DT2470 HT24 Music Informatics

Final Project

Wenrui Zhao
Information and Network Engineering
wenruiz@kth.se

jingxi Huang
Information and Network Engineering
jingxi@kth.se

Letao Feng
Information and Network Engineering
letao@kth.se

Jingwen Liu
Information and Network Engineering
jingwenl@kth.se

October 23, 2024

1 Introduction

1.1 Background

Music is a universal form of expression that transcends cultural and linguistic boundaries. Throughout history, music has played a crucial role in human emotions, serving as a means for individuals to convey feelings, tell stories, and connect with others. The emotional response elicited by music is profound; studies have shown that different musical elements can evoke a wide range of emotions, such as joy, sadness, nostalgia, and excitement.

In recent years, with the rise of digital music consumption and the increasing reliance on music streaming services, the ability to understand and classify the emotional content of music has become increasingly important. Music Emotion Recognition (MER) is an emerging field that aims to automate the process of identifying the emotional characteristics of musical pieces[1]. This capability has significant implications in various applications, including personalized music recommendation systems, soundtrack design for films and games, and therapeutic uses in mental health.

1.2 Objectives

The primary objective of this project is to develop a Music Emotion Recognition system that can classify music into predefined emotional categories based on the analysis of audio signals. Specifically, the system aims to categorize music into five emotional categories: Romantic, Happy, Sad, Devotional, and Party, using a robust dataset of Hindi film songs.

To achieve this objective, the project will focus on several key tasks:

- **Data Acquisition:** Utilizing the MER500 dataset, which consists of audio clips specifically labeled for emotional content.
- **Feature Extraction:** Implementing advanced techniques to extract meaningful audio features that can effectively represent the emotional characteristics of the music.
- **Model Development:** Employing a range of machine learning and deep learning algorithms to classify the audio clips based on their emotional content.
- **Performance Evaluation:** Assessing the accuracy and reliability of the developed model using various metrics and testing it on both the training dataset and external datasets.

By leveraging modern techniques in signal processing and machine learning, this project not only seeks to contribute to the field of music informatics but also aims to enhance the interactive experience of users by providing them with music that resonates with their emotional state.

2 Method

2.1 Dataset & Environment

For our project we choose MER500 as our dataset. This dataset consists of songs in 5 popular emotional categories for Hindi film songs as Romantic, Happy, Sad, Devotional and Party. About 500 audio files of

about 10 seconds song clip from the original song are available for researchers in music emotion recognition for experimentation.

We set up the environment in Google Colab for our MER project, and the gpu we use is Tesla T4.

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P8              9W /  70W |      3MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Figure 1: System environment configuration

2.2 Feature extraction

Feature extraction is a critical step in the process of music emotion recognition, as it transforms raw audio signals into meaningful representations that can be utilized for classification tasks[2]. In this project, we employ various audio features that capture different aspects of the music, enabling the model to differentiate between emotional categories effectively.

We utilize five key audio features extracted from each audio clip:

- **Mel-frequency Cepstral Coefficients (MFCC):** MFCCs are widely used in speech and audio processing. They represent the short-term power spectrum of sound and are derived from the Fourier Transform of the audio signal. In our implementation, we extract 40 MFCCs and compute their mean over the duration of the audio clip to obtain a fixed-length feature vector.
- **Chroma Feature:** The chroma feature provides a representation of the energy distribution across the 12 different pitch classes. It captures harmonic properties of the audio and is particularly useful for identifying chords and musical keys.
- **Mel Spectrogram:** The mel spectrogram is a representation of the audio signal in the mel frequency scale, which aligns more closely with human auditory perception. We compute the mel spectrogram and average its values to obtain a fixed-length feature vector.
- **Spectral Contrast:** Spectral contrast measures the difference in amplitude between peaks and valleys in the sound spectrum. This feature is important for distinguishing between different timbres and textures in music.
- **Tonnetz:** The tonnetz (or tonal centroid features) captures the harmonic relations within music. It provides insight into the emotional content of the audio, as different tonnetz features correlate with different emotions.

The extracted features are concatenated into a single feature vector, which will be used for further analysis and model training. By utilizing a combination of time-domain and frequency-domain features, we can capture the essential characteristics of music that correlate with emotional content. This lays a strong foundation for the subsequent modeling and classification tasks.

2.3 Models and Algorithms

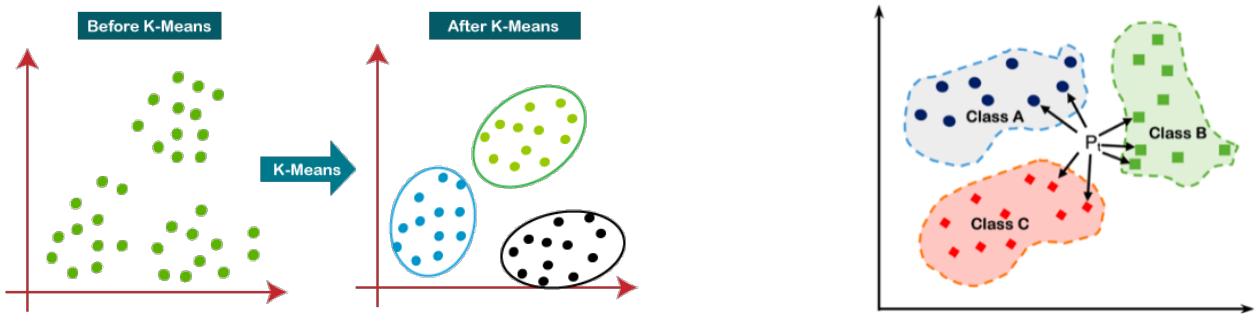
2.3.1 Kmeans

K-Means is an unsupervised learning algorithm used for clustering (Figure 2a). In our project on music emotion recognition, K-Means is used to group songs into clusters corresponding to different emotional categories, based on the extracted audio features we mentioned before.

Specifically, we used the K-Means algorithm to cluster audio samples into five emotional categories: Sad, Devotional, Happy, Party, and Romantic. The features were extracted from the WAV files using the librosa library, including MFCCs, chroma, spectral contrast, and others. We applied K-Means with 5 clusters to group the audio features, and then used a label-mapping function to align the clustering results with the true labels.

This is necessary, because unlike KNN, K-Means does not rely on labeled data during training, but instead attempts to discover inherent structures in the data. However, if we later want to calculate the accuracy, we need to know the true labels for comparison.

To evaluate the clustering performance, we calculated the accuracy and confusion matrix by comparing the mapped labels with the true labels in both the training and test datasets.



(a) K-means (b) KNN

Figure 2: Principles of K-means and KNN

2.3.2 KNN

K-Nearest Neighbors (KNN) is a simple and effective supervised learning algorithm used for classification tasks (Figure 2b). Unlike K-Means, KNN relies on labeled training data to make predictions. This method uses the training data to measure the distances between data points, and then predicts the new sample's label by identifying the most common label among the closest neighbors.

The features we extracted before served as the input for the KNN classifier. The KNN algorithm classifies each sample based on the majority label of its k nearest neighbors in the feature space. After many experiments, we finally decide to set k equals to 3, as this value provided good results.

After training the KNN classifier, we evaluated its performance by calculating the accuracy on both the training and test datasets.

2.3.3 SVM

Support Vector Machine (SVM) is a supervised learning algorithm widely used for classification tasks. SVM works by finding the optimal hyperplane that maximizes the margin between different classes in a given feature space. This hyperplane is chosen such that it separates the data points of different classes with the largest possible margin, minimizing classification errors. In a two-class problem, the optimal hyperplane is the one that maximizes the distance from the nearest support vectors of each class. For non-linearly separable data, SVM can transform the original feature space into a higher-dimensional space where a linear hyperplane can separate the data.

In this experiment, we used SVM to classify the extracted features from the dataset. We employed the SVC model. We used model.fit to train the model automatically. Then we used model.predict to get the predicted labels. Finally, we compared the predicted labels with true labels to obtain the accuracy of this model. We have tried different kernels such as linear kernel, poly kernel, and RBF kernel. It turned out that the linear kernel performs best.

2.3.4 Random Forest

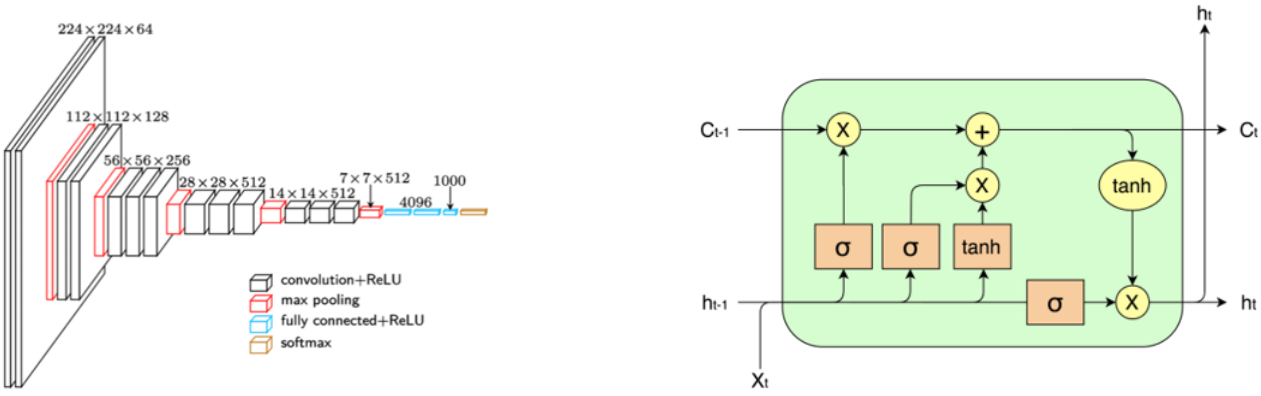
Random Forest is an ensemble learning method primarily used for classification and regression tasks. It builds multiple decision trees during training and outputs the class. Each tree is trained on a random subset of the data and features, introducing diversity and reducing overfitting. This helps improve the model's accuracy and robustness. The randomness of tree construction makes Random Forest less sensitive to data and more generalizable. The Random Forest model is suitable for audio classification tasks where features can be highly correlated.

In our music emotion recognition task, we used Random Forest to classify five different emotional categories based on extracted audio features including MFCCs, chroma, spectrum, and so on. After feature extraction from the WAV files, we trained a Random Forest classifier with 100 decision trees. The classifier was trained on the training set and evaluated on the test set. We use accuracy as the key performance metric.

2.3.5 VGG16

VGG16 is a supervised convolutional neural network (CNN) model with simple yet effective architecture. The structure of VGG16 is shown in Figure 3(a). It includes 13 convolutional layers, 5 max-pooling layers, 3 fully connected layers and a final Softmax classification layer. The convolutional kernel size is fixed at 3*3, which not only reduces the number of parameters but also increases the network's ability to express non-linear relationships. This architecture is designed to progressively capture more complex features from images, making it effective for classification tasks.

In our project, we extract the mel spectrogram as input. Compare to other features, mel spectrograms provide a richer and more detailed two-dimensional representation of audio signals, which allows the model to learn complex features directly from the raw spectrogram. We use the Adam optimizer, set the batch size to 32 and train 50 epochs on the training dataset. Finally, we evaluate its performance on the testing dataset.



(a) VGG16 (b) LSTM

Figure 3: Structures of VGG16 and LSTM

2.3.6 VGG16+LSTM

In order to improve the performance, we add the Long Short-Term Memory(LSTM) network on VGG16. The structure of LSTM is shown in Figure 3(a). LSTM is a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data, making it particularly effective for tasks involving time-series or sequential inputs. Its unique architecture includes memory cells and gating mechanisms that enable it to selectively retain or forget information over long periods. This makes LSTM well-suited for music emotion recognition, as musical pieces are inherently temporal and contain complex patterns that evolve over time.

We add a LSTM layer with 128 units to the previous VGG16 network while other parts remain the same. We train the model on the training dataset and evaluate its performance on the testing dataset.

3 Results

The results of our algorithms are shown in the table.

Algorithms	Accuracy
K-Means	0.3980
KNN	0.3265
SVM	0.4490
Random Forest	0.4709
VGG16	0.5816
VGG16+LSTM	0.6327

Table 1: Results of different algorithms

4 Discussion and Conclusion

From tabel 1 we can see VGG16/VGG16+LSTM perform much better than the other four algorithms. The reason is that VGG16 can automatically extract complex multi-level features from the spectrogram of music,

capture both local and global patterns and LSTM shows exceptional sequential processing capabilities, ability to capture temporal dependencies, and robustness. While on the other hand, traditional methods rely on handcrafted features and struggle with the high dimensionality and complexity of audio.

However, the highest test accuracy is only 0.6327. One possible reason is that the dataset is relatively small, resulting in the model not being able to learn enough features, or maybe because that the music styles under different labels are similar, e.g. happy and party, making the model not able to distinguish between the two emotions very well.

5 Proposed Improvements

To improve the performance of the Music Emotion Recognition system, several key strategies can be implemented. Firstly, increasing the dataset size by gathering additional, diverse musical styles and emotional categories can enhance the model's ability to learn distinct features associated with different emotions. Additionally, exploring advanced feature extraction methods beyond the traditional MFCCs and chroma features—such as spectral centroid and zero-crossing rate may provide deeper insights into the emotional content of the music. Furthermore, fine-tuning pre-trained models using transfer learning on large music datasets can enable the model to leverage previously learned features, reducing training time while potentially increasing accuracy. Finally, investigating different model architectures, such as GRU or CNN-RNN hybrids, may improve the system's ability to capture both temporal and spectral information effectively, leading to better classification outcomes.

References

[1] Y. E. Kim, E. M. Schmidt, R. Migneco, B. C. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 255–266.

[2] M.Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.* Springer, 2015.