Music Classification Based on Convolutional Neural Networks

Zijian Zhang
Department of Computer Science
University of XYZ
email@example.com

***Abstract*—This paper evaluates the application of convolutional neural networks to classify short audio clips of music. The main objective is to build a robust model capable of accurately classifying audio tracks into predefined genres using Mel spectrogram features as input. We utilize Convolutional Neural Networks to extract hierarchical patterns from the audio representations. The model is trained and evaluated using a multi-step approach, followed by model training with cross-entropy loss and categorical accuracy as the performance metrics. During the classification process, the accuracy of the data is tested using a recommendation and similarity system from the test data itself to determine whether the system can sufficiently determine the type of music being played.**

***Index Terms*—Music Classification, CNN, Deep Learning, Mel Spectrogram, Audio Processing**

I. Introduction

The significance of music classification at the current stage is multifaceted, influencing both the music industry and various technological advancements. Firstly, music classification plays a critical role in enhancing personalized recommendation systems. By categorizing music based on genres, moods, or themes, platforms can offer users more accurate and tailored recommendations, improving user engagement and satisfaction. Additionally, music classification aids in better organization and retrieval of large music libraries, making it easier for users to discover new tracks based on specific categories. Moreover, in the field of music production, classification helps producers and artists understand trends and characteristics of different musical styles, which can inform creative decisions. From a cultural perspective, music classification also contributes to the analysis of emotional and cultural expressions in music, helping researchers gain deeper insights into how music reflects societal moods and emotions. Furthermore, with the rise of digital content creation, music classification technologies play a pivotal role in copyright management and content regulation by automatically identifying and categorizing music for licensing purposes.

II. Related Work

Deep learning, particularly Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers, has become the dominant approach for music genre classification. CNNs are widely used to extract spatial features from spectrograms, Mel spectrograms, and other time-frequency representations of audio. These models have shown remarkable success in automatic feature extraction, allowing them to learn intricate patterns in audio data without needing handcrafted features. The RNNs and Long Short-Term Memory (LSTM) networks are employed to capture the temporal dependencies and sequences in music, making them particularly useful when dealing with music's rhythmic and melodic structures. Recently, Transformers have also been applied to music classification tasks, providing a robust framework to model long-range dependencies in audio sequences, outperforming traditional models in some tasks.

In the related technologies mentioned above, Convolutional Neural Networks have become one of the most influential architectures in deep learning due to their remarkable success in image and speech recognition tasks. CNNs are designed to automatically and adaptively learn spatial hierarchies of features through the use of convolutional layers, pooling layers, and fully connected layers. Over the years, the application of CNNs has expanded far beyond their initial focus on image processing, leading to breakthroughs in various fields including natural language processing (NLP), audio and speech recognition, healthcare, autonomous driving, and even music classification.

In NLP, CNNs have been used for sentiment analysis, machine translation, and text classification. In audio processing, CNNs have been utilized for speech recognition, music classification, and environmental sound classification, with their ability to learn from spectrograms and time-frequency representations being particularly effective. CNNs are widely used for music genre classification, where the audio is transformed into spectrograms, which serve as the input for the CNN. By learning spatial patterns in the spectrograms, CNNs can distinguish between different genres of music, such as rock, classical, jazz, and pop.

Convolutional neural networks have revolutionized fields ranging from computer vision to music classification and healthcare. Their ability to automatically learn hierarchical features from raw data has enabled significant advancements in numerous applications. As CNN research continues to evolve, the development of more efficient, interpretable, and generalizable models will open up new possibilities for real-world applications, especially in areas with limited labeled data or computational resources.

III. Solution

A. Processing of Datasets

Librosa is a powerful Python library widely used for audio and music analysis. It provides a suite of functions for analyzing, processing, and extracting features from audio data. Specifically designed for tasks involving music and sound analysis, librosa is commonly used in applications such as music classification, speech recognition, and sound event detection. librosa offers built-in functions to extract a wide variety of features from audio, such as Mel-frequency cepstral coefficients (MFCCs), Mel spectrograms, chromagram, spectral contrast, and more. These features are essential for many machine learning and signal processing applications. The library works seamlessly with machine learning libraries like TensorFlow, PyTorch, and Scikit-learn. Features extracted by librosa can be easily fed into machine learning algorithms for classification, clustering, and other tasks.

The Mel scale is based on the human ear's logarithmic perception of frequency. Our hearing sensitivity varies with frequency: we are more sensitive to changes in lower frequencies than higher frequencies. The Mel scale approximates this non-linear relationship, making it a more perceptually relevant feature for sound analysis. This allows the Mel spectrogram to capture the essential perceptual characteristics of sound, making it a better representation of audio signals for human-centered tasks, such as music genre classification, speech recognition, or environmental sound classification. Therefore, we use Librosa to extract features from the original audio, extract log-melspectrogram features from the original audio, and then Feed it into a neural network for training. Figure 1 is Mel spectrum that process the audio diagram. At this point, the original audio is processed into an input suitable for feeding into the model.

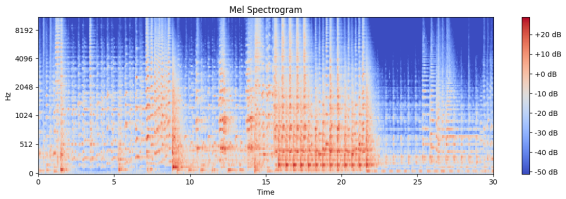

Fig. 1. Mel Spectrogram of an Audio Sample.

Data augmentation is a technique used to artificially increase the size of a training dataset by applying various transformations to the existing data. In the context of music genre classification, data augmentation involves modifying audio files in ways that preserve their label while introducing variability. This process helps create more diverse training examples, which can improve the model's robustness and generalization ability. By applying transformations like pitch shifting, time stretching, adding noise, and volume adjustment, data augmentation artificially increases the size and diversity of the training dataset. Additionally, it enhances the model's ability to handle real-world variations such as background noise, tempo changes, and varying recording conditions. Data augmentation is also effective in addressing class imbalance by generating more samples for underrepresented genres, improving the overall performance of the model. Furthermore, it allows the model to learn more general features, as it is exposed to a wider range of examples, and ensures it can recognize different music characteristics despite variations in audio quality or playback conditions.

B. Model Construction

Creating a Convolutional Neural Network (CNN) involves several key steps, each designed to enable the model to learn hierarchical patterns in data such as images, audio, or video. The process starts with defining the input layer, where the shape of the data is specified (height, width, and number of channels for images). Next, I add convolutional layers that apply filters (kernels) to the input data. These filters learn to detect features like edges, textures, and patterns in the data. The activation function (commonly ReLU) is applied to introduce non-linearity.

Following the convolutional layers, pooling layers are added to reduce the spatial dimensions of the feature maps. MaxPooling is the most common method, selecting the maximum value from small windows to reduce computational load and prevent overfitting. Optionally, batch normalization can be included to stabilize and accelerate the training process by normalizing activations.

To avoid overfitting, dropout layers may be added, which randomly disable a fraction of neurons during training. After these layers, the model's output is typically flattened to a 1D vector, which is then fed into fully connected (dense) layers. These layers learn to make decisions based on the extracted features, and the final layer outputs the predicted class probabilities using a softmax activation for multi-class classification or a sigmoid activation for binary classification.

Once the architecture is defined, the model is compiled with a loss function (categorical cross-entropy), an optimizer (Adam), and metrics. The model is then trained on the data, during which weights are updated via backpropagation. Finally, the model is evaluated on test data to measure its performance, and it can be used to make predictions on new, unseen data. This process enables CNNs to effectively classify complex patterns in diverse data types.

C. Training and Result

In this audio classification project, the goal is to train a deep learning model to categorize audio files into distinct genres based on their Mel spectrogram features. The training and evaluation process follows a structured approach to ensure the model learns effectively and generalizes well to unseen data.

Splitting the dataset into three parts: a training set for model training, a validation set for hyperparameter tuning, and a test set for final evaluation. We typically use an 80-10-10 or 70-15-15 split. This division is critical for assessing the model's generalization capability, as training data is used for learning,

validation data is used for tuning, and the test set serves to evaluate performance on unseen data.

The model training process begins by compiling the model using the Adam optimizer, which adapts the learning rate to optimize performance efficiently. The loss function chosen is categorical cross-entropy, suitable for multi-class classification tasks. Accuracy is used as the primary metric to monitor the model's performance during training. The model is trained over a specified number of epochs, which represents the number of times the entire training dataset is passed through the network. Batch size determines how many samples are processed before the model updates its weights. During training, the model is validated using the validation set to monitor for overfitting, which can occur if the model performs well on training data but poorly on unseen data. The goal is to achieve a balance where the model performs well on both training and validation sets, avoiding overfitting.

After training, the model is evaluated using the test set. The "evaluate()" function provides the test loss and accuracy, offering an initial insight into the model's performance on unseen data. Additionally, the model generates predictions on the test set, which are then converted into class labels by taking the highest probability for each sample. This allows for detailed performance evaluation using metrics such as precision, recall, and the F1-score, all of which are calculated using the classification report from Scikit-learn. These metrics provide a more nuanced understanding of how well the model handles different classes, especially in cases of imbalanced data. Related evaluation metrics can be described as follows:

- Accuracy is the most straightforward evaluation metric and indicates the overall correctness of the model (percentage of correctly classified samples).
- Precision: Measures the accuracy of the positive predictions. High precision means the model makes fewer false positives.
- Recall: Measures the model's ability to correctly identify all the positive samples. High recall means the model makes fewer false negatives.
- F1-score: The harmonic mean of precision and recall, which provides a balance between the two. It is especially useful when the dataset is imbalanced.

TABLE I
EVALUATION METRICS

Genre	Precision	Recall	F1-score
Blues	0.98	0.90	0.90
Classical	0.87	1.00	0.90
Country	0.90	0.87	0.86
Disco	0.92	0.84	0.92
HipHop	0.90	0.94	0.93
Jazz	0.89	0.91	0.88
Metal	0.94	1.00	0.96
Pop	0.82	0.88	0.85
Reggae	0.91	0.84	0.89
Rock	0.89	0.90	0.90

The confusion matrix is another valuable tool for performance analysis. It visually shows the model's ability to correctly classify each class and highlights areas where misclassifications occur. By examining the confusion matrix, one can identify which genres the model confuses with others, providing insights into where the model may require further improvement. From Fig 2, we can see that the overall performance of the model is good, with pop music having the lowest accuracy in classification, followed by hip-hop music.

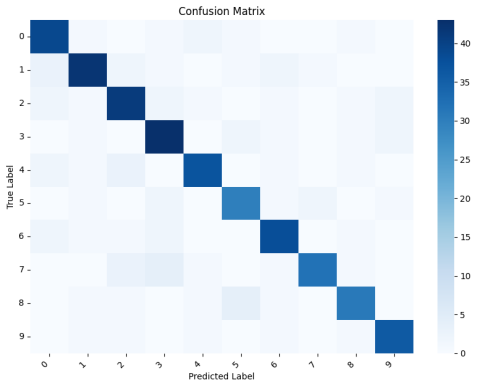

Fig. 2. Confusion Matrix for Genre Classification.

Hyperparameter tuning is an important step in this process. The learning rate, batch size, and number of epochs all influence the model's performance. Experimenting with these parameters, along with regularization techniques like Dropout or L2 regularization, can improve the model's ability to generalize and prevent overfitting. If the model performs poorly, further data augmentation (adding noise or applying time-stretching) can be explored to enhance the dataset's diversity and improve robustness.

IV. DISCUSSION

A small dataset significantly limits the model's ability to learn representative features. The model memorizes the training data but fails to generalize to unseen samples. With insufficient data, the model might not encounter enough variations of each genre, making it unable to learn robust patterns. Additionally, if the dataset lacks diversity—such as representing only a few subgenres or only specific types of audio files—the model may not generalize well to real-world scenarios. To overcome this, expanding the dataset through data augmentation or obtaining more samples from varied sources could improve model generalization.

Achieving high accuracy in music genre classification is often challenging due to the intrinsic complexity of audio features and the similarity between genres. Different genres may share similar sound characteristics, such as rhythm, tempo, or instrumentation, making it difficult for a model to distinguish between them, especially if the genres are musically close. For example, genres like pop and rock may share many overlapping features, which can result in lower accuracy. In such cases, improving the feature extraction process (e.g., using more advanced audio features like MFCCs

or spectrogram variations) or exploring other deep learning models might help improve accuracy, but the complexity of music genres will always remain a limiting factor.

V. Summary

The purpose of this article is to verify the application of convolutional neural networks. The model has good performance and solve the classification problem with around 90 accuracy. The experimental results also prove that deep learning is more suitable for extracting higher-order features of objects feature.

Acknowledgment

The author thanks the academic supervisors and peers who supported the dataset preparation and modeling experiments.

References

[1] B. McFee, et al., “librosa: Audio and Music Signal Analysis in Python,” in Proc. of the 14th Python in Science Conf., 2015.
[2] Y. LeCun et al., ”Deep learning,” Nature, vol. 521, pp. 436–444, 2015.
[3] A. Vaswani et al., ”Attention Is All You Need,” in NeurIPS, 2017.
[4] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” IEEE Trans. on Speech and Audio Processing, vol. 10, no. 5, 2002.