AN ABSTRACT OF THE
RESEARCH-PROJECT-IN-LIEU-OF-THESIS OF

Shengxuan Wang for the degree of Master of Science in Computer Science presented on March 14th, 2025.

Title: Comparative Analysis of Sentiment in Chinese Opera with Different Features

Abstract

Chinese Opera, recognized as an Intangible Cultural Heritage by UNESCO, represents the expression of traditional Chinese culture through its unique art form that combines singing, acting, and recitative. Understanding the emotional expression of Chinese Opera can help scholars and enthusiasts study it more deeply. There are many approaches to classify the emotional state of general music using deep learning with different models. This study will analyze and compare the performance of these methods on Chinese Opera using different features. We first conducted extensive data corrections and cleansing to address several data challenges in existing datasets of Chinese Opera, which we release as the Opera Singing Dataset-2025. We then compared the performance of spectrograms, raw audio, lyrics, and latent representation from pretrained models as features. The results demonstrate that models trained on lyrics outperformed all of the different acoustic features. Through this analysis, we conclude that the complexity of Chinese Opera performances added challenges not present with other types of music; challenges that are further aggravated by the lack of high quality annotated data. Many traditional approaches to automatic music sentiment analysis may not scale well to Chinese Opera, and as such, new and specific methods may need to be developed alongside the collection of additional labeled datasets.

©Copyright by Shengxuan Wang
March 14th, 2025
All Rights Reserved

Comparative Analysis of Sentiment in Chinese Opera with Different Features

by

Shengxuan Wang

A RESEARCH-PROJECT-IN-LIEU-OF-THESIS

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 14th, 2025
Commencement June 2025

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Patrick Donnelly, for his guidance and encouragement throughout my learning, research, and writing. This project would not have been possible without his expertise. I am also grateful to all the members of the Soundbendor Lab. Our group discussions enriched my understanding, and I appreciate their willingness to share ideas and provide feedback to help refine my work.

Special thanks to all the professors and instructors I encountered during my master's program. The knowledge I gained from them laid the foundation for my research.

I would also like to acknowledge `https://www.xikao.com` and `https://www.jingju.com` for maintaining and providing detailed metadata on Chinese Opera, which has been invaluable to my work.

Finally, I am deeply grateful to my parents and my girlfriend for their unwavering support and encouragement, which has motivated me throughout this journey.

TABLE OF CONTENTS

Page

1 Introduction 1

1.1 Motivation . . . 1

1.2 Solutions and Experiments . . . 1

1.3 Discovery . . . 2

2 Background 3

2.1 Chinese Opera . . . 3

2.2 Deep Learning Models . . . 5

2.2.1 Convolutional Neural Network . . . 5

2.2.2 Long Short-Term Memory Network . . . 5

2.2.3 Convolutional Recurrent Neural Network . . . 7

2.3 Evaluation Metrics . . . 7

2.3.1 Accuracy . . . 8

2.3.2 Precision . . . 8

2.3.3 Recall . . . 9

2.3.4 F_1-score . . . 9

3 Related Work 10

3.1 Audio Features . . . 10

3.1.1 Raw Audio . . . 10

3.1.2 Spectrogram Features . . . 10

3.2 Sentiment Analysis and Classification . . . 11

3.2.1 Traditional Machine Learning Methods . . . 12

3.2.2 Deep Learning Methods . . . 12

3.3 Pretrained Models . . . 13

3.3.1 Pretrained Music Model: MERT . . . 14

3.3.2 Pretrained Language Model: BERT . . . 14

4 Dataset 15

4.1 Singing Voice Audio Dataset . . . 15

4.2 Data Problems and Solutions . . . 17

4.2.1 Formatting inconsistency . . . 17

4.2.2 Entry Mismatching . . . 18

4.2.3 Audio Trimming . . . 18

TABLE OF CONTENTS (Continued)

Page

4.2.4 Preparing for Machine Learning 18
4.2.5 Insufficient Data 20
4.3 Jingju A Cappella Singing Dataset 21
4.4 Adding lyrics 21
4.5 Opera Singing Dataset-2025 21
4.5.1 Data Profile 22
4.5.2 Data Features 23
4.5.3 Target Class 26
4.6 Summary 26

5 Experimental Design 27
5.1 Audio Segmenting 27
5.2 A Specialized Folds Splitting Strategy 28
5.2.1 Information Leakage 28
5.2.2 Fold Imbalance 28
5.2.3 Solution 29
5.3 3-level evaluation method with Voting 31
5.4 Folds aggregation 32

6 Experiments 33
6.1 Spectrograms Features 33
6.1.1 CNN 33
6.1.2 CRNN 34
6.2 Raw Audio Features 34
6.2.1 CNN 35
6.2.2 LSTM 35
6.2.3 Use MERT Embeddings 36
6.3 Lyric Features and Multimodal Learning 37

7 Discussion 39
7.1 Cross-feature Comparison 39
7.2 Data Scarcity 40

TABLE OF CONTENTS (Continued)

Page

8 Conclusion 42

8.1 Future Work 42

8.2 Final Thoughts 43

Bibliography 44

LIST OF FIGURES

Figure Page

2.1 Illustration depicting costume and makeup in Chinese Opera (from [34]). 4

2.2 The standard convolutional filters (left) are separated into depth-wise convolution filters (top right) and point-wise convolution filters (bottom right). Figure reproduced from [17]. 6

2.3 A single LSTM cell, the "x" and "+" symbols are the element-wise multiplication and addition operator. "σ" represents the sigmoid activation function. Figure reproduced from [29]. 7

2.4 A diagram illustrating precision and recall. 9

4.1 Depiction of the original dataset directory tree (left) compared to new dataset directory tree (right). 19

4.2 Depiction showing the bit rate distribution (left), number of channels distribution (middle), and sample rates distribution (right) of all Chinese Opera recordings. 22

4.3 Visualization of the distributions of the features in the dataset. 25

5.1 The number of recordings that represent each song. 29

5.2 The total durations of each song in the dataset. 29

5.3 Flow chart depicting the fold splitting strategy. 30

5.4 Example illustrating the three levels of evaluation. 32

6.1 Visualization of the Audio-Lyrics fusion model. 38

LIST OF TABLES

Table		Page
2.1	Frequency of different roles and voice forms, considering four singers [36].	4
4.1	Example excerpt of the dataset spreadsheet. Some Mandarin text is removed or translated.	16
4.2	Duration statistics of all Chinese Opera recordings	22
6.1	Comparison of results between CNN and CNN-LSTM.	34
6.2	Comparison of results between Vanilla CNN and MobileNets.	35
6.3	Structure designs of LSTM Models.	36
6.4	Comparison of results between LSTM and BiLSTM.	36
6.5	Comparison of results between MERT using CNN or FC as the classification layer.	37
6.6	Comparison of results between lyrics features using BERT alone and a multi-model approach.	38
7.1	Comparison among the best result from each feature set and model.	40
7.2	Performance of LSTM before and after adding more data.	40

Chapter 1: Introduction

Music Emotion Recognition (MER) is an important research area in the field of Music Information Retrieval (MIR). This technology can be widely used in the applications of music recommendation, emotion computing and human-computer interaction. In recently years, the rapid development of deep learning technology has provided new methods which can be applied to MER tasks, such as the application of Convolutional Neural Network (CNNs) and Recurrent Neural Network (RNNs) [18].

1.1 Motivation

Despite existing MER research mainly focuses on pop music or classical music, there is a lack of MER research on Chinese Opera. Chinese Opera is a complex on-stage art form that combines singing, recitative, acting, and other performance forms [41]. In Chapter 2, we provide more background of Chinese Opera and various deep learning methods. In addition, datasets of Chinese Opera are rare and sparse and there are no standardized datasets suitable for sentiment classification. These issues have limited potential research on sentiment analysis of Chinese Opera. Therefore, to fill this gap, we strive to verify whether deep learning methods can be applied to sentiment classification for Chinese Opera. In particular, we want to compare the performance of models trained using different music feature modalities. For this purpose, we first collected labeled Chinese Opera data. Then, considering different feature modalities, we performed several deep learning approaches and compare their respective performance.

1.2 Solutions and Experiments

In this project, we first collected and cleansed the existing Chinese Opera datasets, introduced lyrics as new feature modality, organized the dataset for machine learning tasks. These efforts resulted in the construction of a new dataset entitled the "Opera Singing Dataset-2025". In Chapter 4, we introduce this dataset and describe how it

is formed and explain its characteristics. Using this dataset, we extracted a variety of features, such as raw waveforms, mel spectrograms, and MFCCs. We also augmented the dataset to include lyrics for use as a feature modality. Finally we trained different deep learning models using those features and cross compared the performance among different features and models. In Chapter 5, we explain the experimental design. In Chapter 6, we show results of models that are trained with different features.

1.3 Discovery

In Chapter 7 and Chapter 8, we discuss our discoveries and present potential future applications. Overall, the results from our experiments show that the model trained by textual feature (lyrics) achieves better results compared to other acoustic features. We hypothesized that the performance of the experimental models will achieve only limited success, due to the small size of the dataset, but, nevertheless, the results show the potential benefits of deep learning in the task of sentiment classification of Chinese Opera. We identify important future work, such as dataset expansion and model optimization to further improve the performance, as this will provide more ideas and support for digital analysis and application of Chinese Opera.

Chapter 2: Background

In this chapter, we give the background and relevant information about Chinese Opera, as well as deep learning models and evaluation metrics that are used in this project.

2.1 Chinese Opera

Chinese Opera (Xiqu) is a traditional art form that merges singing and performance. Its origins can be traced back to early periods in China, and it reached its mature form around the Song Dynasty (AD 960-1279). The performance form of Chinese Opera varies, it often involves singing, acting, recitative, instrumental, dance, story-telling, martial arts, acrobatics, costume, and make-up art [40]. Figure 2.1 shows an example of the costume and makeup in a Chinese Opera.

There are over a hundreds genres and regional variations of Chinese Opera, some of them are still popular even in the modern era. For example, these variants include Beijing Opera (Jingju), Henan Opera (Yuju), Jiangsu Opera (kunqu), among others.

Chinese Opera has a role category system that need to be followed, and all of the types feature distinguished voice styles and makeup art. In Beijing Opera, there are four major role types: Sheng (male role), Dan (female role), Jing (colorful-faced role), and Chou (comical role). Each major role also include some subtypes. For example, there are Lao Sheng and Lao Dan signifies older male role and older female role, while Wu Sheng and Wu Dan indicate male warriors and female warriors [36].

Unlike most of the western Opera, which focuses primarily on singing, the voice performance in Chinese Opera is a combination of singing, dialogue (speech), and recitative (speech-like singing). Table 2.1 presents the mean and standard deviation of their fundamental frequency of different roles and voice forms from a study considering four singers [36]. This table shows that even the acoustic characteristics of a same role could be drastically different from each other when performing different voice forms. This demonstrates the complexity of understanding the acoustic characteristics of Chinese Opera.



Figure 2.1: Illustration depicting costume and makeup in Chinese Opera (from [34]).

Role		**Frequency (Hz)**		
		Dialogue	Recitative	Singing
Lao Sheng	Mean	162.7	271.9	282.2
	Standard Deviation	27.1	22.2	34.1
Jing	Mean	147.4	316.5	372.2
	Standard Deviation	41.5	68.7	27.4

Table 2.1: Frequency of different roles and voice forms, considering four singers [36].

2.2 Deep Learning Models

In this section we briefly review several machine learning architectures considered in this work, such as Convolutional Neural Networks, Long Short-Term Memory Networks, and some of their variants, including MobileNets, Bidirectional Long Short-Term Memory Networks, and Convolutional Recurrent Neural Networks.

2.2.1 Convolutional Neural Network

In the field of deep learning, the Convolutional Neural Network (CNN) is a popular network architecture, which has achieved great success in many computer vision tasks [19]. A CNN extracts local features through a series of multi-layer convolutional neural layers, and uses pooling layers to reduce feature dimensionality. This structure effectively exploits the spatial locality of image data and improves the generalization ability of the model.

With the popularity of mobile smart devices, the limitations of traditional CNNs that rely on higher computational resources begin to emerge and, subsequently, have attracted broad interest in academic research. Researchers have proposed a variety of lightweight CNNs to address these limitations. Among them, MobileNets [17] utilized depth-wise separable convolution to decompose the convolution operation into two steps: depth-wise convolution and point-wise convolution. Figure 2.2 shows how MobileNets separate transitional convolution filters into depth-wise convolution and point-wise convolution. This model significantly reduces the number of parameters of the model while guaranteeing model performance, thus reducing the dependence on high computational resources. Furthermore, it indirectly improves the generalization ability of the model, which, to some extent, alleviates problems with model overfitting.

2.2.2 Long Short-Term Memory Network

The Long Short-Term Memory Network (LSTM) [16] is a special variant of a Recurrent Neural Network (RNN). Because of chain derivation, the gradient in conventional RNN is constantly multiplied during back propagation, which will cause it to decay exponentially. This phenomenon is known as the vanishing gradient [3]. LSTM invokes a gate control mechanism to address this problem. An LSTM network can consist of a number of

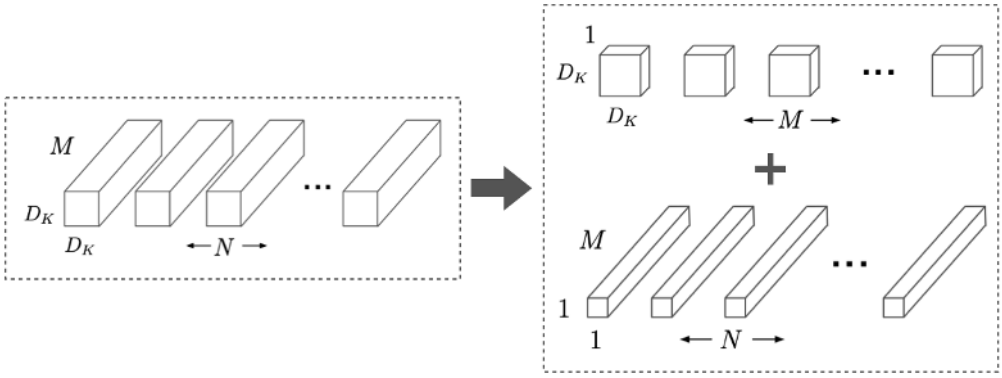


Figure 2.2: The standard convolutional filters (left) are separated into depth-wise convolution filters (top right) and point-wise convolution filters (bottom right). Figure reproduced from [17].

memory cells, and each LSTM cell includes three gates: Forget Gate, Input Gate, and Output Gate [10]. Figure 2.3 shows how these three gates work together in a single LSTM cell. The Forget Gate controls how much information from the previous cell state is retained. The Input Gate determines how much new information is added to the cell state. The Output Gate regulates how much of the updated cell state is passed to the hidden state for the next time step. With this mechanism, LSTM can selectively keep or forget information from long sequential data to better learn the long- and short-term dependencies present in the data.

A Bidirectional Long Short-Term Memory Network (BiLSTM) enhances the feature extraction process of the traditional LSTM by adding consideration of the forward and reverse directions of the sequence [12]. A BiLSTM contains two independent LSTM layers: a forward LSTM and a reverse LSTM process the sequence of length T in temporal order (from $t = 1$ to $t = T$) and in reverse temporal order (from $t = T$ to $t = 1$), respectively. At each time step, the hidden states in both directions are concatenated to form the final output. This bidirectional structure can capture the dependencies of the sequence in both order, which is well suited to handle tasks that need to take contextual information into account, such as semantic analysis and machine translation in natural language processing [33].

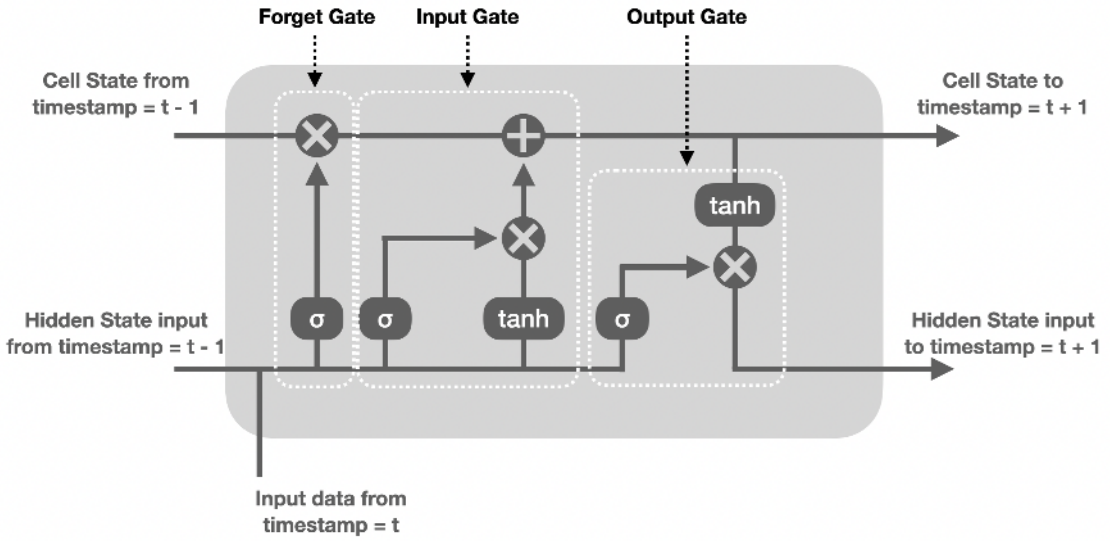


Figure 2.3: A single LSTM cell, the "x" and "+" symbols are the element-wise multiplication and addition operator. "σ" represents the sigmoid activation function. Figure reproduced from [29].

2.2.3 Convolutional Recurrent Neural Network

The Convolutional Recurrent Neural Network (CRNN) [31] is a hybrid network that combines elements of CNN and RNN architectures. It is commonly used to handle tasks with both spatial and temporal dependencies. A CRNN typically has three components: convolutional layers for extracting spatial features, recurrent layers for capturing temporal features, and transcription layers for converting to the target output form. For example, a simple CRNN classification network may consist of a CNN layer, an LSTM layer, and a Softmax classifier layer. The structure of CRNN takes full advantage of the strengths of CNN in spatial extraction and RNN in sequence modeling, and performs well in many sequence recognition tasks, especially in the field of scene text recognition.

2.3 Evaluation Metrics

For classification tasks, the most common and straightforward evaluation metrics is accuracy. However, the use of accuracy by itself may represent the results in cases of

severe data imbalance. Therefore, it is important to also consider other metrics, such as precision, recall and the F_1-score.

2.3.1 Accuracy

Accuracy is defined as the true predictions over the total instances, as shown in Equation 2.1.

$$\text{Accuracy} = \frac{\text{True Predictions}}{\text{Total Instances}} \tag{2.1}$$

For a positive-negative binary classification, which result in true predictions and false predictions, we can consider the concepts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Their explanations are as follows:

TP: Correctly predicted positive instances

TN: Correctly predicted negative instances

FP: Incorrectly predicted positive instances

FN: Incorrectly predicted negative instances

Accuracy also can be defined using these terms, as shown in Equation 2.2.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.2}$$

These terms are also important to define the metrics of Precision, Recall, and F_1.

2.3.2 Precision

Precision measures the proportion of positive instances being predicted correctly out of all positive predictions. Figure 2.4 shows a visualization of this proportion. Precision is defined in Equation 2.3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.3}$$

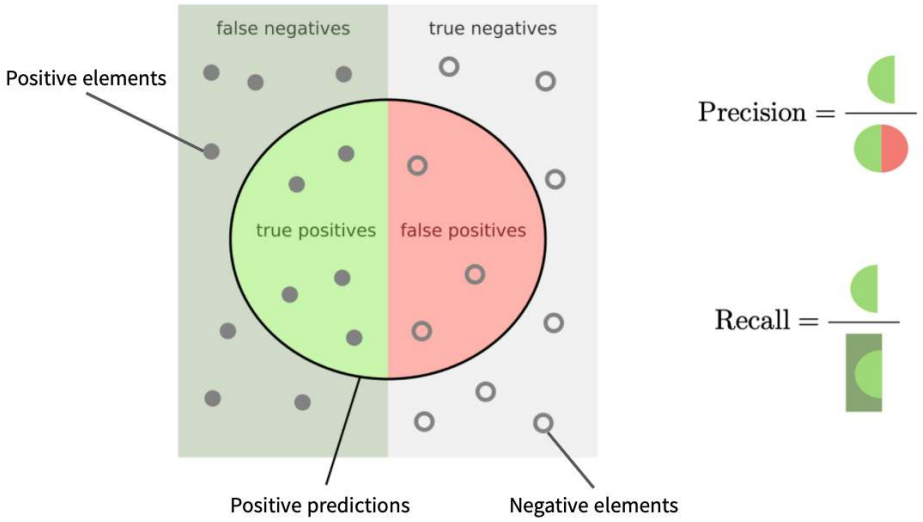


Figure 2.4: A diagram illustrating precision and recall.

2.3.3 Recall

Recall measures the proportion of positive instance being predicted correctly out of all positive instances. Figure 2.4 shows a visualization of this proportion. Recall is defined in Equation 2.4.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4}$$

2.3.4 F_1-score

The F_1-score is the harmonic mean of precision and recall. It is useful especially when there are discrepancies between precision and recall. The F_1-score can be calculated as shown in Equation 2.5.

$$\text{F}_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

Together, these metrics provide a more comprehensive analysis when evaluating a classification model.

Chapter 3: Related Work

In this chapter, we introduce the different audio features and their characteristic. Then we discuss how previous research explore and attempt to solve music sentiment classification using various methods.

3.1 Audio Features

For audio related sentiment analysis, it is crucial to extract meaningful features of the audio. As such, this project will focus on comparing the experimental performance using different audio features. Commonly, audio features can be categorized into two main groups: (1) features based on raw audio and (2) features based on spectrograms. This section will introduce these features including the raw sound waveform, mel spectrogram, and Mel-Frequency Cepstral Coefficients (MFCCs).

3.1.1 Raw Audio

Raw audio is usually stored as a sound waveform, which is a direct representation of the amplitude change of an audio signal in the time domain. This waveform naturally captures the time-domain characteristics of audio. Although raw audio data contains most of the primitive information about the audio signal, using it directly for data science related tasks often comes with high computational costs and, accordingly, experiments using raw audio cannot always achieve optimal results given the high complexity of the signal [9].

3.1.2 Spectrogram Features

A spectrogram is a visual representation that converts an audio signal from the time domain to the frequency domain. Common spectrogram-related features include the mel spectrogram and Mel-Frequency Cepstral Coefficients.

3.1.2.1 Mel Spectrograms

Mel spectrograms are spectrograms based on the mel scale. The mel scale is a nonlinear frequency scale based on the perceptual characteristics of the human ears, which more accurately simulates human perception of pitch. The Mel spectrogram is obtained by performing Short-Time Fourier Transform (STFT) on the audio signal and converting the frequency to the mel scale by applying a transformation to the frequency (see Equation 3.1), where f is the frequency value and m is the corresponding mel scale value. Mel-spectrograms are widely used in music sentiment analysis because it captures the harmonic structure and timbre information of the audio signal while reducing data redundancy [13].

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right) \tag{3.1}$$

3.1.2.2 MFCCs

Mel-Frequency Cepstral Coefficients (MFCCs) are features extracted from a cepstral analysis technique based on the mel spectrogram and are widely used in speech and music signal processing. MFCCs extract key frequency-domain features by converting the Mel spectrogram into the cepstral domain. Specifically, MFCCs enhance the details of the low-frequency components and compresses the dynamic range of the data through logarithmic operations, while reducing the dimensionality of the data by discarding higher-order coefficients during the Discrete Cosine Transform (DCT) [24]. This process not only reduces the computational complexity but also enables MFCCs to capture the spectral envelope, which represents the overall shape of the spectrum. Since the spectral envelope is closely related to timbre and emotional expression, MFCCs have significant application value in tasks involving music sentiment analysis [37].

3.2 Sentiment Analysis and Classification

To recognize people's sentiment, early researchers focused on features from physiological methods. Picard analyzed volunteers' sentiments under different circumstances with wearable devices [27]. Liu [22] defines sentiment computing as an interdisciplinary science that fuses cognitive psychology, natural language processing, and multi-modal recogni-

tion knowledge.

In recent years, sentiment analysis has become an popular and important area of research in the field of artificial intelligence and natural language processing, specifically, in tasks that focus on understanding and analyzing which emotions are expressed in multimedia sources.

3.2.1 Traditional Machine Learning Methods

Historically, traditional machine learning methods have been used for capturing sentiment information. Yang et al. [42] applied a regression approach to classify music emotion into the two dimensional arousal-valence plane, allowing for more precise emotion retrieval as compared to categorical models of emotions. The authors achieved an R^2 statistic of 58.3% for the arousal dimension and 28.1% for the valence dimension with Support Vector Regression (SVR). Colace and Casaburi [7] used the same arousal-valence plane and a focus on regression techniques. Notable, they found that Additive Regression yielded the best results for both valence and arousal with lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) compared to other techniques like Linear and Radial Basis Function (RBF) Regression. With regard to the impact of feature engineering, Mokhsin et al. [26] extracted audio timbre features from both vocal and instrumental parts of songs, training an Artificial Neural Network. Their results indicated that using both vocal and instrumental timbres improved emotion classification accuracy compared to using either modality alone.

3.2.2 Deep Learning Methods

In the context of the rise of deep learning approaches and technologies, the paper titled "A Review of Sentiment Semantic Analysis Technology and Progress" by Wang et al. [39] provides a comprehensive discussion of sentiment analysis methods, categorizing them into single modality and multimodal approaches. This review discusses how different modalities (e.g., text, audio and images) can be integrated for comprehensive sentiment classification. For example, Cai et al. [6] developed a method that using different types of CNN architectures to fuse text-image feature pairs in social media. Aljanaki and Soleymani [1] introduced the MediaEval Database for Emotional Analysis

in Music (DEAM). Their findings suggest that RNN-based approaches combined with large and well-annotated dataset are most effective for dynamic Music Emotion Recognition (MER). Liu et al. [23] utilized spectrograms as features and employing CNNs for emotion classification. With the use of spectrograms, the proposed CNN model can automatically learn relevant features. Their results show that the CNN model can outperform a traditional Support Vector Machine (SVM) classifier, achieving an average accuracy of 72.4% compared to 38.5% for the SVM.

In recent years, more and more research has focused on using deep learning for emotion classification task. Sarkar et al. [30] presents a novel dataset named `JUMusEmoDB`, which includes 400 audio clips of Indian Classical Music with binary emotion classes: *happy* and *sad*. The research demonstrates the effectiveness of using some deep CNN architectures in music emotion classification with Mel spectrograms. Mate et al. [25] focused on emotion classification of songs employing CRNN by combining CNN and LSTM. They achieved the highest accuracy with CNN at 92.71%. These studies highlight the advantages of CNNs for music emotion classification, while also acknowledging limitations related to dataset size and dataset understanding. Zheng [2] utilized Time Convolutional Neural Networks (TCN) and LSTMs to process the time-frequency characteristics of music signals. Additionally, the authors incorporated an attention mechanism to enhance the model's ability to focus on key emotional segments in music, which improved the accuracy in emotion classification tasks.

3.3 Pretrained Models

Pretrained models are machine learning models that are pretrained on large data to learn generic feature representations that can be applied to various downstream tasks [14]. This can help reduce the dependency and necessity of large labeled datasets. More and more researchers have found that by pretraining model, the models can be transferred to other tasks to obtain promising results, even if the target task has limited labeled data available [38].

3.3.1 Pretrained Music Model: MERT

In the context of music, pretrained model have emerged as powerful tools for tasks such as music information retrieval and music classification. MERT (**M**usic und**ER**standing model with large-scale self-supervised **T**raining) [20] stands out as a state-of-the-art model designed specifically for acoustic music understanding. MERT is based on self-supervised learning (SSL). Using Residual Vector Quantization-Variable Auto-Encoder (RVQ-VAE) and Constant Q-Transform (CQT) as teacher models, MERT utilizes Masked Language Modeling (MLM) to randomly mask some of the features and is trained on large-scale unlabeled music data. Whereas RVQ-VAE extracts acoustic features such as timbre characteristics, CQT captures musical information such as pitch. Therefore, MERT can effectively learn the contextual correlations in audio, achieving a deep understanding and generalization ability of music. However, there is no specific published research utilizing MERT for sentiment classification. This project will test the performance of using MERT for sentiment classification in Chinese Opera.

3.3.2 Pretrained Language Model: BERT

Since lyrics, an important dimension in Chinese Opera, can express the sentiment of the opera alongside the sung melody, we want to consider lyrics as one of the feature modalities in this research. In relation to pretrained language model, BERT [8] captures semantic information in text through bidirectional contextual modeling, which relies on MLM and Next Sentence Prediction as pretraining tasks. This allows BERT to learn rich language representations for a variety of NLP tasks, including sentiment analysis. Souza and Filho [35] conducted a comprehensive study on pretrained and fine-tuned BERT for sentiment analysis in excerpts written in Brazilian Portuguese. They also explored various aggregation strategies for embeddings produced by BERT's output layer. Their results emphasized the effectiveness of BERT and its variants, showing the potential of BERT for sentiment tasks.

Chapter 4: Dataset

In this research, we first considered a singing voice audio data set called Singing Voice Audio Dataset from Isophonics, a resource from the Centre for Digital Music (C4DM) at Queen Mary, University of London [4]. C4DM is an influential Digital Music research group. The purpose of this dataset is to solve the impediment of the lack of data in the singing audio analysis area. Later on, we incorporated another dataset, the Jingju a cappella singing dataset [11] and integrated it with the previous dataset in order to address some data problems in the previous dataset. The following sections will introduce each of these two datasets, along with the reasons and the method of integrating them. Ultimately, we released this integrated dataset named the Opera Singing Dataset-2025.

4.1 Singing Voice Audio Dataset

The Singing Voice Audio dataset contains 70 songs sung by 28 professionally trained or amateur singers, totaling 139 audio recordings. There are three main genre among all the songs: Chinese Opera (Beijing Opera and others), Chinese modern song, and Western (European) Opera. All the data are provided as raw audio files (`.wav`). The authors also supply an Excel spreadsheet with metadata and features of the songs. Table 4.1 illustrates two rows of example of the dataset spreadsheet. This table is transposed due to its length. The first column includes the name of the features, the second column includes their corresponding values. We performed some global changes to achieve a more a professional and consistent formatting. For example, we standardized the capitalization and fixed typographic errors.

Since our research will focus on Chinese Opera, seven Chinese modern song examples are excluded from the working dataset as non-opera data. For the convenience of potential future work, the examples of Western Opera are retained but will not be used in model training in our experiments. After these changes, we found several more inconsistencies between data and the spreadsheet during the process of data profiling and cleansing the data.

Name	*male_08*	
File	polyphonic/western/male_08 monophonic/western/male_08	
No. of positive songs	4	
No. of negative songs	0	
Singer name	Juntong Pan	
School	Central academy of drama	
Singer gender	Male	
Singer nationality	Chinese	
Singer level	unprofessional	
Mono	yes	yes
Poly		
Singing type	Western Opera	Western Opera
Title of the song	Non piu andrai, farfallone amoroso	Vaga luna che inargenti
Song file name	pos_7, pos_8	pos_5, pos_6
No. of recordings	2	2
Musical accompaniment	no	no
Emotion	Positive (persuading, enthusiastic, commanding)	Positive (Praising the moon and the lovely night.)

Table 4.1: Example excerpt of the dataset spreadsheet. Some Mandarin text is removed or translated.

4.2 Data Problems and Solutions

In this section, we examine the data problems and mistakes we encountered in this dataset, and the solutions we applied to these problems.

4.2.1 Formatting inconsistency

As we mentioned in last section, we made some changes given the inconsistent and unconventional formatting in the original table provided by the author. These errors include:

1. **Capitalization inconsistency**:

   Some feature names are capitalized, and some are not. We fixed these inconsistencies by standardizing all the feature names to be capitalized.

2. **Typographic mistakes**:

   In the file of metadata, some directory are given using backslashes instead of forward slashes. We fixed this by standardizing them into forward slashes, as is the standard in Linux operating system.

3. **Use of color filling**:

   For the feature `Mono` (monophonic) and `Poly` (polyphonic), the author used color filling to indicate the value: filled with green means "yes", whereas no filling means "no". This is not a standard way to show the values of a data feature and is difficult for a computer to automatically interpret. Therefore, we fix this issue by replacing the color with text value "yes" to explain the attribute and we leave it empty to indicate "no".

After realizing the dataset might not be well composed, we manually examined every data entry and the audio files in this dataset to verify this dataset for consistency and integrity.

4.2.2 Entry Mismatching

We found that there are some data documented in the spreadsheet do not show up in the audio set. Vice versa, there are some audio files in the audio folder do not have a row of features listed in the spreadsheet. We fixed the latter problem by manually labeling the entries. However, the former problem is unsolvable, as we do not have the data, so we simply removed those data entries.

In addition, there are some audio files that should belong to the same song, but are labeled as a different and given an incorrect song title instead. In this case, after getting the correct song information, we merged those audio together and gave them the correct labels.

We used information for correcting this dataset from `https://www.xikao.com` and `https://www.jingju.com`. However, there is one song for which we cannot find any pertinent information, and we excluded it from the working dataset.

4.2.3 Audio Trimming

After examining each audio in the dataset, we found that some of them had a few seconds of silence at the beginning or end of the excerpt. To avoid these silence from affecting the training or evaluation of our models, we trimmed these silenced parts.

4.2.4 Preparing for Machine Learning

Even after fixing those improprieties and mistakes, this spreadsheet is still not ready for processing by a computer for our machine learning tasks, even if all the information is correct. First of all, the spreadsheet is in an Excel format with rich formatting, which, for example, contains merged cells. Also, it is gives personal information for singers (e.g., name), and treats audio (e.g., song file name) as one of the features of the singers (see Table 4.1). However, we desire to treat the audio as the data, and the rest of the information as features or metadata. Accordingly, we also changed the file structure in which the audio is saved. The audio files were originally saved in individual folders, differentiated by singer. We change this structure to organize them by song. Figure 4.1 is an illustration comparing the directory tree representations between the old and new folder structure. These numerically named folders have numbers that correspond to the

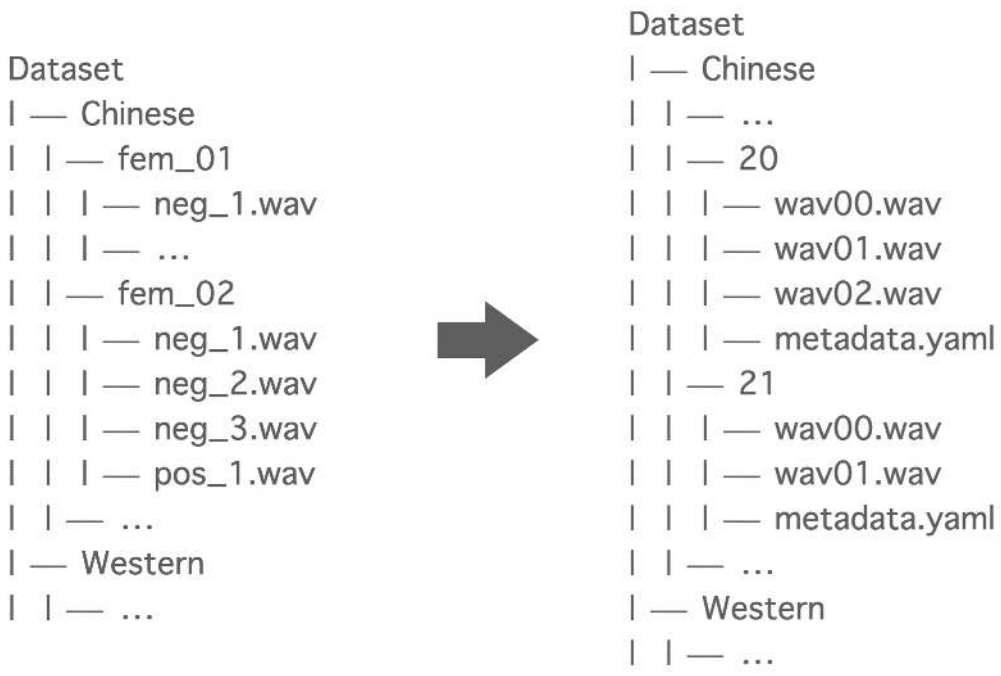


Figure 4.1: Depiction of the original dataset directory tree (left) compared to new dataset directory tree (right).

song identifiers (Song ID) that we assigned.

Because of the nature of these audio data which contain multiple pieces of information, we used metadata structure to store features for each song. Specifically, we used the `YAML` file format to do so. Below is an excerpt of the template with explanation of our `YAML` metadata file, although some entries have been excluded due to the space constraints:

```
emotion_binary: 1/0 (Positive or negative)
files:
  wav00:
    file_dir: Directory to this wav file
    info:
      bit_rate: Bit rate
      channel_number: Number of the channels
```

```
      duration: Audio duration
      if_a_cappella: True/False (if it is a-cappella)
      sample_rate: Sample rate
    singer:
      bio_gender: Mal/fem (male or female)
      id: singer ID
      level:  Professional/intermediate/amateur
      name: Singer name
  wav01:
    ... ...
language: Ch/we (chinese or western language)
singing_type:
  role: Sheng/dan (gender of the role)
  singing: Jingju/yuju/...
song_dir: Directory to this song
song_id: Song ID
... ...
```

We use an automated script to generate all the metadata for the whole dataset according to the corrected information provided by the Excel file. Then we set aside this Excel file. `CSV` files with target class for training will be generated and maintained within each song's folder. We provide more information of how the `CSV` file is generated and maintained in Section 4.5.3.

4.2.5 Insufficient Data

After fixing the data problems above, including the removal of irrelevant data and merging the necessary recordings together, this dataset contains 55 songs and 123 recordings accordingly (46 songs and 103 recordings for Chinese Opera). All the features are stored in `YAML` files. The size of this dataset is still objectively small and insufficient for exploration with machine learning. Hence, we found and incorporated another data set, the Jingju a cappella singing dataset [11], to add more data. In the next section, we introduce this dataset, and discuss how we integrate it into our new dataset.

4.3 Jingju A Cappella Singing Dataset

Jingju a cappella singing dataset is another dataset of Beijing Opera. The recordings are performed by a group of singers, some of whom may have had professional training. While some of the features in this dataset are not particularly helpful to our research, this dataset also includes song titles, scene titles, and role types as labels, which are useful information for this research and help us to enlarge the original dataset. In order to do so, we retrieve the audio file from this dataset and create `YAML` data annotations and place them into the proper folder structure accordingly. This dataset is divided into multiple parts by the author. Because each audio file needs to be inspected and manually assigned to a song, only `part1` is used in this research project due to limited time.

4.4 Adding lyrics

To factor in another modality of features that differ from those extracted from the audio, we decided to consider lyrics because lyrics convey the emotion in opera along with the music. Most information about the operas are publicly online, so these lyrics can be easily collected from the internet. After collecting and labeling all the original lyrics to the dataset, we first tested the use of Google Translate and ChatGPT to translate the lyrics from Chinese to English. However, because some of the opera lyrics are written in an older form of Chinese, the language often differs from modern Mandarin. As a result, neither Google Translate nor ChatGPT achieved a high quality translations. Therefore, we used ChatGPT to first translate the old Chinese into modern Mandarin using prompt "`Translate these old Chinese into modern Mandarin: {original lyrics}`", then using prompt "`Translate these Chinese lyrics into English: {output from the first prompt}`" to translate the lyrics into English. By using these technique, the English translation make more sense and better conveys the original feelings of the opera.

4.5 Opera Singing Dataset-2025

After fixing the numerous problems with the data and merging those two dataset together, we released this modified dataset as the Opera Singing Dataset-2025. In this section, we discuss the data and the features of this merged dataset.

Duration (Sec)			
Maximum	Minimum	Mean	Standard deviation
469.46	17.82	154.27	84.33

Table 4.2: Duration statistics of all Chinese Opera recordings

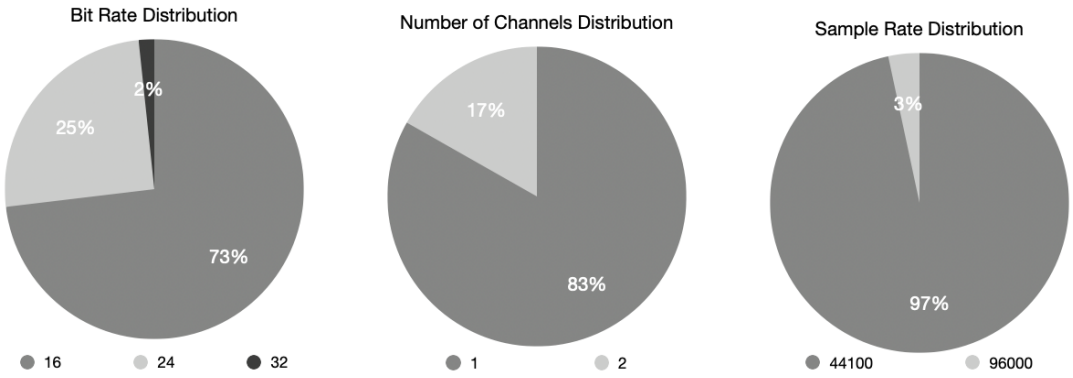


Figure 4.2: Depiction showing the bit rate distribution (left), number of channels distribution (middle), and sample rates distribution (right) of all Chinese Opera recordings.

4.5.1 Data Profile

This dataset contain 61 songs (52 Chinese Opera songs and 9 western opera songs), which has 139 recordings in different versions accordingly (119 for Chinese Opera songs and 20 for Western Opera songs). Compared to the original dataset, we have increased the number of songs by 10.91% (13.04% for Chinese Opera only), and increased the number of recordings by 13.00% (15.53% for Chinese Opera only).

We use `sox`[1] command tool and the `pysox`[2] Python package to examine and process all the audio data. Because this project only focuses on Chinese Opera data, we profile the specifications of only the Chinese Opera recordings, including duration, bit rate, number of channels, and sample rates. Table 4.2 shows the statistics of the durations. Figure 4.2 shows the bit rate, number of channels, and sample rates distributions respectively.

In order to keep all the audio data consistent, we unify all the audio into the same

[1] `https://sourceforge.net/projects/sox/`
[2] `https://pypi.org/project/pysox/`

format using `sox` command tool: single channel, 16000 sample rate, and 16 bit rate.

4.5.2 Data Features

In the `YAML` meta data file, we consider those entries as features, which may be fed into models as input features for learning, also could be picked as classification target:

`emotion`

Description of the emotion of the song. For example, *worried* or *happy*.

`emotion_binary`

A binary representation of the emotion of the song. 0 represents *negative*, and 1 represents *positive*.

`if_a_cappella`

Flag indicating if this recording has music accompaniment or not. A cappella is a term from choral singing, which means that the singing voices stand alone, without musical instruments. So "true" indicates this piece is without music accompaniment, whereas "false" indicates this recording has music accompaniment.

`singer - bio_gender`

The gender (at birth) of the singer who sings on this recording in which "mal" represents male, and "fem" represents female.

`singer - level`

The opera singing level of this singer. There are three level in this dataset:

1. Amateur: the performer has no opera singing experience.
2. Professional: the performer has professionally training in opera singing.
3. Mixed: A mixture of Opera Piaoyou. Piaoyou is a Chinese Opera term which means an enthusiast in Chinese Opera. Piaoyou all have the skills to perform on stage but may not be trained professionally. Due to this fact, we also consider Piaoyou as professional because they all have the same level of skill when singing opera.

`language`

The language in this opera. "ch" represent Chinese, and "we" represent Western languages.

`singing_type - role`

The role type of the character in this audio piece in Chinese Opera. This feature includes Sheng and Dan. That also represent the gender of the role that the singer is playing. Sheng represents male character while Dan represents female character (see details in Section 2.1). This feature is missing for Western opera because this is a unique characteristic of Chinese Opera.

In Chinese Opera, it is common for a singer to perform for a character who has an opposite gender. Therefore, singer's gender sometimes are not always aligned with the role's gender.

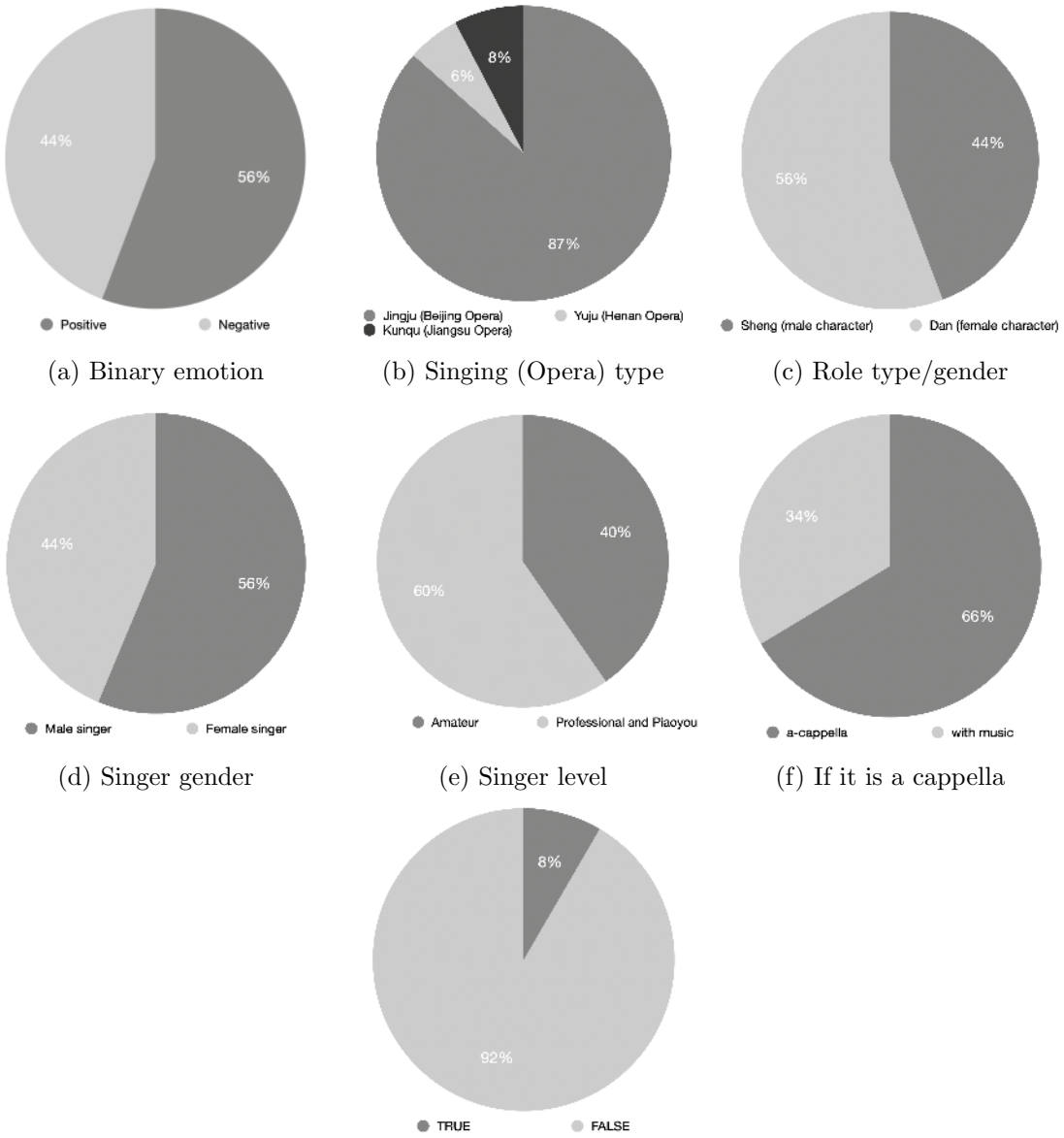
`singing_type - singing`

The type of the Chinese Opera. It includes Jingju (Beijing Opera), Yuju (Henan Opera), and Kunqu (Kun Opera). This feature is absent for Western opera because this feature is only referencing Chinese Opera.

The distributions of these features are shown in Figure 4.3. Note that the distribution for binary emotion, singing type, role type are listed by song, because those features stay the same no matter how they are performed differently for each song. Whereas, singer gender, singer level, if it is a cappella, and if the singer is performing a role with an opposite gender are given for each recording because those features are related to how an individual performer performs a song.

The rest of the entries in `YAML` are considered metadata, and will not be fed into any machine learning model. This includes, for example, the file path, bit rate, audio length, singer name, etc.

Outside the `YAML` file, the raw audio itself will be processed and used as features to our machine learning models. We will be comparing several representations of the audio as inputs to our models.



(a) Binary emotion

(b) Singing (Opera) type

(c) Role type/gender

(d) Singer gender

(e) Singer level

(f) If it is a cappella

(g) If the singer is performing a role with an opposite gender

Figure 4.3: Visualization of the distributions of the features in the dataset.

4.5.3 Target Class

The target class is the class that a machine learning model aims to predict when the task is being defined. In this research, our task is sentiment classification, so binary emotion ("positive" or "negative") is assigned as our target class. However, some other data features can be assigned as target class as well for some other tasks. For the convenience of possible future work, we generated `CSV` files for multiple classes, such as binary emotion, singer gender, singer level, role type, if it is a cappella, and if it is Jingju. The reason we picked these classes is that we observed that these features have approximately balanced distribution (see Figure 4.3). An example of a `CSV` file is below:

```
name,emotion_binary,bio_gender,level,role,acappella,jingju
wav00.wav,0,1,1,1,1,1
wav01.wav,0,1,1,1,1,1
wav02.wav,0,1,1,1,1,1
```

Our script takes an argument to determine which class from the `CSV` file is used as target class. In this project, we only consider the `emotion_binary` row as the target class.

4.6 Summary

During data collection, we collected Chinese Opera singing data from Singing Voice Audio Dataset and Jingju a cappella singing dataset, fixed existing problems with the data, and added lyrics as a new feature. We also created `YAML` metadata for easier feature retrieval, then named this revised dataset as Opera Singing Dataset-2025. Last, we checked the profile and feature distribution of the dataset. Although it is still a small dataset, it is in a proper state of organization for training and evaluate deep learning models.

Chapter 5: Experimental Design

In order to compare the performance of our machine learning models learning among different feature modalities, we use a cross validation strategy with a folds aggregation strategy to evaluate the models.

For model training efficiency and performance, we segment the audio in the dataset into 30-second segments for training and evaluation. Because there are multiple versions of the same opera piece exist in the dataset, we need to make sure only one opera piece be assigned into any individual fold. We designed a task-specific splitting strategy to address this problem. Furthermore, we designed a 3-level evaluation due to the nature of out dataset structure: segment level, recording level, and song level.

This Chapter will focus on the experimental design. The specific models, experiments, and results will be discussed in Chapter 6: Experiments.

5.1 Audio Segmenting

In audio-related machine learning tasks, segmenting audio of varying lengths into fixed-length segments for training is a widely adopted preprocessing strategy [28]. This practice can simplify the design of model architectures, for example, avoiding the need of dynamically adjusting the network when training CNNs and LSTMs. In addition, learning with small segments can also improve training efficiency by reducing the amount of computation per iteration, thus potentially speeding up model convergence. Finally, by segmenting long audio into several segments, the training dataset can be expanded which will specifically help address the limitation of insufficient data using small datasets. Therefore, segmenting not only optimizes the model training process, also provides important support for improvements in model performance [15].

In our pipeline, we made a script to segment the audio into desired length. We also designed multiple solution options for residual segments (i.e. when the last segment that is shorter than the desired length) such as silence padding, circular padding, and dropping.

Silence padding : Pad the residual segments with silence until the desired length.

Circular padding : Pad the residual segments with the beginning of the audio until the desired length.

Dropping : Drop the residual segments.

In this project, based on preliminary experiments, we chose 30 seconds as segment length and silence padding as the residual solution. The 119 Chinese Opera recordings resulted in 535 segments following the segmentation procedure.

5.2 A Specialized Folds Splitting Strategy

In this section, we first explain why we need a specialized strategy in Section 5.2.1 and Section 5.2.2. We discuss the designed strategy in Section 5.2.3.

5.2.1 Information Leakage

Although the data is in the form of audio recordings, some of these recordings represent the same song. If we simply shuffle the whole dataset and split it into training set and test set, it is inevitable that a recording in the training set and another recording in the test set will be performances of the same song. This means they will share the same melody and lyrics. Having those recordings in both training set and test set will cause information leakage [32]. Hence, a typical data shuffle and split strategy will not work well with this dataset. To solve this problem, we only shuffle and split data among different songs in order to keep all recordings of the same song in the same fold.

5.2.2 Fold Imbalance

Figure 5.1 shows the number of recordings of each song, reflecting how many different recordings represent each individual song. As we can see, different songs have different number of recordings. Meanwhile, if we check the total duration of the recordings that represent each song, Figure 5.2 shows that this variable can vary from about 100 seconds to 1000 seconds. These two aspects together will lead to another problem. When we segment each recording to equal length, the number of pieces of each song will have

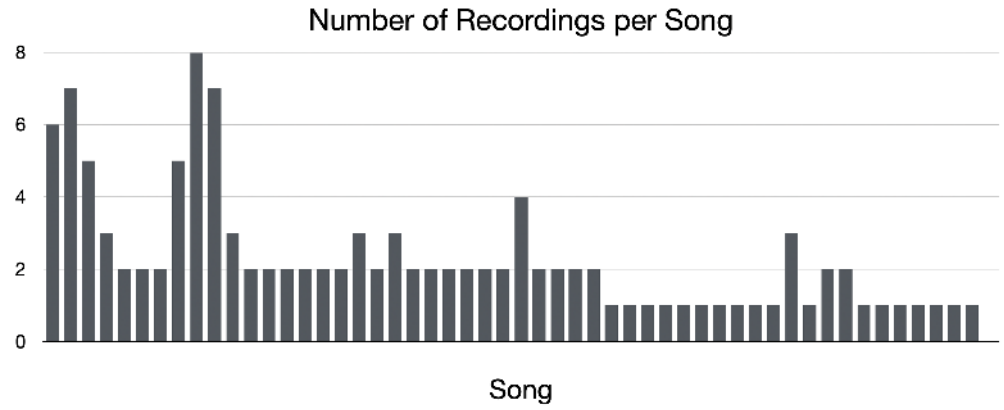


Figure 5.1: The number of recordings that represent each song.

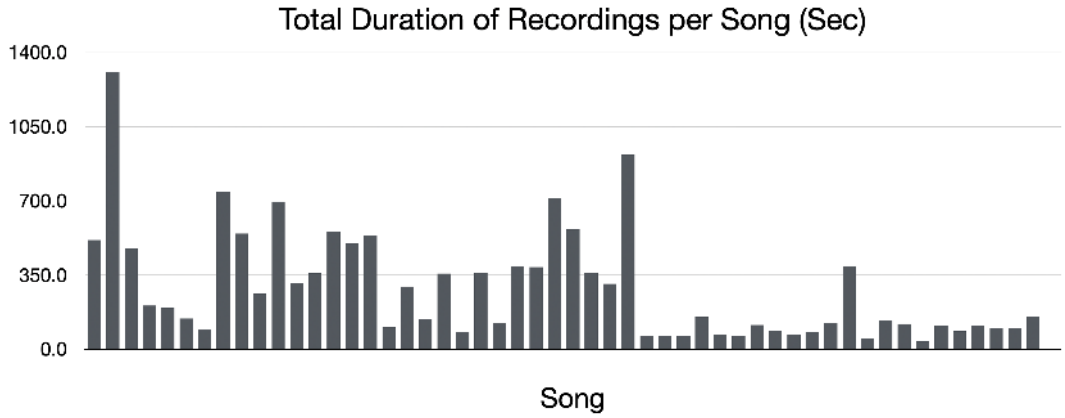


Figure 5.2: The total durations of each song in the dataset.

significant difference. For example, if we split each recording into 10 seconds segments, the amount of segments that represent one song will vary from about 10 segments to 1000 segments. Therefore, it is hard to control the test set proportion and independence. When it comes to cross validation, it is even harder to balance the size of each fold.

5.2.3 Solution

To solve these two problems discussed above, we develop a algorithm to split dataset in a particular way which can meet both the two requirements: balance of the size of each

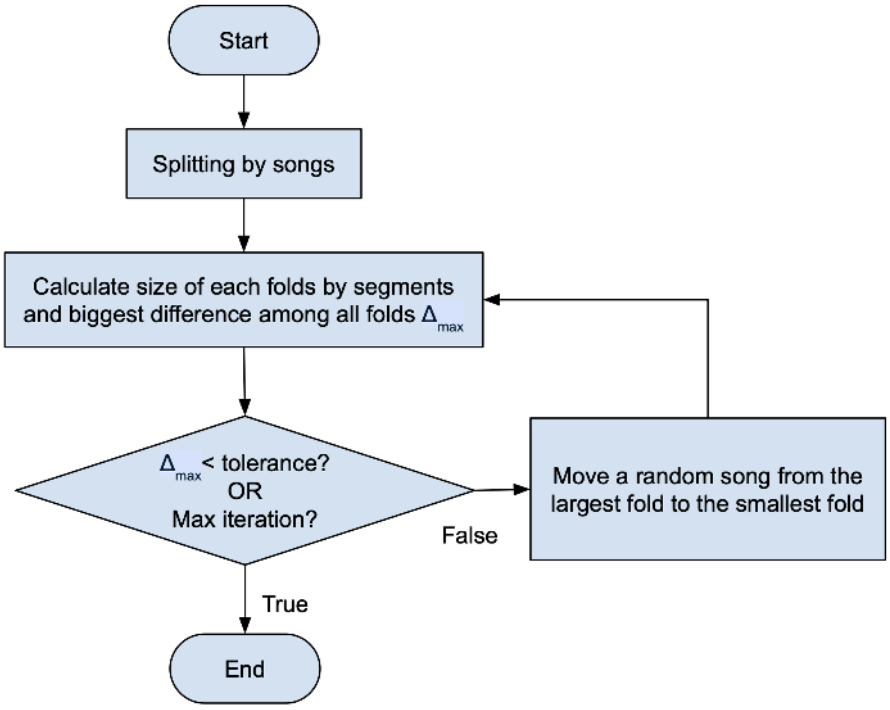


Figure 5.3: Flow chart depicting the fold splitting strategy.

fold and the prevention information leakage.

After segmenting all the songs audio into specific length, we first split the data regarding to songs into N folds. Since we have 52 Chinese Opera songs, each fold will receive around $52/N$ songs. Then we compute how many pieces are they in each fold, we called it the **real size** of a fold. Next we compute the biggest difference among all the folds. If this difference is bigger than our set tolerance, we will move a random song from the fold that has the biggest real size to the smallest one. Then we compute the biggest difference again, continuing until the biggest difference is not bigger than the tolerance or the iteration reaches the maximum number of iterations. Figure 5.3 depicts a flowchart of this fold splitting strategy.

5.3 3-level evaluation method with Voting

When performing cross validation, a direct evaluation is needed to check the segment accuracy. However, the performance among segments can not represent the performance of the model because the purpose of the model is to predict the sentiment of a recording instead of a segment. Therefore we proposed a 3-level evaluation method for this situation. The 3-levels are: segment level, recording level, and song level.

Segment level: Segment level is the evaluation among all the segments of the dataset, in which each segment of audio is treated as one data point. This is the default level used in training and inference of a model.

Recording level: After getting the predictions of segments, we aggregate the prediction by voting from the segments that represent the same recording. The majority prediction will be the prediction for one recording.

Song level: Similarly, we aggregate the prediction from recordings that are belong to the same song by voting. The majority prediction will be the prediction for one individual song.

Figure 5.4 is an illustration of an example of this 3-level evaluation. Some recordings represent the same song, and one recording will be split into segments. Segments are the data directly passed into the network for training and inference. Segment level evaluation is evaluating the performance directly on segments. Recording level evaluation and Song level evaluation are evaluating the aggregation of the predictions that are from the same recording or same song.

During training time, only segment accuracy can be evaluated in real time. During validation and test time, each recording or song is an independent data that should be analyzed. However, we argue that for a same opera but different recordings, especially when they are sung by different singers or using different instruments, may convey different emotions. Furthermore, the aggregation by songs will drastically reduce the data set which will reduce the fairness. Therefore, the evaluation in this project will focus only on the recording level.

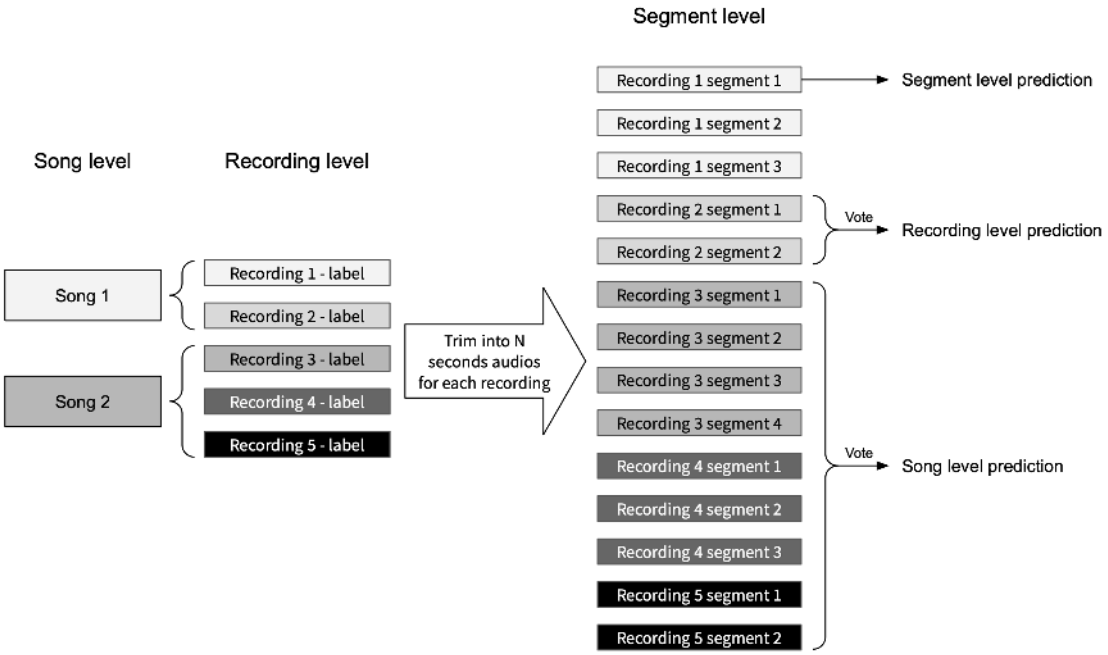


Figure 5.4: Example illustrating the three levels of evaluation.

5.4 Folds aggregation

In order to view a more comprehensive evaluation of an experimental approach, at the very end of every cross validation experiment, we aggregate the predictions from all folds and compute the final scores for the method.

Chapter 6: Experiments

Traditional approaches for audio classification often use mel spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs) as features. These features capture the time-frequency characteristics of audio. However, such methods, which are techniques for dimensionality reduction, may overlook certain information present in the raw audio signal. To address this, we aim to investigate the impact of using raw audio waveforms directly, and compare to the use of spectrograms and MFCCs as input to machine learning models. Additionally, we explore the performance of using the lyrics of the opera as an input feature space.

6.1 Spectrograms Features

Although the use of spectrograms have achieved successes in music sentiment classification in previous research, we question whether they are appropriate for the Chinese Opera dataset given the unique characteristics of Chinese Opera that we discussed in Section 2.1. Therefore, in this section, we will investigate the performance of spectrogram features with CNN and its variant. In our experiments, we utilized the `TorchAudio` transformation to obtain mel spectrograms and MFCCs.

6.1.1 CNN

Since CNNs are designed to handle two-dimensional data, we first selected 2D CNNs for mel spectrograms and MFCCs features. We configured a 4-layer CNN with output size of 32, 64, 128, 256, and kernel size of 5, padding of 2, with max pooling using a kernel size of 2.

The results in the upper portion of Table 6.1 show that models trained with MFCCs show a slightly better performance than those using mel spectrograms on our task. The reason could be that MFCCs are a significantly more compact representation and show a greater robustness to noise, which may serve as advantages to the the learning

	Features	Epochs	Accuracy	Precision	Recall	F_1
CNN	Mel spectrograms	500	0.50	0.56	0.59	0.57
	MFCCs	500	**0.52**	**0.58**	**0.60**	**0.59**
CNN-LSTM	Mel spectrograms	200	**0.56**	**0.59**	**0.73**	**0.65**
	MFCCs	200	0.45	0.49	0.55	0.52

Table 6.1: Comparison of results between CNN and CNN-LSTM.

process. However, neither feature set achieved an overtly successful result. These results demonstrate the limitation of traditional sentiment methods on Chinese Opera using a limited dataset.

6.1.2 CRNN

In order to attempt to capture more temporal information, we construct a CRNN model by stacking two sequential 2D CNN layers with two sequential LSTM layers.

When using this CNN-LSTM architecture, the result demonstrate a better performance than using CNN solely with mel spectrogram features, presented in Table 6.1. This indicates that by adding in LSTM to model temporal information, the model was able better able to learn from the spectrograms features. And because MFCC is more compact representation and, thus potentially losing some of the temporal information, make it less suitable for LSTM models that rely on temporal information alone.

6.2 Raw Audio Features

To investigate the performance of machine learning models that use raw audio signals, we utilize `TorchAudio` to directly extract waveform signals from the audio, which are then used as features. Since the audio was previously unified into single channel in Section 4.5.1, the waveform signals are one-dimensional. For these one-dimensional features, we experimented with CNNs, LSTMs, and a pretrained transformer model.

	Epochs	Accuracy	Precision	Recall	F_1	Model Size
Vanilla 1D CNN	500	0.56	0.65	0.54	0.59	0.2 M
MobileNets-OG	500	0.50	0.57	0.57	0.57	3 M
MobileNets-simple	500	0.55	0.61	0.61	**0.61**	2 M
MobileNets-small	500	0.50	0.57	0.51	0.54	0.17 M

Table 6.2: Comparison of results between Vanilla CNN and MobileNets.

6.2.1 CNN

We configured a 4-layer 1D CNN network with output size of 32, 64, 128, 256, and kernel size of 3, with max pooling with kernel size of 2.

Additionally, we applied the original MobileNets V1 with 14 layers and alongside two smaller variations: a 10-layer simplified version and a smaller 6-layer version.

The results of this experiment are shown in Table 6.2. We found that the simplified version of MobileNets was better able to learn, and outperformed the vanilla CNN with higher F_1 score. This could be because that MobileNets has smaller parameter size, which potentially reduces the risk of overfitting and leading to better generalization. Because our dataset is rather small, that could be a significant advantage.

We can also observe that with depth-wise separable convolutions, the model is more lightweight thus potentially more efficient. Overall, the 6-layer MobileNets network is smaller than the 4-layer CNN.

6.2.2 LSTM

Because raw audio is contextual and temporal data, we design experiments to assess the effects of using LSTM models.

We explored various architectural designs and regularization techniques for LSTM networks. After iterative experimentation and evaluation, three structures were chosen for the final stages of analysis and testing. Base on our preliminary experiments, we constructed an LSTM with only 2 layers to prevent overfitting and ensure training efficiency. Additionally, we consider an optional dense layer. The regularization techniques options that were chosen are L_2 regularization and dropout. Table 6.3 demonstrates the three structure designs. Design 1 has no dense layer in between the two LSTM layers,

	Layer 1	Dense Layer	Layer 2	Regularization
Design 1	Yes		Yes	L_2 ($\lambda = 10^{-2}$)
Design 2	Yes	Yes	Yes	L_2 ($\lambda = 10^{-2}$)
Design 3	Yes	Yes	Yes	Dropout ($rate = 0.3$)

Table 6.3: Structure designs of LSTM Models.

	Design	Epochs	Accuracy	Precision	Recall	F_1
LSTM	Design 1	50	0.55	0.6	0.67	0.63
	Design 2	100	**0.57**	**0.62**	**0.67**	**0.64**
	Design 3	100	0.53	0.61	0.51	0.56
BiLSTM	Design 1	300	0.54	0.61	0.55	0.58
	Design 2	100	0.5	0.58	0.52	0.55
	Design 3	100	0.53	0.60	0.58	0.59

Table 6.4: Comparison of results between LSTM and BiLSTM.

while design 2 and design 3 does use a dense layer between the LSTM layers. Design 1 and design 2 uses L_2 regularization with the regularization parameter λ of 10^{-2}, and design 3 use dropout with rate of 0.3. Additionally, we experimented using the same structures with a BiLSTM model.

The results of these experiments are shown as Table 6.4. We observe that the LSTM with dense layer using L_2 regularization yielded the best result as compared to the other variants. Although BiLSTM models have the ability to process the input in both directions, we observed a poorer performance than the use of standard LSTMs. This could be because the BiLSTM models we constructed suffered from overfitting due to the dataset's small size. Moreover, because music is only supposed to be interpreted in one direction, changing order in a notes progression might affect the understanding of sentiment in the music, and learning with the backward direction might not be as effective as learning with one direction.

6.2.3 Use MERT Embeddings

We used MERT to extract acoustic embeddings of the audio input to use as features. The model we selected is `MERT-v1-95M`. The input is the waveform of the raw audio, the direct output of MERT is a 3-dimensional tensor. The three dimensions represent

		Epochs	Accuracy	Precision	Recall	F_1
CNN	Reduced layer and time	100	0.57	0.59	0.65	0.62
	Reduced layer	100	0.31	0.39	0.48	0.43
	Reduced time	100	0.55	0.58	0.65	0.61
FC	Reduced layer and time	100	0.53	0.56	0.62	0.59
	Reduced layer	100	0.47	0.51	0.72	0.60
	Reduced time	100	**0.57**	**0.58**	**0.70**	**0.63**

Table 6.5: Comparison of results between MERT using CNN or FC as the classification layer.

layers, time steps, and features. For computational efficiency, we explored different ways to reduce the dimensionality. Respectively, we tried to (1) reduce both layer and time dimension, (2) reduce only the layer dimension, and (3) reduce only the time dimension. For the classification layer, we explored the use of a CNN and a fully connected neural network (FC).

These results are shown in Table 6.5. We can see that FC model with reduced time achieved the best result compared to the other models. We also observed that reducing layer reduced model's performance drastically, which demonstrates that the variety of MERT features in different layer levels is more important than the sequential changes in time for this task.

6.3 Lyric Features and Multimodal Learning

After collecting and translating all the lyrics as discussed in Section 4.4, we used BERT to extract features from the English translation of the lyrics' text. Because the case of the text is not relevant to our task, we used `BERT-base-uncased`. We used a fully connected network to classify the latent feature. Similar to the experiment using raw audio with MERT, a fully connected network is used for classify the extracted features.

To investigate the potential of multimodal learning, we experimented with the early fusion method. We fused the latent feature from raw audio with a LSTM layer and the latent features from the lyrics with BERT by concatenating these two feature spaces together. Then, we classified the concatenated feature with a fully connected network. The structure of this multimodal model is visualized in Figure 6.1.

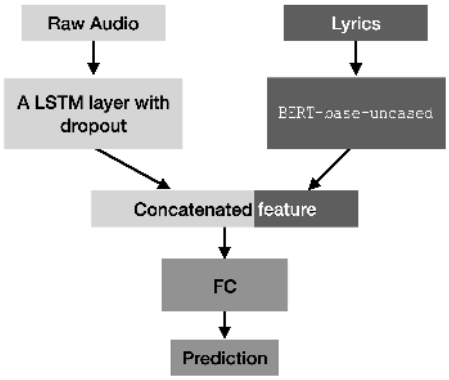


Figure 6.1: Visualization of the Audio-Lyrics fusion model.

	Epochs	Accuracy	Precision	Recall	F_1
Lyrics with BERT	200	**0.67**	**0.69**	**0.73**	**0.71**
Audio-Lyrics Fusion	300	0.50	0.57	0.51	0.54

Table 6.6: Comparison of results between lyrics features using BERT alone and a multi-model approach.

These results are shown in Table 6.6. When using only lyrics feature extracted from BERT, the model achieved better results than when we fuse it with audio data. Because we only investigate one multimodal feature fusion, we believe that more specific fusion designs need to be investigated in order to observe any potential advantages of multimodal fusion. However, these results still demonstrate that fusing features naïvely might not improve performance for this task compared to approaches than focus on a single feature space.

Chapter 7: Discussion

In this section, we discuss our findings and provide a comprehensive comparison of the results we presented in Section 6.

7.1 Cross-feature Comparison

In our experiments, we evaluated the performance of models trained on different features, including spectrograms, raw audio as a waveform, raw audio as MERT embeddings, and opera lyrics. Among these modalities, the first three feature types are considered acoustic features, while lyrics are the only textual feature.

Table 6.1 shows that when using mel spectrogram as training data, the CRNN model outperforms the CNN-only model given its advantage of combining the benefits of both a CNN and an LSTM. For models trained on raw audio, the results given in Table 6.2 and Table 6.4 demonstrate the advantages of the LSTM over the CNN. However, the BiLSTM did not outperform the LSTM despite its consideration of bidirectional dependencies. The results shown in Table 6.5 demonstrate the capability of our downstream task using the embeddings generated by MERT from raw audio. And Table 6.6 shows the results of the downstream task using BERT with lyrics features and late fusion model using both lyrics and raw audio. This comparison shows that our attempted fusion model instead reduces model performance compared to one that is only trained on lyrics features.

In Table 7.1, we show the cross-feature comparison among the best results from each feature. For acoustic features, the LSTM model trained on raw audio and fully connected network trained on MERT embeddings achieved the highest accuracy. However, a CNN-LSTM model trained on mel spectrograms achieved a higher F_1 score, suggests that it yielded a better balance of predicting positive and negative classes. In addition to this, all these best models that were trained from acoustic features showed less difference in performance, indicating that these features sets can perform similarly when selecting the appropriate models and training methods.

When comparing acoustic features and lyrics feature, we can observe that the model

	Features	Method	Accuracy	Precision	Recall	F_1
Acoustic	Mel spectrogram	CNN-LSTM	0.56	0.59	0.73	**0.65**
	Raw Audio	LSTM	**0.57**	0.62	0.67	0.64
	MERT embeddings	FC	**0.57**	0.58	0.70	0.63
Textual	Lyrics	BERT	**0.67**	**0.69**	**0.73**	**0.71**

Table 7.1: Comparison among the best result from each feature set and model.

Number of Songs	Number of Recordings	Accuracy	Precision	Recall	F_1
46	103	0.52	0.61	0.63	0.62
52	119	**0.57**	**0.62**	**0.67**	**0.64**

Table 7.2: Performance of LSTM before and after adding more data.

trained on lyrics outperformed all acoustic-based models across all metrics. This result suggests that the model designed in this project did not manage to learn more sentiment information from various acoustic features than from the textual lyric features alone. This may be caused by the unique acoustic characteristics of Chinese Opera, as discussed in Section 2.1. Some specialized singing techniques and performance styles of Chinese Opera are quite different from other music, thus making it difficult to extract effective sentiment information from common acoustic features. In contrast, expressions of emotions in lyrics are more direct and straightforward, and, as such, models trained on lyrics seem to easily outperform models trained on acoustic features.

7.2 Data Scarcity

In Sections 4.2.5 and 4.3 we discussed how the first dataset we explored, the Singing Voice Audio Dataset, is very small so we needed to integrated another dataset, the Jingju A Cappella Singing Dataset, to create a new aggregate dataset, which we name the Opera Singing Dataset-2025. In fact, before combing the datasets, we also ran some experiments with LSTM models. The best result we achieved using LSTM from before and after merging two datasets is shown in Table 7.2, both are trained on raw audio as waveform. After increasing the dataset size by 13%, the accuracy of the model is improved by 10%. Given that the result is still not ideal, this suggests that we do not have sufficient data to properly train a deep learning model on audio features alone.

However, the observed improvement still indicates that data scarcity is a major challenge for this task and that performance improvement can be achieved by incorporating more additional data.

Chapter 8: Conclusion

This goal of this project is to compare the performance of deep learning models trained on different features of Chinese Opera for sentiment classification given the unique characteristics of this genre of music. We collected all existing datasets but only managed to find very limited and poorly annotated data given the niche popularity of Chinese Opera. We then cleansed, corrected and relabeled the data, and added the lyrics as an additional feature. With these efforts, we created a new dataset named Opera Singing Dataset-2025. In the experimental preparation stage, given the specific organization of the dataset, a typical cross validation setting would lead to information leakage and fold imbalance problems. Therefore, we designed a specialized folds splitting strategy to address these issues. To this end, we defined three levels of evaluation for the dataset: segment level, recording level, and song level and confirmed the use of recording level as the evaluation method.

In our experiments, we compare the performance of various models trained on spectrograms, raw audio, and lyrics. For the acoustic feature, experimenting with different models and methods resulted in models with accuracy of 56%-57% and F_1 score of 64%-65%. While the model trained on the lyrics feature outperform the former models, achieving accuracy of 67% and F_1 score of 71%, demonstrating that it is easier to learn sentiment information from a more straightforward expressive feature such as lyrics, especially when the amount of available training data is low. On the other hand, acoustic features require more specialized model architectures and methods. In addition, the complexity and specific acoustic properties of Chinese Opera adds more difficulty to the task of learning directly from acoustic features.

8.1 Future Work

There is a lot more potential for the continuation of the research explored in this project. First of all, considering the unique characteristic of Chinese Opera, we should focus on the design of features and models according to some domain knowledge in order to

provide more relevance and context to the learning models. Secondly, we discussed how expanding the dataset by 13% resulted in a 10% accuracy improvement in Section 7.2. Therefore, given the availability of new data in the future, we believe that an expansion the size of the training set will only further improve performance. For example, we need to collect and label more Chinese Opera singing audio data from other data sources. Thirdly, similar to how we considered lyrics as a extra modality, images or even videos from the Chinese Opera could potentially also serve as effective modalities since Chinese Opera contains a focus on performance in addition to the singing itself. Finally, due to time constraints, we have only explored direct late fusion across raw audio and lyrics. However, there are many newer multimodal learning methods that have the potential to be applied in our task, such as Intermediate Fusion [5] and Attention-based Fusion [21]. Furthermore, additional features should also be considered into multimodal learning, such as the aforementioned images and videos.

8.2 Final Thoughts

In conclusion, this project evaluated sentiment classification models trained on different features extracted from recordings of Chinese Opera using a cleansed and expanded dataset. Our results shows potential of some of the features, but we found many limitations based on the dataset size and quality. More future work needs to be explored to address those challenges. If the performance is improved further, this project will have various applications, such as automatically cataloging Chinese Opera, automatically generating playlists based on sentiment, and analyses of Beijing Opera culture.

Bibliography

[1] Anna Aljanaki, Yi-Hsuan Yang, and Mohammad Soleymani. Developing a benchmark for emotional analysis of music. *PLOS ONE*, 12(3):1–22, 03 2017.

[2] Zheng Anna. Research on music emotion recognition and intelligent classification system based on deep learning. In *2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, pages 1061–1065, 2024.

[3] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[4] Dawn AA Black, Ma Li, and Mi Tian. Automatic identification of emotional cues in Chinese opera singing. *13th International Conference on Music Perception and Cognition (ICMPC-2014)*, pages 250–255, 2014.

[5] Said Yacine Boulahia, Abdenour Amamra, Mohamed Ridha Madi, and Said Daikh. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Machine Vision and Applications*, 32(6):121, 2021.

[6] Guoyong Cai and Binbin Xia. Convolutional neural networks for multimedia sentiment analysis. In *Natural Language Processing and Chinese Computing: 4th CCF Conference, NLPCC 2015, Nanchang, China, October 9-13, 2015, Proceedings 4*, pages 159–167. Springer, 2015.

[7] Francesco Colace and Luca Casaburi. An approach for sentiment classification of music. In *International Conference on Enterprise Information Systems*, volume 2, pages 421–426. ScitePress, 2016.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[9] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, 2014.

[10] F.A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with LSTM. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, 1999.

[11] Rong Gong, Rafael Caro Repetto, and Yile Yang. Jingju a cappella singing dataset part1, March 2017.

[12] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, 2005.

[13] Thomas Grill and Jan Schluter. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1296–1300, 2015.

[14] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[15] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 131–135. IEEE, 2017.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[18] Xingguo Jiang, Yuchao Zhang, Guojun Lin, and Ling Yu. Music emotion recognition based on deep learning: A review. *IEEE Access*, 12:157716–157745, 2024.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger

Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. MERT: Acoustic music understanding model with large-scale self-supervised training, 2024.

[21] Yuancheng Li, Shenglong Zhou, and Hui Chen. Attention-based fusion factor in FPN for object detection. *Applied Intelligence*, 52(13):15547–15556, 2022.

[22] Bing Liu. *Sentiment analysis and opinion mining*. Springer Nature, 2012.

[23] Tong Liu, Li Han, Liangkai Ma, and Dongwei Guo. Audio-based deep music emotion recognition. In *AIP conference proceedings*, volume 1967. AIP Publishing, 2018.

[24] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *International Society for Music Information Retrieval (ISMIR)*, volume 270, page 11. Plymouth, MA, 2000.

[25] Nikita Mate, Durva Akre, Gaurav Patil, Gopal Sakarkar, and Thomas Anung Basuki. Emotion classification of songs using deep learning. In *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, pages 303–308, 2022.

[26] Mudiana Binti Mokhsin, Nurlaila Binti Rosli, Wan Adilah Wan Adnan, and Norehan Abdul Manaf. Automatic music emotion classification using artificial neural network based on vocal and instrumental sound timbres. In *New Trends in Software Methodologies, Tools and Techniques*, pages 3–14. IOS Press, 2014.

[27] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[28] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019.

[29] Divish Rengasamy, Mina Jafari, Benjamin Rothwell, Xin Chen, and Grazziela P Figueredo. Deep learning with dynamically weighted loss function for sensor-based prognostics and health management. *Sensors*, 20(3):723, 2020.

[30] Uddalok Sarkar, Sayan Nag, Medha Basu, Archi Banerjee, Shankha Sanyal, Ranjan Sengupta, and Dipak Ghosh. Neural network architectures to classify emotions in indian classical music, 2021.

[31] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017.

[32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[33] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. The performance of LSTM and BiLSTM in forecasting time series. In *2019 International conference on Big Data*, pages 3285–3292. IEEE, 2019.

[34] Wang-Ngai Siu and Peter Lovrick. *Chinese opera images and stories*. University of Washington Press, Seattle, 1st ed. edition, 1997.

[35] Frederico Souza and João Filho. Bert for sentiment analysis: Pre-trained and fine-tuned alternatives, 2022.

[36] Johan Sundberg, Lide Gu, Qiang Huang, and Ping Huang. Acoustical study of classical Peking opera singing. *Journal of Voice*, 26(2):137–143, 2012.

[37] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[38] Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across NLP tasks, 2020.

[39] Yiming Wang, Yuan Rao, and Lianwei Wu. A review of sentiment semantic analysis technology and progress. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 452–455. IEEE, 2017.

[40] Elizabeth Wichmann. Tradition and innovation in contemporary Beijing opera performance. *TDR (1988-)*, 34(1):146–178, 1990.

[41] Xiaoning Xiao. The use of traditional opera elements in the characterization process of contemporary stage drama. *Highlights in Art and Design*, 4(1):13–16, 2023.

[42] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H Chen. A regression approach to music emotion recognition. *IEEE Transactions on audio, speech, and language processing*, 16(2):448–457, 2008.