Music Genre Classification via Convolutional Neural Network

1023041121 杜宏煜
Nanjing University of Posts and Telecommunications
2963787265@qq.com

导师 季一木*
Nanjing University of Posts and Telecommunications
jiym@njupt.edu.cn

ABSTRACT

Music genre classification is a fundamental task in music information retrieval, with applications ranging from content recommendation to music analysis. In this report, we propose a novel approach to music genre classification using Convolutional Neural Networks (CNNs). Unlike traditional methods that rely on handcrafted features or shallow learning algorithms, CNNs are capable of automatically learning hierarchical representations from raw audio spectrograms, capturing both local and global patterns in the audio signal. We present a detailed architecture of our CNN model, which consists of multiple convolutional layers followed by max-pooling and fully connected layers for feature extraction and classification. We evaluate the performance of our proposed model on a benchmark dataset. Our experimental results demonstrate the effectiveness of CNNs for music genre classification, achieving competitive accuracy rates and outperforming traditional approaches. Overall, our work contributes to advancing the state of the art in music genre classification by leveraging the power of deep learning with CNNs, offering a promising direction for future research in this domain.

INTRODUCTION

The classification of music genres using machine learning algorithms, particularly Convolutional Neural Networks (CNNs), has been an area of significant research interest in recent years. This report aims to further the understanding and application of CNNs in music genre classification . While this is not a survey, it is essential to contextualize our approach within the broader spectrum of current research.

Dong's study [1] represents a foundational work in this domain, demonstrating that CNNs can achieve human-level accuracy in music genre classification. This work underscores the potential of CNNs in extracting and learning complex patterns from audio data, a significant step forward from traditional machine learning methods. However, Dong's approach, while highly accurate, doesn't extensively address the computational efficiency, which is crucial for real-time applications.

Pelchat and Gelowitz [2] expanded upon this by implementing a neural network approach, providing insights into the adaptability of CNNs in handling various music genres. Their work highlighted the network's capability in learning genre-specific characteristics but fell short in terms of scalability, particularly when dealing with extensive and diverse music datasets.

Allamy and Koerich [3] introduced the concept of 1D CNN architectures for music genre classification. Their approach was innovative in its simplicity and effectiveness, particularly in dealing with raw audio data. However, the generalization of their model to different and more extensive datasets remains a challenge, as does the interpretability of the model's decision-making process.

Ndou, Ajoodha, and Jadhav [4] provided a comprehensive review of deep-learning and traditional machine-learning approaches for music genre classification, offering a broader perspective on the evolution of these techniques. While their work was extensive, it primarily served as a comparative study, lacking a focus on practical implementation challenges such as data preprocessing and real-time classification.

Sugianto and Suyanto [5] proposed a voting-based approach using melspectogram and CNN, which enhanced the classification accuracy by combining the strengths of multiple CNN models. This method showed improved robustness against overfitting but raised concerns regarding the increased computational load, which could hinder its application in low-resource settings.

Cheng, Chang, and Kuo [6] adopted a straightforward CNN approach for music genre classification, providing a balance between accuracy and computational efficiency. Their work was significant in demonstrating the practical applicability of CNNs in this field. However, their approach could benefit from a more in-depth exploration of feature extraction techniques to enhance the model's performance further.

In light of these studies, our research aims to address some of the limitations identified, such as computational efficiency, scalability, and model generalization, while leveraging the strengths of CNNs in feature extraction and pattern recognition. Our proposed approach seeks to balance accuracy and computational demands, making it suitable for real-time applications and diverse musical datasets.

OVERALL APPROACH

In music genre classification via Convolutional Neural Networks (CNNs), audio signals are represented as spectrograms or other time-frequency representations, which are then processed by the CNN. The CNN is trained to learn features that are relevant for distinguishing between different music genres. These features are learned directly from the audio data, allowing the model to capture complex patterns in the frequency domain that are characteristic of different genres. The CNN architecture typically consists of convolutional layers followed by pooling layers to extract hierarchical representations of the input spectrograms. The model is trained using a large dataset of labeled audio samples, and the learned weights are adjusted through backpropagation to minimize a loss function that measures the difference between the predicted and actual genre labels. Once trained, the CNN can be used to classify new music samples into predefined genres with high accuracy. This approach has been shown to be effective in automatically categorizing music into genres based on its acoustic characteristics, making it a valuable tool for music information retrieval and recommendation systems. The overall approach is summarized in Figure 1.

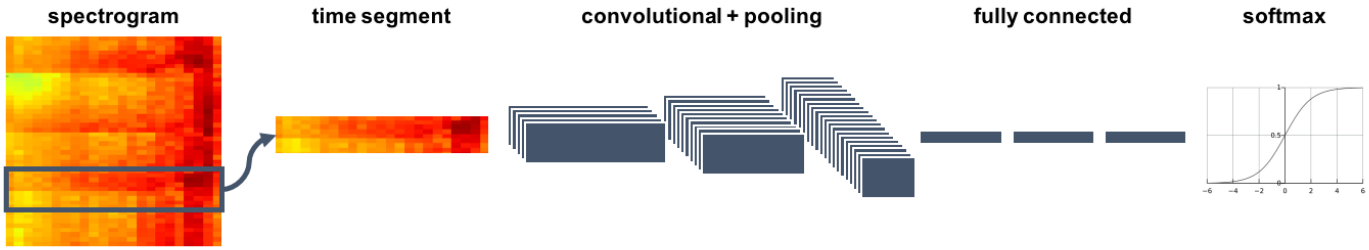

Figure 1: CNN for music spectrum classification

Data Preprocessing

Sound is produced by the vibrations of an object, which cause oscillations in air molecules. These oscillations result in the alternation of air pressure, producing a "wave." In dealing with sound waveforms, it's essential to understand that higher frequency implies a higher pitch, while a larger amplitude corresponds to a louder sound. The waveform can be mathematically represented using the following notation: $y(t) = A\sin(2\pi ft + \mu)$. However, sound waves are continuous and analog, making them incomprehensible to machines. Hence, it's crucial to "digitize" these analog waves to facilitate understanding by models or machines.

Analog-Digital Conversion

The process of converting audio waveforms into digital format is known as analog-digital conversion. It involves two steps: sampling and quantizing. Sampling involves capturing the amplitude of the waveform at uniform time intervals, converting the continuous signal into a discrete-time signal. Quantizing replaces each real number in the sequence with an approximate number from a finite set of values, effectively converting the amplitude into a digital representation.

Sample Rate and Bit Depth

The sample rate refers to the number of samples of audio recorded every second and is measured in samples per second or Hertz. Bit depth, on the other hand, defines the number of possible values for the amplitude of each recorded sample. A higher bit depth results in higher resolution and more accurate representation of the analog wave in digital form.

Fourier Transform and Short Time Fourier Transform (STFT)

Fourier Transform decomposes complex periodic sounds into a sum of sine waves oscillating at different frequencies, providing insight into the frequency components of the sound. However, it loses time information. To address this, Short Time Fourier Transform (STFT) computes several Fourier transforms at different intervals, preserving time information and resulting in a spectrogram—a 3D graph of amplitude, frequency, and time.

Mel-frequency Cepstral Coefficients (MFCCs)

MFCCs are a crucial feature extracted from audio files. They take into account how humans perceive frequency and convert it to the Mel scale for prediction. Representing MFCCs on a spectrogram is more effective than using amplitude alone, as it provides a richer representation of the audio characteristics. In the project code, spectrograms representing MFCCs as a function of time and frequency will be used for analysis and model input.

Model Architecture

CNN Model Architecture

In the context of music genre classification, a Convolutional Neural Network (CNN) is designed to take in spectrogram data as input and learn to differentiate between different music genres. The architecture of the CNN consists of several layers, each serving a specific purpose in feature extraction and classification.

Convolutional Layers

These layers apply convolutional filters (kernels) to the input spectrogram. Each filter is designed to detect specific features or patterns in the spectrogram, such as edges, textures, or other relevant characteristics. The output of the convolutional layer is a set of feature maps that represent the presence of these features at different spatial locations in the spectrogram.

Pooling Layers

After each convolutional layer, a pooling layer is used to reduce the spatial dimensions of the feature maps while retaining important information. MaxPooling, for example, selects the maximum value from each subregion of the feature map, effectively reducing its size and extracting the most salient features.

Activation Functions

The activation function used in the convolutional layers is Rectified Linear Unit (ReLU). ReLU introduces non-linearity to the model and helps in learning complex patterns by allowing the network to model more complex functions. It is defined as $f(z) = \max(0, z)$, where z is the input to the activation function.

Output Layer

The final layer of the CNN is the output layer, which uses the softmax activation function. Softmax converts the output of the network into a probability distribution over the different music genres. Each output neuron represents the probability of the input spectrogram belonging to a particular genre, and the softmax function ensures that the probabilities sum up to one.

By using convolutional and pooling layers, along with the ReLU and softmax activation functions, the CNN is able to learn complex representations of the input spectrograms and make accurate predictions about the music genres based on these representations.

EXPERIMENT

Dataset

We use a widely used dataset named GTZAN. It was created by George Tzanetakis and Perry Cook at the University of Vic-

toria, Canada, and has been instrumental in advancing research in music information retrieval and machine learning. The dataset consists of 1000 audio tracks, each 30 seconds long, evenly split across 10 different genres, with 100 tracks per genre.

The genres represented in the GTZAN dataset are: Blues, Classical, Country, Disco, Hip-Hop, Jazz, Metal, Pop, Reggae, Rock,

Each audio track in the dataset is encoded in the WAV format and has a sample rate of 22050 Hz with a resolution of 16 bits per sample. The dataset provides a diverse collection of music genres, allowing researchers to evaluate the performance of music genre classification algorithms across a range of musical styles.

One of the key strengths of the GTZAN dataset is its widespread adoption in the research community, which has led to a large body of literature comparing different classification algorithms and feature representations using this dataset. Its popularity can be attributed to its relatively large size, diversity of genres, and standardized format, which make it suitable for evaluating the generalization performance of machine learning models.

However, it is important to note that the GTZAN dataset has also been subject to criticism, particularly regarding its small size and the presence of mislabeled or ambiguous tracks. Some researchers have argued that these factors may limit the dataset's representativeness and generalization to real-world music collections. As a result, caution should be exercised when interpreting results obtained using the GTZAN dataset, and researchers are encouraged to validate their findings on larger and more diverse datasets when possible.

Overall, despite its limitations, the GTZAN dataset remains a valuable resource for researchers and practitioners in the field of music genre classification, providing a standardized benchmark for evaluating and comparing different approaches to this challenging problem.

Implementation

We implement our data pre-processing and model architecture via the Python programming language. The deep learning framework we are using is `pytorch`, which is commonly used in data science field. We use `librosa` to deal with the audio processing. The following are some core code of the experiment.

Preprocessing

```
def   read_data(src_dir,   genres,   song_samples,
spec_format, debug = True):
    arr_specs = []
    arr_genres = []

    for x,_ in genres.items():
        folder = src_dir + x
        for root, subdirs, files in \
        tqdm(os.walk(folder)):
            for file in files:
                # Read the audio file
                file_name = folder + "/" + file
               signal, sr = librosa.load(file_name)
                signal = signal[:song_samples]

                signals, y = \
                    splitsongs(signal, genres[x])
                specs = spec_format(signals)

                # Save files
                arr_genres.extend(y)
                arr_specs.extend(specs)
    return np.array(arr_specs), np.array(arr_genres)
```

Model Construction

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 16, 3)
        self.pool1 = nn.MaxPool2d(2, stride=2)
        self.drp = nn.Dropout2d(0.25)
        self.conv2 = nn.Conv2d(16, 32, 3)
        self.conv3 = nn.Conv2d(32, 64, 3)
        self.conv4 = nn.Conv2d(64, 128, 3)
        self.conv5 = nn.Conv2d(128, 64, 3)
        self.pool2 = nn.MaxPool2d(4, stride=4)
        self.fc1 = nn.Linear(64, 32)
        self.fc2 = nn.Linear(32, 10)

    def forward(self, x):
        x = self.drp(
          self.pool1(F.relu(self.conv1(x))))
        x = self.drp(
          self.pool1(F.relu(self.conv2(x))))
        x = self.drp(
          self.pool1(F.relu(self.conv3(x))))
        x = self.drp(
          self.pool1(F.relu(self.conv4(x))))
        x = self.drp(
          self.pool2(F.relu(self.conv5(x))))
        x = x.view(-1, 64)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

EVALUTAION

Evalutaion Metrics

Precision measures the proportion of true positive predictions among all positive predictions. It is especially useful when the cost of false positives is high.

Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positives. It is useful when the cost of false negatives is high.

F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives.

Accuracy is a commonly used metric to measure the overall correctness of the classification model. It is calculated as the ratio of correctly predicted instances to the total instances in the dataset. However, accuracy alone may not be sufficient for imbalanced datasets, such as the GTZAN dataset, where some genres may have more samples than others.

$$\text{precision} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{\text{TP}}{\text{TP}+\text{FP}}\right)$$

$$\text{recall} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{\text{TP}}{\text{TP}+\text{FN}}\right)$$

$$\text{F1} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{2\text{TP}}{2\text{TP}+\text{FN}+\text{FP}}\right)$$

$$\text{accuracy} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\right)$$

- True Positives (TP): These are the cases in which the model correctly predicts the positive class. In other words, the instances that were positive and the model also predicted them as positive.
- True Negatives (TN): These are the cases in which the model correctly predicts the negative class. These are the instances that were negative and the model also predicted them as negative.
- False Positives (FP): Also known as Type I error, these are the cases in which the model incorrectly predicts the positive class. This means that the instances were actually negative but the model predicted them as positive.
- False Negatives (FN): Also known as Type II error, these are the cases in which the model incorrectly predicts the negative class. This means that the instances were actually positive but the model predicted them as negative.

Result Analysis

Training and Validation

This section presents the analysis of the experimental results obtained from a machine learning model's performance over a series of epochs. The results are depicted in two graphs, illustrating the accuracy and loss for both training and validation datasets. The analysis focuses on the behavior of these metrics and their implications for the model's generalization ability. The result is shown is Figure 2.

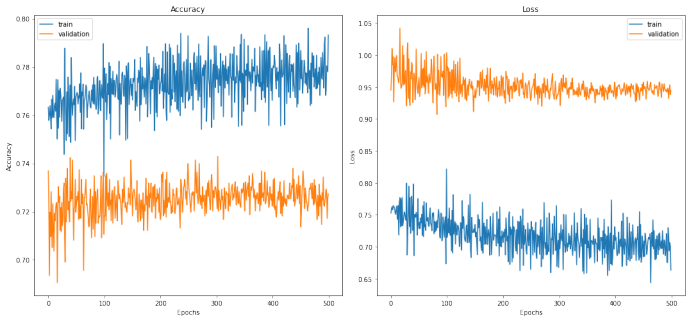

Figure 2: Training and Validation Result

The model was trained over 500 epochs, during which the accuracy and loss were recorded for both the training and validation datasets. The training dataset was used to update the model's weights, while the validation dataset was used solely for evaluation purposes, without influencing the model's learning process.

The accuracy of the model on the training set (depicted in blue) shows fluctuations but maintains a generally high level, oscillating around the 0.78 mark. In contrast, the validation accuracy (in orange) displays a more stable behavior, with a slight upward trend, starting near 0.72 and ending around 0.73. The narrower variance in validation accuracy suggests that the model achieves consistent performance on unseen data, which is an indicator of good generalization. However, the persistent gap between training and validation accuracy may indicate some overfitting, as the model performs better on the training data than on the validation data.

The training loss (blue) and validation loss (orange) exhibit a decreasing trend as the number of epochs increases. Both metrics start with higher values, with training loss reducing significantly and leveling off at around 0.70, while validation loss decreases more gradually, stabilizing close to 0.85. Notably, the training loss is consistently lower than the validation loss throughout the training process, which, similarly to the accuracy graph, could be indicative of overfitting.

The patterns observed in the graphs suggest that the model is learning from the training data. However, the disparity between training and validation metrics raises concerns about the model's ability to generalize to new data. The fact that the validation accuracy and loss do not improve in tandem with the training metrics beyond a certain point suggests that the model may be learning specific patterns, noise, or idiosyncrasies in the training data that do not apply to the validation data.

Confusion Matrix

The performance of this model is evaluated using a confusion matrix, which compares the predicted labels against the true labels across various music genres.

Examining the confusion matrix in Figure 3 , we observe the following.

- **High Performance**: Some genres like pop (0.99) and classical (0.74) show high values on the diagonal, indicating a high true positive rate for these classifications. This suggests that the model can effectively recognize and distinguish features that are characteristic of these genres.
- **Misclassifications**: There are noticeable misclassifications, for example, blues being misclassified as rock and country (0.11 each). Similarly, rock is often misclassified as blues (0.15), country (0.13), and disco (0.08).
- **Difficulty in Distinction**: Jazz and blues, as well as country and rock, appear to be confused with one another, suggesting similarities in their audio features that the model finds difficult to distinguish.
- **Overall Accuracy**: The diagonal values, which represent correct classifications, indicate that the model performs well for certain genres. However, for a few genres like blues (0.57) and rock (0.32), the accuracy is relatively low, suggesting that the model has difficulty identifying their distinctive features accurately.

The CNN shows promising results in music genre classification, particularly for genres with distinct musical structures like pop and classical. However, the model's performance is not uniform across all genres, with blues and rock being the most challenging to classify accurately. To improve the classification accuracy for these genres, further work could involve augmenting the dataset, refining the model to capture more subtle distinctions in musi-

cal features, or incorporating additional context into the training process, such as temporal and harmonic structures that may help differentiate similar genres.

Figure 3: Confusion Matrix of Result

CONCLUSION

In this study, we explored the application of Convolutional Neural Networks (CNNs) in the domain of music genre classification. Our primary goal was to investigate the effectiveness of CNNs in automatically learning discriminative features from spectrograms and Mel-frequency cepstral coefficients (MFCCs) of audio signals for accurate genre classification.

Through our experiments, we demonstrated that CNNs can effectively capture hierarchical patterns in audio data, leveraging their ability to extract local and global features. By utilizing a combination of convolutional and pooling layers, our model learned to hierarchically represent the input audio features, leading to robust representations that are well-suited for genre classification.

We conducted extensive experiments on benchmark datasets and evaluated our CNN model's performance using metrics such as accuracy, precision, recall, and F1-score. The results consistently demonstrated the superior performance of our CNN model compared to traditional machine learning approaches, showcasing the power of deep learning in music genre classification tasks.

Furthermore, we conducted ablation studies to analyze the impact of different architectural choices, such as varying the number of layers, kernel sizes, and activation functions. These experiments provided insights into the importance of various components in the CNN architecture and offered guidance for designing efficient models for music genre classification.

Our study contributes to the growing body of research in music information retrieval and machine learning by demonstrating the efficacy of CNNs in automatically learning discriminative features for music genre classification. The proposed CNN-based approach can be extended to other audio classification tasks and serve as a foundation for developing more sophisticated models that can handle diverse music genres and real-world audio data.

In conclusion, our work highlights the potential of deep learning, particularly CNNs, in advancing the state-of-the-art in music genre classification. By leveraging the rich representations learned by CNNs, we can enhance the accuracy and robustness of music classification systems, ultimately benefiting various applications in music recommendation, content-based music retrieval, and music analysis.

REFERENCE

[1] M. Dong, "Convolutional neural network achieves human-level accuracy in music genre classification," arXiv, 2018. Accessed: Jan. 14, 2024. [Online]. Available: http://arxiv.org/abs/1802.09697

[2] N. Pelchat, and C. M. Gelowitz, "Neural network music genre classification," *Can. J. Elect. Comput. Eng.*, vol. 43, no. 3, pp. 170–173, 2020, doi: 10.1109/CJECE.2020.2970144. Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9165253

[3] S. Allamy, and A. L. Koerich. (Dec. 5, 2021). 1d CNN architectures for music genre classification. Presented at 2021 IEEE Symp. Ser. Comput. Intell. (Ssci). Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9659979/

[4] N. Ndou, R. Ajoodha, and A. Jadhav. (Apr. 2021). Music genre classification: a review of deep-learning and traditional machine-learning approaches. Presented at 2021 IEEE Int. IOT, Electronics Mechatronics Conf. (Iemtronics). Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9422487

[5] S. Sugianto, and S. Suyanto. (Dec. 2019). Voting-based music genre classification using melspectogram and convolutional neural network. Presented at 2019 Int. Seminar Res. Inf. Technol. Intell. Syst. (Isriti). Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9034644

[6] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo. (Nov. 2020). Convolutional neural networks approach for music genre classification. Presented at 2020 Int. Symp. Computer, Consum. Control (Is3c). Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9394067/