.fr

Abstract. Embedding music genre classifiers in music recommendation systems offers a satisfying user experience. It predicts music tracks depending on the user's taste in music. In this paper, we propose a preprocessing approach for generating STFT spectrograms and upgrades to a CNN-based music classifier named Bottom-up Broadcast Neural Network (BBNN). These upgrades concern the expansion of the number of inception and dense blocks, as well as the enhancement of the inception block through reduction block implementation. The proposed approach is able to outperform state-of-the-art music genre classifiers in terms of accuracy scores. It achieves an accuracy of 97.51% and 74.39% over the GTZAN and the FMA dataset respectively. Code is available at `https://github.com/elachkarcharbel/music-genre-classifier`.

Keywords: Music Genre Classification · STFT Spectrogram · CNN · Music Recommendation Systems.

1 Introduction

Modern studies found interest in building robust music classifiers to automate genre classification of unlabeled music tracks. There were diverse approaches in their feature engineering process as well as the neural network selection [1,2,3,4,5].

In this paper, we propose a custom approach for music genre classification. STFT spectrograms are generated and diversified by slicing each spectrogram into multiple slices to ensure a variety of visual representations among the same music track. Furthermore, upgrades to a state-of-the-art Convolutional Neural Network (CNN) network for music genre classification named BBNN [2] are proposed. **The contribution of this paper relies on two main improvements:** expanding the number of inception and dense blocks of the network and enhancing the inception block by implementing the reduction block B proposed in [6] instead of the existing block inspired by [7]. The proposition is evaluated through its application using the GTZAN [8] and the FMA [9] music datasets.

The remainder of this paper is organized as follows: in Section 2, we discuss the recent music related classifiers used on the two datasets. In Section 3, we present the preprocessing process in addition to the contributed upgrades. Section 4 explores the experimental results of the proposed upgrades over competitive CNN networks, followed by a conclusion and future work thoughts in Section 5.

2 Related Work

Many studies took advantage of deep learning technologies to build efficient music genre classifiers. They adapted visual-related features (audio spectrogram) to build CNNs for audio classification tasks [1,4,11]. The audio data is converted to spectrograms and used as input features to CNN classifiers. These spectrograms are the visual representation of the spectrum of frequencies of the audio signal. As mentioned in Section 1, the proposed contribution is validated through experimental results. These experiments are applied using both the GTZAN dataset [8] and the FMA dataset [9]. Thus, the most recent and relevant publications over the two datasets are presented below.

Starting with GTZAN-related publications, a framework achieved an accuracy of 93.7% over the GTZAN dataset by producing a multilinear subspace analysis. It reduced the dimension of cortical representations of music signals [10]. Further studies took profit from DNNs and CNNs to try reaching higher accuracies over music datasets. Inspired by multilingual techniques for automatic speech recognition, a multilingual DNN was used in [4] for music genre classification purposes. It was able to achieve an accuracy of 93.4% through 10-fold cross-validation over the GTZAN dataset. Several approaches used CNN-based networks but were not able to exceed the accuracy of 91% such as [1,11,12,13]. Others tried refining their results by overcoming the blurry classification of certain genres inside the GTZAN dataset. Their study did not surpass the accuracies mentioned previously [3]. After several attempts to outperform the accuracy reached in [10], three publications succeeded in using Mel spectrograms as input features to their DNNs. The use of convolutional long-short term memory-based neural networks (CNN LSTM) in combination with a transfer learning model helped in achieving an accuracy of 94.20% in [14]. As for the two remaining publications, the BBNN network proposed in [2] was able to achieve an accuracy of 93.90% by fully exploiting Mel spectrograms as a low-level feature for the music genre classification. The GIF generation method proposed in [5] was able to achieve the highest accuracy of 94.70% by providing efficient audio processing for animated GIF generation through acoustic features. Although this dataset has several faults [15], it is still the most dataset used in music genre classification use cases. These faults are taken into consideration in the preprocessing process that we will develop in later sections.

Concerning the FMA-related publications, a method of vertically slicing STFT spectrograms took place, in addition to applying oversampling and undersampling techniques for data augmentation purposes. This method achieved an F-score of 62.20% using an MLP classifier [16]. Another study trained a convolutional recurrent neural network (C-RNN) using raw audio to provide a real-time classification of FMA's music genres. It achieved an accuracy of 65.23% [17]. Motivated by FMA's challenges, an approach of two Deep Convolutional Neural Networks (DCNN) was proposed to classify music genres. The first DCNN was trained by the whole artist labels simultaneously, and the second was trained with a subset of the artist labels based on the artist's identity. This approach achieved an accuracy of 57.91% taking Mel spectrograms as input features to the DCNNs created [18]. Moreover, a method proposed in [13] took advantage of Densely Connected Convolutional Networks (DenseNet), found to be better than Residual Neural Network (ResNet) in music classification studies. It achieved an accuracy of 68.20% over the small subset of FMA.

3 Proposed Approach

In this section, the BBNN network proposed in [2] is briefly introduced. Later, the proposed approach is elaborated while mentioning the proposed upgrades to achieve higher accuracy results against the GTZAN and the FMA dataset.

As mentioned in the related work, the Bottom-up Broadcast Neural Network (BBNN) is a recent CNN architecture that fully exploits the low-level features of a spectrogram. It takes the multi-scale time-frequency information transferring suitable semantic features for the decision-making layers [2]. The BBNN network consists of inception blocks interconnected through dense blocks. The inception block is inspired by the inception v1 module proposed in [7] while adding a Batch Normalization (BN) operation and a Rectified Linear Unit activation (ReLU) before each convolution. This approach relied on generating coloured Mel spectrograms from the music tracks while providing the latter as input features to the CNN network. The spectrograms had the size of 647x128 and were used as-is for training purposes. This network was able to achieve the second-best accuracy over the GTZAN dataset (93.90%) by stacking three inception blocks with their corresponding dense connections.

3.1 Preprocessing

Spectrograms are the key to successful music genre classification using CNN-based networks. Based on the approaches mentioned in Section 2, greyscale STFT spectrograms are adopted instead of coloured Mel spectrograms. The majority of CNN-based music genre classifiers relied on Mel spectrograms since STFT spectrograms required greater GPU memory for their increased quantity of embedded features. Thus, we use STFT spectrograms in our experiments to leverage the latter increase on accuracy scores, in addition to the availability of efficient GPUs for experimental testing. Using the Sound eXchange (SOX) package, the greyscale spectrograms are generated with a size of 600x128. As expressed in Section 2, the GTZAN dataset has several faults [15]. For instance, three audio tracks were discarded while recursively generating the spectrograms using the SOX package. Each music track of the discarded ones was associated with a separate genre of the dataset. Therefore, we randomly removed a single

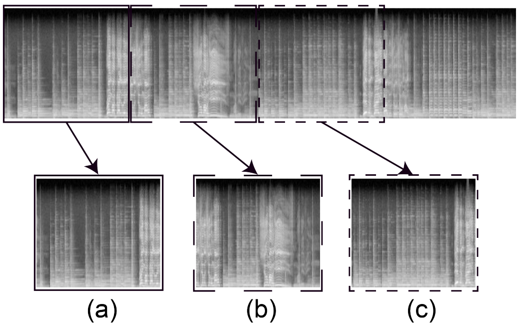

Fig. 1. Spectrogram slicing approach

audio track from the remaining genres to normalize the number of music tracks per genre.

Subsequently, the Python Imaging Libray (PIL) is used to slice the STFT spectrograms into multiple images. The spectrogram is divided into three to four separated slices. Each slice is a normalized 128x128 slice that represents a 6.4 seconds track out of the initial 30 seconds music tracks. Therefore, the last one and a half slices of the spectrogram are discarded, keeping only the first three slices (a, b and c in Fig. 1). This approach is mainly used for better data preparation for CNNs by normalizing the spectrogram's width and height. It also increases the diversity of the music genres, since spectrograms variate dependently on the time axis. Thus, this normalization does not accentuate overfitting due to the variety in every spectrogram's slices. It is important to mention that the discarded slices may hold useful data for our classification. However, we adopted this approach to limit the number of training/testing images as well as ensuring the obtention of the same number of slices per music track (music tracks length is not always consistent to 30 seconds).

3.2 Network Contribution

Inspired by the BBNN network [2], custom modifications are proposed to achieve higher accuracy results. Even though the BBNN stacks three inception blocks connected with dense blocks, the trained model possessed a tiny size (only 0.18 M). Using a small sample of both datasets, we performed a hyperparameter search taking the number of inception and dense blocks as the hyperparameter in question. The search result showed that the optimal number of blocks is equal to 6 for achieving the greatest accuracy. At this stage, the proposed network consisted of doubling the number of inception and dense blocks in the Broadcast Module (BM) of the BBNN, leaving the remaining layers (Shallow, Transition, and Decision) as proposed in [2].

Increasing the number of blocks reflected an increase in accuracy scores. On the other hand, it expanded the size of the training model and slowed the training

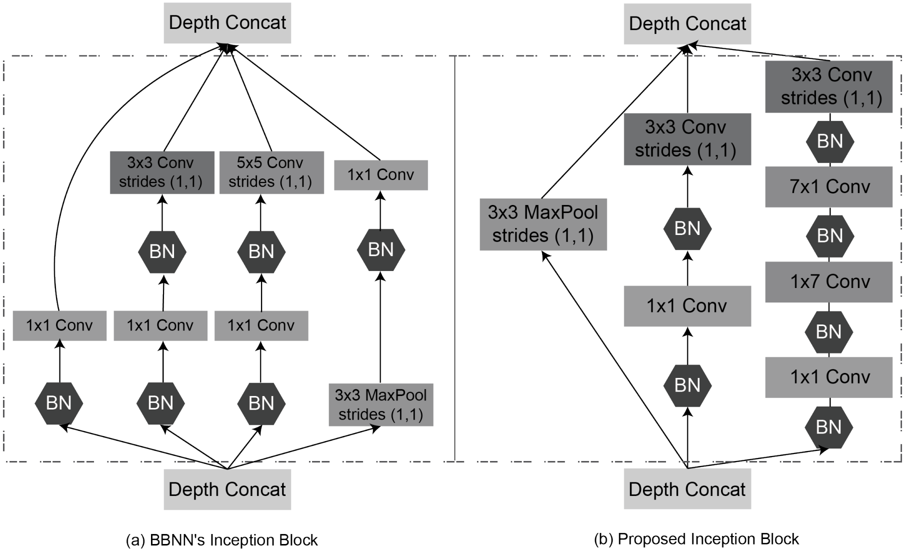

Fig. 2. Proposed inception block modifications over the BBNN network

process. Consequently, the architecture of the BBNN network was modified to reduce significant drawbacks due to overfitting and computation problems in the inception v1 block [6]. Many CNN related studies, in particular a music-related study in [13], proved that dense blocks are better than residual blocks. Thus, it was decided to keep the dense connection of the BBNN network intact. Moreover, the BBNN network relied on the inception v1 proposed in [7] while adding BN and ReLU operations before each convolution. The original inception v1 was found computationally expensive as well as prone to overfitting in many cases. At this stage, the next contribution was to replace the modified inception v1 blocks with modified inception v4 blocks in order to improve the computation efficiency and most importantly to increase the accuracy. As mentioned in [6], the earlier inception modules (v1, v2, v3) were found more complicated than necessary. They proposed specialized "Reduction Blocks" A and B to change the width and height of the grid. This change produces a performance boost by applying uniform and simplified operations to the network. Figure 2 presents the modified inception blocks in detail. The block on the left concerns the custom inception v1 block of BBNN, and the block on the right concerns our proposed inception v4 block. As previously mentioned, the left block is inspired by the inception v1 block in [7], while adding BN and ReLU operation before each convolution. On the other hand, the proposed inception block is inspired by the "Reduction Block B" introduced in [6]. Compared with BBNN's inception block in [2], the "Reduction Block B" of inception v4 [6] reduces the network complexity by mainly removing unnecessary 1x1 convolution operations and replacing the 5x5 convolution with a stack of 1x7, 7x1, and 3x3 convolution operations. Also, it accentuates memory optimization to backpropagation by implementing the factorization technique of inception v3. This technique is responsible to reduce the dimensionality of convolution layers, which reduce overfitting problems. In this matter, it was proposed to use the same architecture as the "Reduction Block B", while implementing BN and ReLU operations before each convolution.

4 Experimental Evaluation

In this section, the training hyperparameters are presented while evaluating the proposed contribution against state-of-the-art music genre classifiers. The training operations are performed using an NVIDIA Tesla V100 SXM2 GPU with 32 GB of memory.

4.1 Hyperparameters and Training Details

As mentioned in Section 3, the input images were prepared by generating a STFT spectrogram out of each music track of the GTZAN and the FMA dataset. Each spectrogram (600x128) was sliced into 128x128 slices, taking only the first three slices as a visual representation of each music track. At this stage, the input images for GTZAN classification were 297 slices of spectrograms per genre (99 music tracks per genre), and the input images for FMA classification were 3000 per genre (1000 music tracks per genre).

Inspired by BBNN [2], the proposed network upgrades were added as well as the hyperparameters to start the training. Considering that the BBNN network was initially tested against the GTZAN dataset [8], the same hyperparameters as the BBNN network were used for this case. The ADAM optimizer was selected to minimize the categorical cross-entropy between music genre labels, a batch size of 8 and an epoch size equal to 100. An initial learning rate of 0.01 was configured, while automatically decreasing its value by a factor of 0.5 once the loss stops improving after 3 epochs. The early stopping mechanism was implemented to prevent overfitting, and the GTZAN input spectrograms were fed to the classifier through 10-folds cross-validation training. Since all related publications used different dataset split ratios, the same ratio as BBNN's [2] is adopted to compare our results with BBNN in particular and with other publications in general. Thus, the training, testing and validation sets were randomly divided following an 8/1/1 proportion (80% for training, 10% for testing, and 10% for validation). The resulting training and testing accuracies were calculated by averaging all the accuracies concluded in the cross-validation folds.

Concerning the FMA dataset, the increase in the batch size revealed an accuracy increase. However, the same hyperparameters as GTZAN were used, in addition to keeping the same value of the batch size (8), to align our results with the existing ones.

Before initiating the training, the inception block's training parameters were calculated for both, the BBNN network and the proposed approach. This calculation showed that the proposed inception block uses less than 26.78 percentage points (*pp*) of BBNN's inception block parameters.

4.2 Testing Results

In the tables below (Table 1 and Table 2), the proposed approach is compared to the most recent and accurate methods. These methods either rely on deep learning models or hand-crafted feature descriptors to provide an efficient classification of the GTZAN and the FMA datasets.

Table 1. Comparative table for GTZAN classification methods in terms of accuracy (%)

GTZAN Classification		
Methods	Preprocessing	Accuracy
AuDeep[1]	Mel Spectrogram	85.40
NNet2[11]	STFT	87.40
Hybrid model[3]	MFCC, SSD, etc.	88.30
Transform learning[12]	MFCC	89.80
DenseNet+Data augmentation[13]	STFT Spectrogram	90.20
Multi-DNN[4]	MFCC	93.40
TPNTF[10]	MFCC	93.70
BBNN[2]	Mel Spectrogram	93.90
DNN+Transfer learning[14]	Mel Spectrogram	94.20
GIF generation Framework[5]	MFCC Spectrogram	94.70
Our approach	**STFT Spectrogram**	**97.51**

Table 1 compares the music genre classifiers used on the GTZAN dataset. It shows the different methods used over this dataset, including its preprocessing features and the resulted accuracies. As mentioned in Section 2, each method relied on a different preprocessing and training approach to achieve the highest accuracy possible. The classification methods are enumerated in ascending order based on the accuracy score. As for the proposed approach, its related fields are displayed in bold in the table. The results show that the proposed method can outperform the accuracy of the BBNN network [2] specifically by 3.61 *pp*, and outperform the highest accuracy mentioned [5] by 2.81 *pp*.

Table 2. Comparative table for FMA classification methods in terms of accuracy (%)

FMA Classification (fma-small subset)		
Methods	Preprocessing	Accuracy
Representation learning[18]	Mel Spectrogram	57.91
BBNN[2]	Mel Spectrogram	61.11
SongNet[17]	Raw audio	65.23
DenseNet+Data augmentation[13]	STFT Spectrogram	68.20
Our approach	**STFT Spectrogram**	**74.39**

As for the small subset of FMA, Table 2 presents the methods applied over the latter to provide accurate music genre classification. Similar to Table 1, this table shows the different methods used over this dataset, in addition to the preprocessing features used and the resulted accuracies. As for the proposed approach, it outperformed the highest accuracy over the FMA small subset [13] by 6.19 *pp*. Since the proposed approach was inspired by the BBNN network and the latter is not tested against the small subset of FMA, the BBNN Github code [1] was used as-is over this dataset for experimentation purposes. It resulted in an accuracy of 61.11%, found to be less than 13.28 *pp* of the proposed approach.
It is important to note that the outperformance against the related publications is not limited to the proposed network contribution only. The proposed preprocessing process assisted in this outperformance, especially with the GTZAN faults, where we reduced the number of music tracks per genre. Furthermore, the idea of slicing the generated spectrograms to obtain a diversity of visual representations among the same music track.

5 Conclusion and Future Work

In this paper, upgrades to a CNN-based music genre classifier named BBNN are proposed, in addition to a custom preprocessing process for generating STFT spectrograms out of the music tracks. The experiment results showed that the proposed approach was able to outperform existing methods in terms of accuracy. It achieved an accuracy of 97.51% and 74.39% over the GTZAN and the FMA datasets individually while outperforming the best GTZAN and FMA classification methods by 2.81 *pp* and 6.19 *pp* respectively. Also, the proposed approach was found to be better in terms of accuracy while relying on an optimized inception block that uses fewer training parameters to achieve greater results. Our future work should focus on leveraging recent technologies, such as audio and visual transformers, while focusing on reducing the model size and speeding the training process, to create greater music genre classifiers.

Acknowledgments

This work was funded by the “Agence universitaire de la Francophonie” (AUF) and supported by the EIPHI Graduate School (contract ANR-17-EURE-0002). Computations have been performed on the supercomputer facilities of the “Mésocentre de Franche-Comté”.

References

1. Freitag, Michael, Shahin Amiriparian, Sergey Pugachevskiy, N. Cummins and Björn Schuller. “auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks.” Journal of Machine Learning Research, vol. 18, 12 2017.

[1] `https://github.com/CaifengLiu/music-genre-classification`

2. Liu, Caifeng, L. Feng, Guochao Liu, Huibing Wang and Shenglan Liu. “Bottom-up Broadcast Neural Network For Music Genre Classification.” Multimedia Tools and Applications, vol. 80, pp.1-19, 02 2021.
3. Karunakaran, Nagamanoj and A. Arya. “A Scalable Hybrid Classifier for Music Genre Classification using Machine Learning Concepts and Spark.” 2018 International Conference on Intelligent Autonomous Systems (ICoIAS) pp.128-135, 2018.
4. Dai, J., Wenju Liu, Chongjia Ni, Like Dong and H. Yang. “”multilingual” Deep Neural Network for Music Genre Classification.” INTERSPEECH 2015.
5. Mujtaba, G., Lee, S., Kim, J. et al. ”Client-driven animated GIF generation framework using an acoustic feature,” Multimedia Tools and Applications (2021). https://doi.org/10.1007/s11042-020-10236-6
6. Szegedy, Christian, S. Ioffe, V. Vanhoucke and Alexander Amir Alemi. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.” AAAI (2017).
7. Szegedy, Christian, W. Liu, Y. Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, V. Vanhoucke and Andrew Rabinovich. “Going deeper with convolutions.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 1-9.
8. G. Tzanetakis, G. Essl, and P. Cook, “Automatic musical genre classification of audio signals,” 2001.[Online]. Available: `http://ismir2001.ismir.net/pdf/478tzanetakis.pdf`
9. Defferrard, M., Kirell Benzi, P. Vandergheynst and X. Bresson. “FMA: A Dataset for Music Analysis.” in 18th International Society for Music Information Retrieval Conference (ISMIR), 2017. [Online]. Available: `https://arxiv.org/abs/1612.01840`
10. Panagakis, Yannis and Constantine Kotropoulos. “Music genre classification via Topology Preserving Non-Negative Tensor Factorization and sparse representations.” 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (2010): 249-252.
11. Zhang, Weibin, Wenkang Lei, Xiangmin Xu and Xiaofen Xing. “Improved Music Genre Classification with Convolutional Neural Networks.” INTERSPEECH (2016).
12. Choi, Keunwoo, György Fazekas, M. Sandler and Kyunghyun Cho. “Transfer Learning for Music Classification and Regression Tasks.” ArXiv abs/1703.09179 (2017)
13. Bian, Wenhao, J. Wang, Bojin Zhuang, Jiankui Yang, Shaojun Wang and J. Xiao. “Audio-Based Music Classification with DenseNet And Data Augmentation.” PRICAI (2019).
14. Ghosal, Deepanway and M. Kolekar. “Music Genre Recognition Using Deep Neural Networks and Transfer Learning.” INTERSPEECH (2018).
15. Sturm, B. L. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. 11, 1–29.(2013). https://doi.org/10.1080/09298215.2014.894533
16. Valerio, Vinicius D., R. M. Pereira, Yandre M. G. Costa, Diego Bertolini and C. N. Silla. “A Resampling Approach for Imbalanceness on Music Genre Classification using Spectrograms.” FLAIRS Conference (2018).
17. Zhang, Chi and Y. Zhang. “SongNet: Real-time Music Classification.” (2018).
18. Park, Jiyoung, Jongpil Lee, Jangyeon Park, Jung-Woo Ha and Juhan Nam. “Representation Learning of Music Using Artist Labels.” ISMIR (2018).