CITY UNIVERSITY OF HONG KONG
香港城市大學

Audio Musical Genre Classification using Convolutional Neural Networks and Pitch and Tempo Transformations
使用捲積神經網絡及聲調速度轉換的音頻音樂流派分類研究

Submitted to
Department of Computer Science
電腦科學系
in Partial Fulfillment of the Requirements
for the Degree of Master of Philosophy
哲學碩士學位

by

Li Lihua
黎立華

September 2010
二零一零年九月

Abstract

Musical genre classification is a potential yet challenging task in the field of music information retrieval. As an important first step of any genre classification system, music feature extraction is a critical process that will drastically affect the final performance. In this thesis, we will try to address two important questions of the feature extraction stage: 1) is there any potential alternative techniques for musical feature extraction when traditional audio feature sets seem to meet their performance bottlenecks? 2) is the widely used MFCC feature purely a timbral feature set so that it is invariant to changes in musical key and tempo in the songs? To answer the first question, we propose a novel approach to extract musical pattern features in audio music using convolutional neural network (CNN), a model widely adopted in image information retrieval tasks. Our experiments show that CNN has strong capacity to capture informative features from the variations of musical patterns with minimal prior knowledge provided. To answer the second question, we investigate the invariance of MFCC to musical key and tempo, and show that MFCCs in fact encode both timbral and key information. We also show that musical genres, which should be independent of key, are in fact influenced by the fundamental keys of the instruments involved. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. We propose an approach to address this problem, which consists of augmenting classifier training and prediction with various key and tempo transformations of the songs. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisor Dr. Antoni Bert Chan for his guidance and suggestion during my study and research at City University of Hong Kong. Due to my slow start at my research topic and the switch of supervisors, it was almost impossible for me to graduate on schedule. When I set out to search for a new supervisor, professors turned me down because of my poor publication background, until I meet Dr. Chan. He picked me up, guided me through the darkest hours of my career. Without his expertise in music research and mathematics, it would not have been possible for me to achieve the conference papers, let alone this thesis. He is a brilliant, knowledgeable and caring advisor. It has been such a honor to study with him.

I would also like to thank Dr. Raymond Hau-San Wong for enlightening me to the field of data mining, and eventually my current research area. I still remember the day I asked him for help on research topics, the way he kindly show me the path to machine learning. His data mining course inspired me in various aspects of my research, and I am impressed by his vast knowledge and strict attitude towards teaching and research.

I dedicate my special thanks to Dr. Albert Cheung, who has been a selfless mentor and a caring friend of mine. He has made available his support in a number of ways, no matter it is about research, career or life. He helps me to raise my self-esteem to reach out for my long forsaken dreams, and he opens portals of opportunity so that I can meet and work with top scientists in the world. He inspired me to think high of the person I ought to be, and the achievements in science that I ought to pursue in my life time.

Thanks also goes to my current and former colleagues in the Computer Science

Department for their support to my work and my life in City University of Hong Kong. Thanks to Mr. Ken Tsang, who has given me keen company in the days searching for research topics; Dr. Xiaoyong Wei, who has presented himself a role model of knowledge and helpfulness. Thanks to Tianyong Hao, Qiong Huang, Linda Zheng, Rebecca Wu, Tiesong Zhao, Hung Khoon Tan, Si Wu, Sophy Tan, Shi'ai Zhu and Yang Sun. Thank you all for making my work colorful and enjoyable.

Last but not least, I want to thank my mother for her support since my birth. Thanks for her devotion and encouragement to my study. Thanks for the endless love she gave me.

Contents

List of Tables

List of Figures

List of Abbreviations

CNN	Convolutional Neural Network
DA	Digital-to-Analog
MFCC	Mel-Frequency Cepstral Coefficient
MIR	Music Information Retrieval
STFT	Short Time Fourier Transform
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Why Automatic Music Genre Classification?

I would like the raise the question at the beginning of this paper: why do we need automatic music genre classification, as is the most frequently asked question when I try to present my research to someone who is not familiar with music information retrieval (MIR). The answer to that question is crucial for this whole paper, and I would like to address it with the following two scenarios.

Scenario 1. John is an IT company engineer. He loves music, and he loves listening to it at work and at home. His favorite MP3 player is filled with songs he obtained from various sources. Some of them are ripped from CDs he bought; some are shared by this co-workers; some are downloaded from online digital music retailers such as iTunes and Amazon. One day he tried to build up a play list of Jazz music because he just develops a strong fond of it recently. He soon discovers it a non-trivial task. Simply sorting the names of the songs brings no solution to the problem. Not only because the genres label ”Jazz” may not appear in the names of the files, but also files from different sources follow different naming conventions, rendering name-based batch processing impossible. Some of his tools is capable reading the meta-information stored in the files. It helps finding the songs with proper meta-information, but it is unhelpful with the rest. Perhaps the most secure way is to listen to the songs one by one to determine its genre. But it is simply mission impossible on his ten-thousand-song collection.

Scenario 2. I-Want-To-Listen-To-Music.com is an online digital music retailer company founded in 2000. The company tries to develop a service to display the songs and albums on its web pages by genres and tags, since it assists the user to navigate the database and potentially increases sales. The task turns out to be very difficult. The company has millions of untagged songs in its database. To provide the new service means to label them all. One solution is hiring a team of experts to classify the songs manually. But it is hardly practical in terms of expense and scalability. The CEO of the company wonders whether he could use computer to finish such a task.

As we can see from the two scenarios above, automatic, content-based music classification systems would naturally have both personal-scale and business-scale applications. With the rapid development of digital entertainment industry, we have easy access to digital music in various forms. Nowadays it is not uncommon to possess an MP3 player that stores thousands of songs. For song database organization and play list generation, we will need the help of meta-information such as musical genres, moods, tags, etc. But those information may not necessarily come with the song file. With the help of an automatic, content-based music classification system, we will be able to assign proper labels to song files, and therefore manage the growing song database conveniently. On the other hand, online digital music retailers would also benefit substantially from those systems. The tremendously large song database will be tagged and sort out by computers. Such solution is inexpensive and scalable. The sales would potentially increase as users find it more convenient to navigate through the database.

Music genre classification is a special case of the more generic music content meta-information recognition/tagging systems. Actually genre is typically a kind of meta-information people used to describe musical contents. Similar meta-information includes instrumentation, tempo, artist, etc. The reasons concentrating our work to genre are two fold. First, the concept genre is very widely used nowadays. When we talk about bands or singers, it would be very intuitive to use genre to describe the bands and the music they produce, as oppose to the instrumentation they use or the tempo of the songs. Although it is impossible to argue that genre is more important than other concepts, I believe it makes a strong case as a candidate of meta-information for song classification. Second, music genre classification systems would share a lot of commonplaces with other music content meta-information recognition systems. Once we build up a reliable genre classification system, we would be able to generalize our work to other types tagging systems with some minus modification of the architecture.

1.2 Scope of this work

The scope of this work is focused on a critical issue of audio musical genre classification: musical feature extraction. The elaboration of this thesis is organized as follows;

Chapter 2 generally describe the research field of MIR and the background of the genre classification task. Fundamentals about sounds and human auditory perception are presented to support the later chapters of this thesis.

Chapter 3 focuses on the application of image techniques on the music genre classification problem. As an important processing step, feature extraction plays a critical role that will significantly affect the final classification performance. However, recent researches [32] shows that using only timbral feature sets derived from traditional speech recognition features will limit the performance of genre classification systems. In this chapter, we try to break through the performance bottleneck, using novel feature sets extracted with image information retrieval techniques. This chapter describes the experiments applying convolutional neural network (CNN), a state-of-the-art image digit recognition algorithm, to automatic extraction of musical pattern features. The system architecture, the characteristics of CNN and the classification performance are explained.

Chapter 4 studies the invariance of the widely used MFCC feature set to musical key and tempo. Musical genre is a complex concept associated with various musical attributes, such as instrumentation, key, tempo, musical patterns, etc. In many previous works [41, 6, 15] , the MFCC feature set is considered to be a timbral feature set that contains solely instrumentation information. Our experiments reveals that, apart from the timbral information, the MFCC feature set also to some extent encodes the key information of the songs concerned. The MFCC feature set is not invariant to change in musical key. Likewise, we also investigate into the distribution of musical keys in the GTZAN dataset [41], showing that genre is key-related based on the fundamental keys of the instrumentations. In Chapter 4, the classification system, experiment set-ups and the detailed performance evaluation are presented.

Chapter 5 concludes the thesis and suggest potential directions for future development.

Chapter 2

Audio Music Genre Classification Systems and Feature Extraction

2.1 Classification systems and their evaluations

Classification is a sub-discipline of data mining research. The task description can be very simple: constructing a system which automatically label the category of an incoming item, given some "features" of the item. For instance, we can construct a classification system which labels unknown flowers with their names, given information such as color, petal length, leave length, etc. Such system can be constructed by hand-crafting, or by some automated algorithms. Arguably, the most commonly used scheme for constructing a classification system is via supervised learning: the classification system is constructed automatically using a learning algorithm and a pre-labeled training set. It saves the trouble and prior knowledge needed to hand-craft the classification system, while the actual performance resulted from the supervised learning process is dependent on the learning algorithm and the classification problem concerned. There is no universal learning algorithm that fits all classification problems.

The evaluation of performance of supervised learning algorithms relies on the classification accuracy. Given a specific data set, it is possible to find a specific learning algorithm that yields excellent classification results. However, such classification results may not be generalizable to the real world problems the classification system intends to solve, for the resulted system fits the given data set “too well”. To overcome such a problem, the given data set is usually split into two smaller data sets, one for training, the other reserved for testing. Because the testing set is unknown to the supervised learning algorithm, it serves as the benchmark of the possible performance on real world problems. For more accurate evaluation, the split-training-testing procedure can be carried out multiple times, and the average of the testing performance is used as the evaluation score of the supervised learning algorithm.

2.2 Audio vs. Symbolic

The research of music information retrieval can be generally divided into two subordinate fields: audio music information retrieval and symbolic music information retrieval, by the nature of different types of data concerned. Symbolic music files contains the symbolic representation of songs. For example, the Musical Instrument Digital Interface format (MIDI, .mid) records information such as the note onset time, note pitch, musical effects, instrumentation, etc. It is entirely possible to recover the full score of the song from a well-recorded MIDI file. Similarly, MusicXML is a XML-based music notation file format that stores the actual score of songs. It is the common standard designed for score exchange between different types of scorewriter software. There are also other symbolic music formats used by various musical composition software. Playing a symbolic music file requires a synthesizer that translate the musical notations to actual sounds. The instrumentation library and the capacity of the synthesizer can drastically affect the quality of music generated, given the identical symbolic music file.

On the contrary, audio music files contains the pulse-code modulated digital signals of songs [1]. Basically, the actual sound wave signals or their compressed form are stored in audio music format. Example file formats includes the Waveform Audio File Format (.wav), MPEG-1 Audio Layer 3 format (.mp3) and Free Lossless Audio Codec format (.flac). Playing a audio music file requires a Digital-to-Analog (DA) converter that transform the digitized signals to audible analog sounds. The compressed audio file formats may require an additional decoder layer before the DA converter. The same audio music file should sound very similarly on different machines, even if they are using different types of DA converters.

Based on the characteristics of data, the feature extraction methodology used for symbolic music information retrieval is very different from its audio counterpart. In modern classification frameworks, feature extraction is a critical process layer between the raw data and the classifier. Feature extraction transforms the complex, elusive raw data to a compact set of informative attributes (or the feature vector) that is suitable to utilized as the input of classifiers. It can be considered as a special form of dimensionality reduction. The effectiveness of feature extraction is critical to the later process as it will greatly affect the overall performance. Take genre classification for instance.

[1] In this paper, only digital audio music is concerned. Analog music on cassettes and gramophone records is not considered.

Because the high-level musical representations such as note onsets, pitches and instrumentation are readily available in the files, the feature extraction process for symbolic music genre classification is straight-forward and musicology relevant. The vast set of music theory and other musicology knowledge are directly applicable to the entire feature extraction process. As a result, it would be easier to achieve satisfactory classification accuracy than using only audio features. Following is a list of example symbolic music genre classification systems.

- Tzanetakis et. al. [42] presented his five-genre classification systems using pitch statistics as feature vector and k-nearest-neighbor (KNN) as the classifier. The Pitch Histogram he extracted is basically a 128-dimensional vector indexed by MIDI note numbers. It shows the frequency of occurrence of each note in a musical piece. From the Pitch Histogram he further computes a 4-dimensional feature set that summarizes the major characteristics of the Pitch Histogram. The experiments are carried out on three different types of datasets: purely MIDI data, audio files converted from MIDI data and general audio files. It is shown that, in his experiments using only pitch histogram features, the classification accuracy for purely MIDI data is significantly better than the audio-from-MIDI dataset and the general audio dataset. The experiments well demonstrated the advantage to extract reliable pitch information from symbolic music files over audio music files.

- Basili et. al. [3] presented his classification system on a six-genre MIDI dataset. Various types of feature sets such as melodic intervals, instrumentation, meter/time changes and note extension are extracted to facilitate the classification using six different types of classification algorithms. Investigation of the impact of different musical features on the inductive accuracy is also carried out. They achieved about 60% for multi-class classification accuracy.

- Ponce et. al. [34] adopts the self-organising neural maps (SOM) as their classification model. The features extracted include pitch descriptors, note duration descriptors, silence duration descriptors, etc. They showed a smaller SOM map would produce better overall performance, as their system scored 76.9% and 77.5% in average accuracy for jazz melodies and classic melodies respectively. They further improved their work in [11] where they introduced a feature selection process. Experiments were refined to obtain better results. The average accuracy for jazz melodies and classic melodies classification were boosted to 81.8% and 89.3%.

- McKay et. al. [27] achieved very high accuracy using a hierarchical classification system. They extract 109 features which can be divided into seven categories: instrumentation, musical texture, rhythm, dynamics, pitch statistics, melody and chords. Two classification models, i.e. feed-forward neural networks (NN) and the k-nearest-neighbor (KNN), are used in their system. They also apply the genetic algorithm to the feature selection process to further boost up the classification accuracy. The MIDI dataset they use includes 950 recordings. Categories are distributed in three main genres and further in nine subordinate leave genres. The experiments show that the hierarchical classification scheme scores better than the flat classification scheme as they achieved 90% and 86% for leave genre classification respectively.

On the other hand, feature extraction for audio music information retrieval is more difficult and less musicology relevant. Classifying audio music in the way of symbolic music is hardly possible because of the hardship transforming the audio signals into its

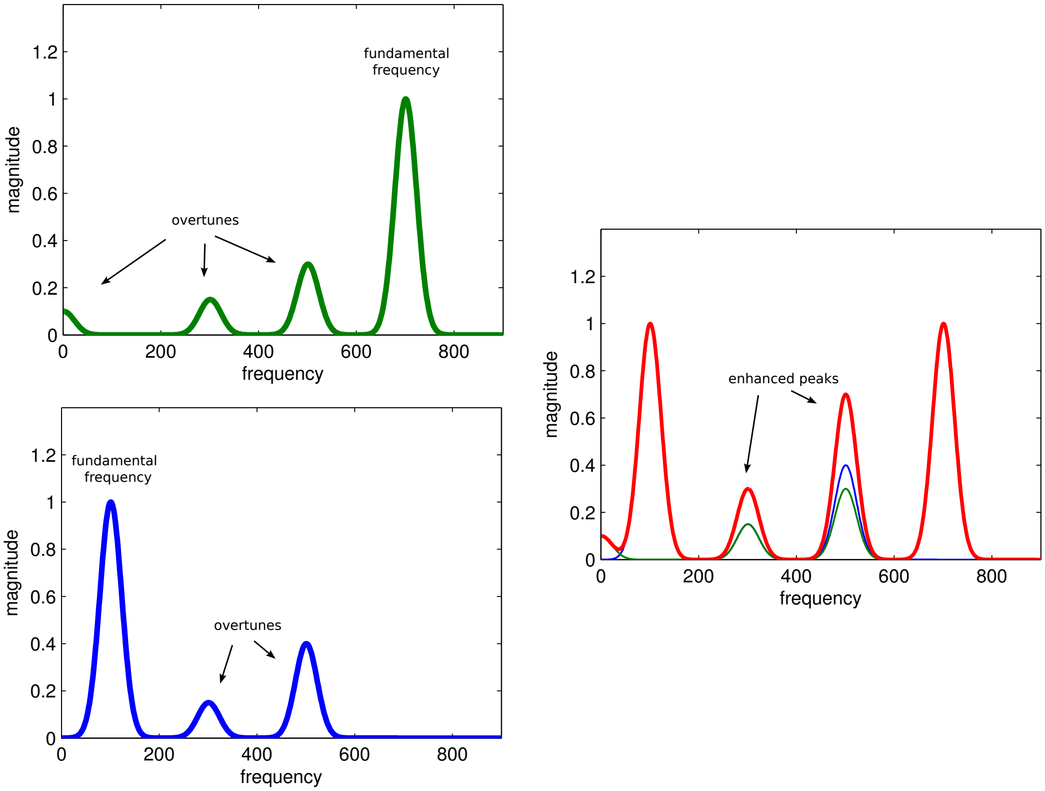

Figure 2.1: The demonstration of audio masking effect.

original score form. Take the extraction of pitch for example, a sound of an musical instrument can be musicologically viewed as composition of a fundamental frequency that determines the pitch, and the overtunes that determines the timbre. It is an easy task to extract pitch and the corresponding instrument in mono-instrument audio signals. But the situation gets very complicated in poly-instrument transcription in which the overtunes of different instruments overlap each other, making the fundamental frequencies not apparent. As we can see in Figure 2.1, the two graphs on the left represent the spectrogram characteristics of two instruments, their fundamental frequencies and overtunes indicated as marked. The graph on the right is the effect combining the sound of two instruments together. We can observe that some overlapping overtunes are enhanced substantially to the extent of approximately the level of fundamental frequencies. The more instrument involved, the more serious such masking effect could be. Such spectrogram masking effect plays an major obstacle in poly-instrument pitch extraction. Similarly, the note onset detection and the instrument extraction turn out to be a serious problem in audio context. At the current state of the art, transforming audio music into its symbolic form is still an unsolved problem under active research. Trying to apply methodologies in symbolic music analysis on auto-transcribed audio data is highly impractical since building up a reliable auto-transcription system for audio music appears to be a more challenging task than audio genre classification itself. In fact, the best candidate scored only about 70% in the 2009 MIREX melody extraction contest [2] , a simpler task than auto-transcription.

Considering the unavailability of reliable symbolic information, researchers seek help from related research fields such as speech recognition for reliable feature extractors. Short-time Fourier transform (STFT) and mel-frequency cepstral coefficients (MFCC) are two feature sets which have been typically widely adopted in audio genre classification systems. The experiments in this thesis also rely heavily on the MFCC feature set. Before listing the example audio music genre classification systems and their feature sets, I would like to go through some details of these two feature sets.

2.3 STFT and MFCC

The Human Ear

Many techniques for processing audio sounds originate from analyzing the auditory perception of human beings. For instance, the standard audio CD sampling rate is 44.1 kHz. The selection of this sampling rate is primarily based on the human audible frequency range, from 20 Hz to 20 kHz. According to the Nyquist-Shannon sampling theorem, ”a sampling rate of more than double the maximum frequency of the signal to be recorded is needed”. And therefore the sampling rate 44.1 kHz just well covers the the full human audible frequency range. Similarly, the extraction of STFT and MFCC feature are largely based on the functionality of human ear.

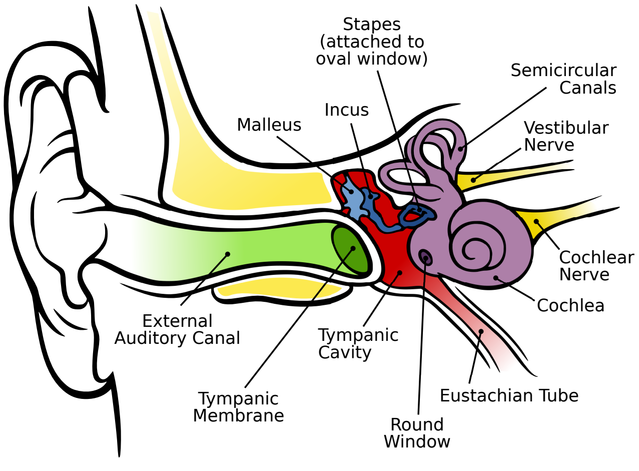

Figure 2.2: The anatomy of human ear.

Figure 2.2 [9] shows the anatomy of human ear. The sound we perceive is actually a form of energy that moves through a kind of medium that passes the energy from the source to our ears. The human ear can be divided into three parts: outer, middle and inner. The outer part of human ear include the visible pinna, the external auditory canal and the tympanic membrane (or the ear drum) that separate the outer ear and middle ear. The middle ear is air-filled cavity immediately behind the tympanic membrane. It contains three smallest bones in human body that connect the the tympanic membrane to the inner ear. The inner ear contains both organs for hearing (the cochlear) and balance control of the body (three semicircular canals). The rear of the inner ear (if we conveniently define the part adjacent to the middle is the ”front”) is attached to two fibers of nerve which transmit signals collected in the ear to the brain for further process.

When the sound wave arrive at our ears, it is collected by the external pinna and transfered to the tympanic membrane via the external auditory canal. The sound wave is then transformed to the vibration of the tympanic membrane. Such vibration is enhanced and transferred to the entry of the inner ear by the three small ear bones. The last ear bone, the stapes, is attached to an oval window of the cochlear. The movements of the ear bones cause pushes on the oval window, resulting in the movement of fluid within the cochlear. When the sound energy arrive in the cochlear in the form of cochlea fluid movement, it is picked up by the receptor cells which fire signals back to the brain.

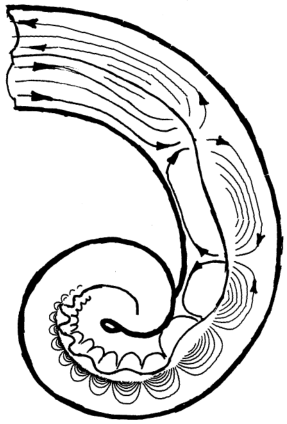

Figure 2.3: The illustration of the basilar membrane.

But what kind of signals is transmitted? Are the signals structured based on different frequencies? Or the signals record the actual form of sound wave? Such question can be answered from two different perspectives. First, the study of the inner structures of the cochlea reveals that the perception of the frequency-dispersed sound of human beings results from the functionality of a stiff structural membrane that runs along the coil of the cochlea, the basilar membrane [4] . When the sound energy comes into the cochlea, different frequency components of it drive different sections of the basilar membrane to vibrate. The vibration of the basilar membrane triggered the associated auditory receptor hair cells to fire neural signals. And therefore, different auditory cells give response for different frequency components of the incoming sound. The cochlea acts more or less like a mechanical frequency analyzer that decomposes the complex acoustical waveform signals into simpler frequency components. Such information is then shipped via nerve fibers to the auditory cortex in the brain. Another answer to the question is obtained from the study of cochlea implants. The cochlea implant is a kind of electronic device that provides the sense of sound to a severely auditory-impaired person. It functions as it capture the environmental sounds and transform the signals to electrical stimulation directly on the auditory nerve fiber cells. Researches on the electrical activity in inferior colliculus cells of cats [29] proved that the electrical nerve signals are organized by frequency bands. Based on such a finding, scientists built up a multi-channel cochlea implant that encodes environmental sounds in electrical stimulus on multiple frequency bands, and later on multi-channel cochlea implants turned out to be a great success. Experiments on a congenitally deaf patient [29] showed that, the multi-channel implant enable the profoundly deaf patient to capture the melody and the tempo of the song ”Where have all the Flowers Gone”. Nowadays multi-channel cochlea implants are widely adopted.

To sum up, the human ear transforms the incoming sound wave into frequency-dispersed nerve signals before the process of brain. Therefore it is biologically intuitive to analyze the sound wave signals by first converting it to the frequency domain, as it mimics the functionality of the human ear.

Short-Time Fourier Transform

Fourier analysis is a set of mathematical techniques which are used to decompose signals into sinusoid waves. The Fourier transform basically converts a time series signal to its frequency domain. When it comes to sounds analysis, it reveals the frequency information inside the sound signals. In the research of sound/music feature extraction, a special form of Fourier transform, the discrete short-time Fourier transform (STFT) is used. This is because audio digital music are discrete signals, and analysis of frequency only makes sense when a short-time window is concerned; sound signals such as speech and music are generally very changeable over time. The following formula shows the calculation of STFT.

$$\textbf{STFT}\left\{x\left[n\right]\right\} \equiv X\left(m,\omega\right) = \sum_{n=-\infty}^{\infty} x\left[n\right] w\left[n-m\right] e^{-j\omega n} \tag{2.1}$$

In the equation above, $x[n]$ represents the the input signal and $w[n]$ represents the window function. In typical applications, the STFT is calculated on a computer using the Fast Fourier Transform (FFT) algorithm since it is significantly faster than the formula listed above while the accuracy is well preserved.

Figure 2.4 shows the generic process of STFT extraction. The original audio signal

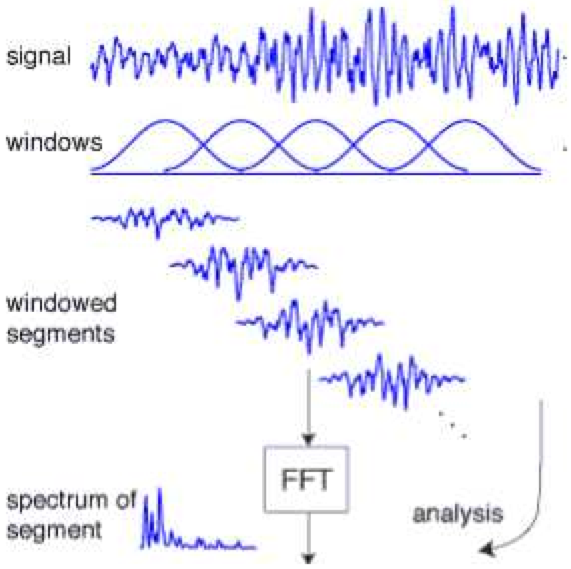

Figure 2.4: The short time Fourier Transform process.

first convolve with a certain type of window function. In this thesis, the window function used is Hamming window. The windowed signals are transformed using the equation listed above. Usually this stage is replaced with a faster algorithm: Fast Fourier Transform. The result of the transform is STFT values.

After the STFT process, the sound signals are transformed into frames of spectrograms which span typically about 20 milliseconds. For audio music genre classifications, additional process steps are often adopted to further condense a frame spectrogram to compact feature sets. Following is a incomplete list of such feature sets [41]

.

- **Spectral Centroid** : The spectral centroid is defined as the ”gravitational center” of a STFT frame spectrogram. It is calculated as

$$C_t = \frac{\sum_{n=1}^{N} M_t\left[n\right] \times n}{\sum_{n=1}^{N} M_t\left[n\right]} \tag{2.2}$$

where $M_t[n]$ represents the magnitude of STFT spectrogram at frame t and frequency bin n. The spectral centroid is a measurement of the spectrogram shape. The larger the value, the more energy in the high frequency bands.

- **Spectral Rolloff** : The spectral rolloff is defined as the frequency R_t below which 85% of spectrogram magnitude is concentrated. It also measures the spectrogram shape.

$$\sum_{n=1}^{R_t} M_t\left[n\right] = 0.85 \times \sum_{n=1}^{N} M_t\left[n\right] \tag{2.3}$$

- **Spectral Flux** : The spectral flux is defined as the squared difference between the normalized magnitudes of two successive STFT spectrogram. It measures the local spectral change amount between two adjacent frames.

$$F_t = \sum_{n=1}^{N} \left(N_t\left[n\right] - N_{t-1}\left[n\right]\right) \tag{2.4}$$

  where $N_t[n]$ and $N_{t-1}[n]$ stand for the magnitude of spectrogram at frequency bin n for frame t and $t-1$ respectively.

- **MFCC** : As described in the following subsection.

Mel-Frequency Cepstral Coefficients

The mel-frequency cepstral coefficients (MFCC) is a compact, short-duration audio feature set extracted based on the STFT spectrogram. It was proposed over thirty years ago [7], and since then it has been widely adopted for various audio processing tasks such as speech recognition [33], environmental sound recognition [25] and musical information retrieval tasks. MFCC and its derivatives have also been used extensively in many audio genre classification systems [6, 15, 28, 41].

The calculation of MFCC include the following four steps[2] .

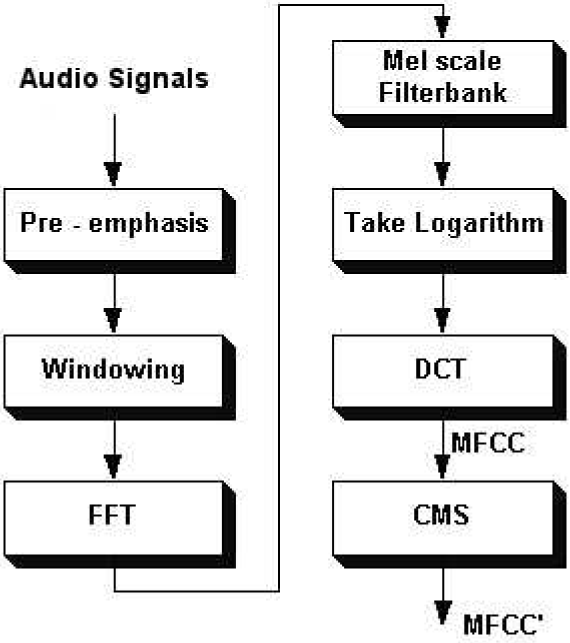

Figure 2.5: The MFCC extraction procedure.

1. Transform the audio signals to frames of spectrogram using STFT (The Pre-emphasis, Windowing, and FFT steps in Figure 2.5).

2. Map frequency bins of these spectrogram to mel-scale. The values of the frequency bins are aggregated into the so-called mel bands using triangular overlapping windows.

3. Take the logs of the value of the mel bands.

[2] The actual parameters such as window number, window shape, etc may vary in applications.

4. Apply a set of discrete cosine transform (DCT) filters on the mel bands as if they were signals. The result is the cepstral coefficients.

5. There is an optional cepstral mean subtraction (CMS) step after the DCT transform. [31] shows thats such a step is performed for noise cancellation. In this thesis, the MFCC values are extracted without such a step.

As we can observe from the list above, MFCC feature set takes several further steps to compress the STFT spectrogram features, reducing the dimensionality from typically several hundreds to below twenty. Behind the magic of these computationally simple steps are the findings of the nature of human auditory perception. The mel scale was originally proposed by Stevens, Volkman and Newman [39] in 1937 as they found out that the linear increase of the perceptive ”pitch distance” would result in exponential increase in the actual frequency hertz. The formula to convert f hertz to m mel is give below.

$$m = 2595 \mathrm{log}_{10}\left(\frac{f}{700}+1\right) = 1127 \mathrm{log}_{e}\left(\frac{f}{700}+1\right) \tag{2.5}$$

In the sense of musicology, it explains the relationship between the musical pitches and their actual frequencies. For example, the pitch of the sound A4 (or Concert A, Middle A) stands for a frequency of 440 Hz [18] . The pitch an octave above A4, the A5, stands for a frequency of 880 Hz, which is double that of A4. The pitch two octaves above A4, the A6, has double the frequency of A5, that is 1760 Hz, instead of the triple of A4’s frequency 1320 Hz. The third step actually transforms the magnitude of the mel bands to the decibel scale. The transform is also based on the human perception of sound intensity. The last step of processing decomposes the mel bands to a set of

DCT coefficients. Research [24] show that, the DCT decomposition has similar effect as the KL transform that decorrelates mel bands components, but it is computationally more efficient. The incorporation of knowledge of human auditory system as well as mathematical techniques makes MFCC very successful in the field of audio information retrieval.

2.4 Genre Classification Systems and Feature Sets

The research of audio music genre classification probably started at late 90s. In the last decade, various classification systems and different kinds of feature sets are proposed to solve the problem. Following is an list of the example systems the feature sets they used.

1. Tzanetakis et. al. [41] proposed his audio music classification system based on the feature sets describe three different aspects of music: timbre, beat and pitch. The derivatives of STFT and MFCC are used as timbral feature sets , while the Pitch Histogram and the Beat Histogram are deviced to capture the pitch and beat characteristics of songs. Experiments are carried out on a 1000-song, 10 genre GTZAN dataset [3] , using classification models such as the k-nearest-neighbor (KNN) algorithm and the Gaussian mixture model (GMM). They achieved 61% classification accuracy on the dataset. Their comparison among the feature sets also revealed that the two timbral feature sets performed significantly better than the pitch and beat feature sets. The experiments were continued in [21] using

[3] This dataset is very widely used and tested with various systems. It can be considered as a sort of benchmark standard. The experiments in later chapters of this thesis are also based on this dataset.

support vector machine (SVM) and the Linear Discriminant Analysis (LDA). The performance was pushed to 71.1% using the full feature set and LDA. The comparison among the feature sets showed similar result as the previous paper.

2. Xu et. al. [44] proposed an audio music classification system using SVM as the classifier. Their feature set includes linear predictive coding (LPC) derived cepstrum, zero crossing rate, spectrum power, MFCC and the Beat Spectrum feature set deviced to capture the beat characteristics of songs. The experiments was carried out on a 100-song, 4 genre dataset. The performance for SVM are compared with other statistical learning model.

3. Meng et. al. [28] carried out their experiments on three different scales of audio features: short-duration, medium-duration and long-duration, for the task of audio music genre classification. The short-duration feature is MFCC with its first six coefficients. The medium-duration features include the various statistical summary of MFCC and derivatives of the zero-crossing rate feature. The long-duration features include the statistics of the medium feature and two beat-related feature sets proposed by other researchers [41, 16] Their experiments show that the long- and medium-duration feature sets derive from MFCCs are most effective in music genre classification. The investigated classifiers include Linear Neural Network and Gaussian classifiers.

4. Lidy et. al. [22] proposed their feature set using psycho-acoustic transforms to construct effective audio feature extractors. The feature sets include the Rhythm patterns, Statistical Spectrum Descriptors and Rhythm Histogram, the functionality of them indicated as their names. Their experiment are carried out on a great variety of datasets, including the GTZAN dataset and datasets used in the

2004 ISMIR contest. Different combination of psycho-acoustic transforms and classification models were evaluated. Their feature sets achieved very remarkable performance, scoring 74.9% classification accuracy on the GTZAN dataset. In their later paper [23] , they incorporated the information extracted by an automatic transcription system to their existing classification model. Although the result of auto-transcription system is far from perfectly reliable, the resulting score still contained sufficient amount of genre-related information to improve the final classification accuracy, scoring 76.8% on the GTZAN dataset.

The list above is by no means the complete list of all systems and feature sets. Apart from the feature sets that is proposed from the perspective of sound and music processing, researchers also tried to attack the problem from some ”alternative” angles. Soltau et. al. [37] tries to train the neural network and use its middle layer as the feature extractor. Similarly, Sundaram et. al. [40] build up their feature extractors by training with some generic sound effect libraries. The feature extracted, the Audio Activity Rate, is further used in the context of music genre classification. Deshpande et. al. [13] perceive the music genre classification problem in the image way. They applied a image information technique, the texture-of-texture approach, to extract meaningful information from MFCC and STFT spectrograms. The three systems above inspired me of seeking alternative approaches to attack the audio genre classification, especially when the performance of traditional ways meet their bottleneck. The detailed attempts will be covered in the following chapters.

Chapter 3

Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network

3.1 Introduction

Automatic audio music genre classification is a promising yet difficult task, as much of the difficulty originates from the modelling of elusive music features. A first step of genre classification, feature extraction from musical data will significantly influence the final classification accuracy. Most of the modern audio music genre classification systems rely heavily on timbral, statistical spectral features. Feature sets pertaining to other musicological aspects such as rhythm and pitch are also proposed, but their performance is far less reliable compared with the timbral feature sets. Additionally, there are few feature sets aiming at the variations of musical patterns. The inadequateness of musical descriptors will certainly impose a constraint on audio music genre classification systems.

In this chapter we propose a novel approach to automatically retrieve musical pattern features from audio music using convolutional neural network (CNN), a model that is adopted in image information retrieval tasks. Migrating technologies from another research field brings new opportunities to break through the current bottleneck of music genre classification. The proposed musical pattern feature extractor has advantages in several aspects. It requires minimal prior knowledge to build up. Once obtained, the process of feature extraction is highly efficient. These two advantages guarantee the scalability of our feature extractors. Moreover, our musical pattern features are complementary to other main-stream feature sets used in other classification systems. Our experiments show that musical data have very similar characteristics to image data so that the variation of musical patterns can be captured using CNN. We also show that the musical pattern features are informative for genre classification tasks.

3.2 Methodology

The previous chapter has presented some example audio music genre classification systems. As we observe, most of the proposed systems concentrate only on feature sets extracted from a short window of audio signals, using statistical measurements such as maximum value, average, deviation, etc. Such features are representative of the ”musical texture” of the excerpt concerned, i.e. timbral description. Feature sets concerning other musicological aspects such as rhythm and pitch are also proposed, but their performance is usually far worse than their timbral counterparts. There are few feature sets which capture the musical variation patterns. Relying only on timbral descriptors would certainly limit the performance of genre classification systems; Aucouturier et. al. [32] indicates that a performance bottleneck exists if only timbral feature sets are used.

The dearth of musical pattern features can be ascribed to the elusive characteristics of musical data; it is typically difficult to hand-craft musical pattern knowledge into feature extractors, as they require extra efforts to hand-craft specific knowledge into their computation processes, which would limit their scalability. To overcome this problem, we propose a novel approach to automatically obtain musical pattern extractors through supervised learning, migrating a widely adopted technology in image information retrieval. We believe that introducing technology in another field brings new opportunities to break through the current bottleneck of audio genre classification.

In this section, we briefly review the CNN and the proposed music genre classification system.

3.2.1 Convolutional Neural Network

Neural networks is a mathematical model inspired by real neural system in animals. The actual structure of the network varies based on the way of connection, the distribution of weights and the training strategies. Arguably, the most commonly used type of neural network is the 3-layer feed-forward neural network which is applied as a generic non-linear classifier. The feed-forward neural network is advantageous in the simpleness of implementation and the classification speed. Such architecture is also very suitable for hardware implementation, which makes the classification even faster.

The design of convolutional neural network (CNN) has its origin in the study of visual neural system. The specific method of connections discovered in cats' visual neurons is responsible for identifying the variations in the topological structure of objects seen [30]. LeCun incorporate such knowledge in his design of CNN [5] so that its first few layers serve as feature extractors that would be automatically acquired via supervised training. It is shown from extensive experiments [5] that CNN has considerable capacity to capture the topological information in visual objects.

There are few applications of CNN in audio analysis despite its successes in vision research. Neural science research [35] shows that ”the early cortical processes and their implementation are similar across sensory modalities” as ”striking similarities of receptive field organization are found in visual, auditory and somatosensory areas”. The CNN model achieves the state-of-the-art performance in handwritten digit recognition tasks based on its structure derived from real visual neural system. Therefore it is reasonable to extend its usage to audio tasks since its structure also reflects the ”receptive fields” connections found in real auditory neural system. The core objective of this paper is to examine and evaluate the possibilities extending the application of CNN to music information retrieval. The evaluation can be further decomposed into the following hypotheses:

- The variations of musical patterns (after a certain form of transform, such as FFT, MFCC) is similar to those in images and therefore can be extracted with CNN.
- The musical pattern descriptors extracted with CNN are informative for distinguishing musical genres.

In the latter part of this chapter, evidence supporting these two hypotheses will be provided.

3.2.2 CNN Architecture for Audio

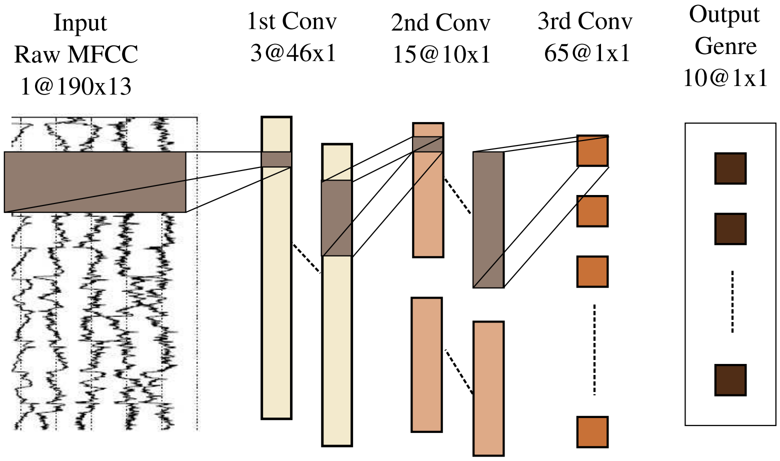

Figure 3.1: CNN to extract musical patterns in MFCC

Figure 3.1 shows the architecture of our CNN model. There are five layers in total, including the input and output layers. The first layer is a 190×13 map, which hosts the 13 MFCCs from 190 adjacent frames of one excerpt. The second layer is a convolutional layer of 3 different kernels of equal size. During convolution, the kernel surveys a fixed 10×13 region in the previous layer, multiplying the input value with its associate weight in the kernel, adding the kernel bias and passing the squashing function. The result is saved and used as the input to the next convolutional layer. After each convolution, the kernel hops 4 steps forward along the input as a process of subsampling. The 3rd and 4th layer function very similarly to the 2nd layer, with 15 and 65 feature maps respectively. Their kernel size is 10×1 and their hop size is 4. Each kernel of a convolutional layer has connections with all the feature maps in the previous layer. The last layer is an output layer with full connections with the 4th layer. The architecture of this model is designed based on the original CNN model used for digit recognition. Image data are 2-D in nature, and therefore the image CNN convolves in two directions on the input image signal, capturing the topological features while ignoring the slight spacial variance. When it comes to audio features, the slight variance we need to cancel is the variance in time. Since adjacent MFCC coefficients do not correlate with each other like the nearby pixels on images, it is not appropriate to apply coefficient-wise convolution on the MFCC maps. All the MFCC coefficients are aggregated in the first layer, turning the 2-D input into 1-D. The later layers operate on 1-D inputs ever since. The parameter selection process is described in Section 3.3.2.

It can be observed from the topology of CNN that the model is a multi-layer neural network with special constraints on the connections in the convolutional layers, so that each artificial neuron only concentrates on a small region of input, just like the receptive field of one biological neuron. Because the kernel is shared across one feature map, it becomes a pattern detector that would acquire high activation when a certain pattern is shown in the input. In our experimental setting, each MFCC frame spans 23ms on the audio signal with 50% overlap with the adjacent frames. Therefore the first convolutional layer (2nd layer) detects basic musical patterns appear in 127ms. Subsequent convolutional layers therefore capture musical patterns in windows size of 541ms and 2.2s, respectively. The CNN is trained using the stochastic gradient descent algorithm [38] for simplicity. The brief description of the algorithm is given below:

For a certain neural network model M, let $E(x_i, w)$ be the error function of the neural network given a training sample vector x_i, and the weight matrix w. The new weight matrices w is updated by

$$w_{new} := w - \alpha \nabla E\left(w, x_i\right) \tag{3.1}$$

as the algorithm sweep through the data set. α in the equation is the learning factor that determines the step size of the gradient descent process, while error function $E(x_i, w)$ is usually defined as the mean squared error between the label of the input and the actual output of the network. The process is repeated multiple times so that the error function converges to a local minimum.

It could be better to implement some heuristic search algorithm such as the genetic algorithm, simulated annealing algorithm, etc, which might be less prone to get trapped in local minima, but that should significantly add to the overall system training time. After convergence, the values in the intermediate convolutional layers can be exported as the features of the corresponding musical excerpt.

The model we use is a modified CNN model presented in [36]. Compared with the traditonal CNN model, we observed that the training is easier, and the capacity loss is negligible. In return, as much as **66.8%** of computational requirement is saved.

3.2.3 Music Genre Classification

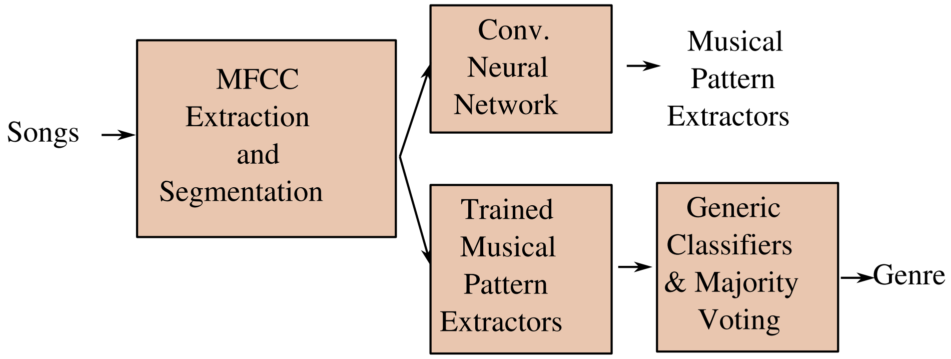
Songs
MFCC Extraction and Segmentation
Conv. Neural Network
Musical Pattern Extractors
Trained Musical Pattern Extractors
Generic Classifiers & Majority Voting
Genre

Figure 3.2: Overview of the classification system

Figure 3.2 shows the overview of our classification system. The first step of the process is MFCC extraction from audio signals. MFCC is an efficient and highly informative feature set that has been widely adopted for audio analysis since its proposal. After MFCC extraction, the input song is transformed into an MFCC map with 13 pixels wide which is then segmented to fit the input size of CNN. Provided the song label, the musical pattern extractors are automatically aquired via supervised learning. Those extractors are used to retrieve high-order, pattern-related features which will later serve as the input of generic, multi-class classifiers such as Decision Tree Classifiers, Support Vector Machine etc. After classification of each song segments, the result is aggregated in a majority voting process to produce the song-level label.

3.3 Results and Analysis

3.3.1 Dataset

The dataset of our experiment is the GTZAN dataset which has been used to evaluate various genre classification systems [41, 6, 23]. It contains 1000 song excerpts of 30 seconds, sampling rate 22050 Hz at 16 bit. Its songs are distributed evenly into 10 different genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae and Rock.

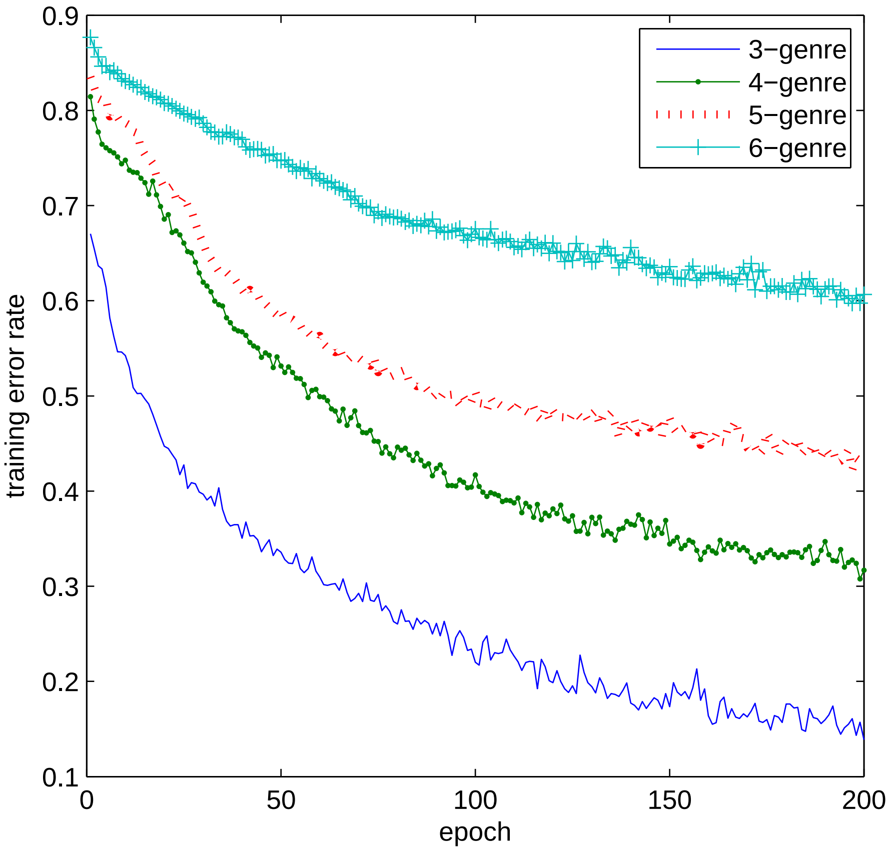

Figure 3.3: Convergence Curve in 200-epoch training

3.3.2 CNN Pattern Extractor

Figure 3.3 shows the convergence of the training error rate of our CNN model, on four sub-datasets extracted from the GTZAN dataset. The smallest dataset contains 3 genres: Classical, Jazz and Rock. The latter datasets increase in size as Disco, Pop and Blues genres are added. From the figure we can observe that the trend of convergence over different datasets is similar, however the training on a 3-genre dataset converges much faster than the training on a 6-genre dataset. This shows the difficulty in training CNN increases drastically when the number of genres involved in training increases. We believe this is because the CNN gets confused with the complexity of the training data and therefore never obtains suitable pattern extractors in the first few layers. Additionally we also found that the combination of genres in the 3-genre subset will not affect the training of CNN. All combinations have very similar curve of convergence.

Based on the observations above, the training of our CNN feature extractors are divided in four parallel models to cover the full 10-genre GTZAN dataset. Three models are arbitrarily selected to cover 9 non-overlapping genres, while one model is deliberately chosen to train on the 3 most difficult-to-classify genres shown in [41], i.e. Blues, Metal and Rock. Dividing the dataset into small subsets to train the CNN feature extractors may have the side-effect that features extracted to classify songs within one subset may not be effective in inter-subset classification, and therefore it may seem more reasonable to select three 4-genre models instead of four 3-genre models. We observe from our experiments that such alternative is unnecessary since features extracted from individual subsets possess a good capacity for inter-subset distinction. Additionally, we also observe that the training of 4-genre subsets is far less effective and less efficient compared with training of 3-genre subsets.

Extensive experiments are also performed towards the selection of CNN network parameters. First is the network layer number. We discover that CNN with more than 3 convolutional layers is exceptionally difficult to train for the network convergence will easily get trapped in local minima. On the other hand, CNNs with less than 3 convolutional layers do not have sufficient capacity for music classification. The convolution/subsampling size is set at 10/4 for similar criteria. Larger convolutional sizes are difficult to train, while smaller ones are subjected to capacity limitation. To determine the feature map numbers in the three convolutional layers, we first set the three parameters sufficiently large, then watch the performance of CNN as we gradually reduce the number. We discover that 3, 15 and 65 is the optimal feature map numbers for the first three convolutional layers. Reducing them further will drastically constrain the capacity of CNN feature extractors.

3.3.3 Evaluation

After obtaining 4 CNNs as described above, we apply the feature extractors on the full dataset to retrieve musical pattern features. We deliberately reserve 20% songs in the training of CNN as to examine the ability of our feature extractors on unseen musical data. The musical pattern features are evaluated using various models in the WEKA machine learning system [17]. We discover that the features scored very well in the 10-genre training evaluation, using a variety of tree classifiers such as J48, Attribute Selected Classifier, etc. The classification accuracy is 84% before the majority voting, and gets even higher afterwards. Additionally, musical excerpts not used in CNN training have minor difference in classification rate compared with excerpts used to train CNNs. This provides evidence to support our hypothesis in Section 3.2 that the variations of musical patterns in the form of MFCC is similar to those of image so that CNN can be used to automatically extract them. In addition, those patterns provide useful information to distinguish musical genres.

However, further experiments on the split test dataset give very poor performance compared with the training evaluation; the accuracy of below 30% is therefore too low to make any reliable judgments. Such result is consistent for different splits of the dataset into training and testing data. It reveals that our current musical pattern extraction model has the deficiency in generalizing the musical patterns learnt to unseen musical data. We further study such phenomenon and found that the reason is two-fold: 1. Musical data is typically abundant in its variation, and therefore it is hardly sufficient for 80 songs to represent all types of variations in one specific genre; 2. The MFCC feature is sensitive to the timbral, temple and key variation of music which further accentuates the shortage in training data.

One practical solution to these problems above is to enlarge the training dataset by adding affine transforms of songs, such as key elevation/lowering, slight tempo shift, etc. Additional data smooths the variation within one genre and boosts the overall generalizability. Similar work can be found in [36]. Alternatively, the MFCC feature input can be replaced with transforms insensitive to timbral, tempo and key variation, such as mel-frequency spectrum or chroma feature [15].

The attempt to study the filters learned in the first few layers also prove very difficult. That is partially due to the difficulties understanding the MFCC inputs. Unlike the topological connections in images which can be easily observed, the information hidden in the MFCC values is hard to tell. After the DCT transform, the MFCC coefficients becomes irregular and random-like. The pre-processing normalization and the first few layers of CNN, patterns extracted from the original MFCC coefficients get more elusive, rendering any meaningful analysis hardly possible. Future works could be directed at learning on different features that contain more inter-feature correlation, e.g. STFT or mel-scale filterbanks.

Our method on musical pattern extractor can be compared with the work in [13], which also applies an image model to audio music genre classification. It is shown that our system possesses better scalability. The texture-of-texture model used in [13] is so highly computational intensive that the authors reduce the training set to 17 songs each category. In comparison our CNN takes less than two hours to obtain feature extractors from a 3-genre, 240-song training set. The efficiency of process can be raised further with parallel computing on different combination of genres.

3.4 Conclusion

In this chapter we presented a methodology to automatically extract musical patterns features from audio music. Using the CNN migrated from the the image information retrieval field, our feature extractors need minimal prior knowledge to construct. Our experiments show that CNN is a viable alternative for automatic feature extraction. Such discovery lends support to our hypothesis that the intrinsic characteristics in the variation of musical data are similar to those of image data. Our CNN model is highly scalable. We also presented our discovery of the optimal parameter set and best practice using CNN on audio music genre classification.

Our experiments reveal that our current model is not robust enough to generalized the training result to unseen musical data. This can be overcome with an enlarged dataset. Furthermore, replacing the MFCCs with other feature sets such as the Chroma feature set would also improve the robustness of our model. Further application of image techniques are likely to produce fruitful results towards music classification.

3.5 Acknowledgement

The text of this chapter, in full, is based on the material as it appears in [20] . The dissertation author was a primary researcher and first author of the cited material.

Chapter 4

Genre Classification and the Invariance of MFCC Features to Key and Tempo

4.1 Introduction

In many genre classification systems, the Mel-frequency cepstral coefficients (MFCCs) [7] have been used as a timbral descriptor [41, 32, 26, 15]. While it is common to think of MFCCs as timbre-related features, due to the short-duration frame on which they are extracted (e.g., 20 milliseconds), it is still uncertain how the key and tempo of a song affects the MFCC features, and hence the subsequent genre classification system.

In this chapter, we attempt to address the following question: are MFCCs invariant to key and tempo? In other words, is MFCC a purely timbral feature set? If the MFCCs are purely timbral features, then they should be invariant to the changes in musical keys and tempo. Otherwise, changes in the musical key and tempo of a song will affect the

MFCCs, which may adversely affect the training of genre classifiers. The contributions of this chapter are three-fold. First, we show that musical genres, which *should* be independent of key, are in fact influenced by the fundamental keys of the instruments involved. Second, we show that MFCCs indeed encode both timbral and key information, i.e., they are not invariant to shifts in musical key. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. Third, we propose an approach to build key-independent genre classifiers, which consists of augmenting the classifier training and prediction phases with various key and tempo transformations of the songs. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

The rest of this chapter is organized as follows. In Section 4.2, we explore the distribution of musical key for different genres. In Section 4.3, we study the invariance of MFCC to musical key and tempo shifts. In Section 4.4, we propose a data-augmented genre classification scheme, based on key and tempo transformations, while in Section 4.5 we present experiments on genre classification using our data-augmented system.

4.2 Key Histograms of the GTZAN dataset

In this section, we explore the relationship between musical genres and musical keys. We manually annotate each song in the GTZAN dataset [41] with their musical “keys”. In this section, we define the concept of “key” as the pitch of the “Do” sound of the song in the solfège scale (Do-Re-Mi scale). Such definition is different from the more common definition — the tonic sound of the scale (e.g., in minor scales the tonic sound

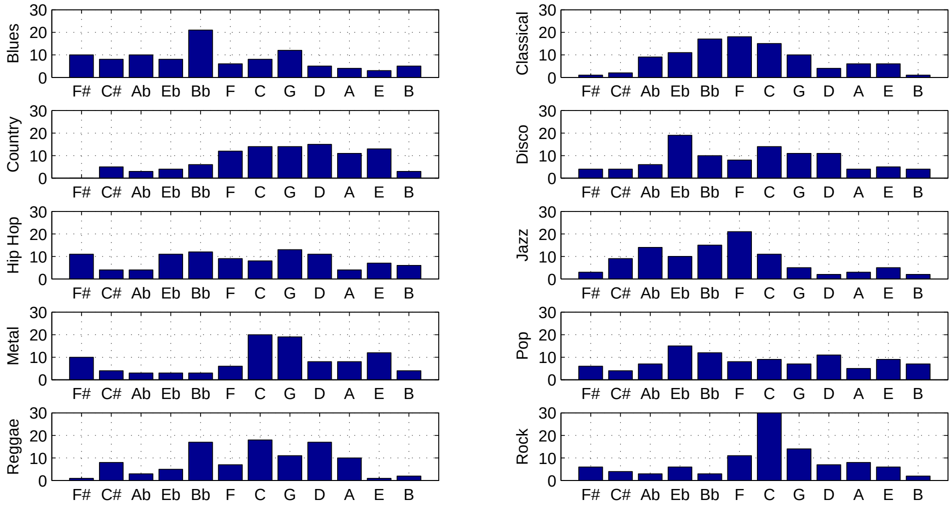

Figure 4.1: Key histograms of the GTZAN dataset on the circle of fifths scale. The vertical axis is the number of songs with a certain key.

is the La sound rather than the Do sound). Because a major scale and its relative minor scale share the identical composition of pitches, it is simpler to annotate both scales with the same label to show that they actually have the same pitch ingredients in the songs (e.g., songs in C major and A minor are both labeled with “C”). In cases where the scale is not apparent, we annotate the key based on the most repeated pitch.

Figure 4.1 shows the key histograms for different genres in the GTZAN dataset, using our annotation criteria, with keys ordered by the circle of fifths (C is in the center). We observe that genre is indeed key-related with the distribution centered around particular keys based on the instrumentation.

- Blues: peaks at B♭ and G. B♭ is the fundamental pitch of many horn instruments. G corresponds to the Do sound for the blues scale in E, which is the fundamental key for guitar.

- Classical: distribution around F, which is in between the horn instrument fundamental B♭ and the piano fundamental C.

- Country: broad distribution around D, with keys that are easy to play on guitars (e.g. G, D, A, E, C).

- Disco: peaks at E♭ and C. Disco frequently employs Blues scale. For C Blues, the Do sound is E♭.

- Hip Hop: distribution is not obvious. This genre typically does not have a key, as the main instruments are human voice and drums.

- Jazz: distribution is skewed towards flat keys (D♭, A♭, E♭, B♭), which are the fundamental horn pitches. The peak at F is similar to that of Classical.

- Metal: peaks at C, G, E and F♯. The G key correspond to E Blues. E is the pitch of the lowest string on guitar. In Metal, the lowest string is used extensively to create a massive feeling. The peak at F♯, corresponding to E♭ Blues, can be explained by the common practice of Metal artists to lower the tuning by one semi-tone, creating an even stronger metal feeling.

- Pop: distribution is not obvious. The peak at E♭ is the Blues-scale of the C key. The distributions of Pop and Disco are similar, due to similar instrumentation.

- Reggae: peaks at C (keyboard), D (guitar), B♭ (horns) and C♯ (B♭ Blues).

- Rock: significant distribution around C. The distribution is be related to the dominance of guitar and piano in this genre. Rock is arguably the most key-related genre in the GTZAN dataset.

In summary, there is a strong correlation between genre and key, with each genre having a unique key distribution. Such correlation most likely stems from the fundamental keys associated with the instruments used in each genre. For instance, the most common kind of clarinet is in the key of B♭, while the alto saxophone is in E♭. The four strings of a violin are tuned by standard to G, D, A and E. The piano has all its white keys in C major. Although it is entirely possible to play a song in any key, some keys are arguably easier to play than others, depending on the instruments used. Hence, the key characteristics of instruments could unexpectedly associate musical keys to specific genres.

4.3 Are MFCCs Invariant to Key and Tempo?

In this section we study the invariance of MFCCs to shifts in musical key and tempo.

The computation steps of MFCC is listed in Chapter 2. In this chapter, the MFCCs are extracted with the CATBox toolbox [8], using 40 mel-bands and 13 DCT coefficients. The frame size is 18 milliseconds, taken every 9 milliseconds.

4.3.1 Key and Tempo Transformations

To examine the changes of MFCC values to shifts in keys and tempos, we apply key shifting and tempo shifting musical transforms to each song in the GTZAN dataset. These transformations consist of sharpening/flattening the song up to 6 semitones, and changing the tempo 5% and 10% faster/slower. The transformations are performed with the WSOLA algorithm [43], which is implemented in the open-source audio editor Audacity [1]. The musical transforms are analogous to affine transforms of images, which deform an image without changing the general shape (e.g. rotating and skewing the number 1). Augmenting the dataset with affine transforms is a common technique in digit recognition tasks [36], where the enlarged training set improves classification accuracy by encouraging invariance to these deformations.

There are doubts that transforming a song to approximate the key-shifted and tempo-shifted version of the songs might not be appropriate, since such transforms might also contaminate the timbral characteristics of the songs. We argue that such an effect is minor for the following three reasons: 1) qualitatively speaking, the transformed songs sound perceptually very similar to the original song recorded in different key and tempo, with critical information for genre classification, such as instruments, musical patterns and rhythm characteristics, still preserved; 2) considering that musical instruments have different timbre in different registers, we limit the key shifts to the range of half an octave (from $\flat 6$ to $\sharp 6$); 3) we compared the MFCC values extracted from MIDI songs and their perfect key-transposed versions, and observed that the MFCC values vary in similar ways as in the key-transformed songs.

4.3.2 Comparison of MFCCs under Key and Tempo Transforms

For genre classification, MFCCs are often aggregated over a long-duration window using statistical methods [41, 6]. Motivated by this fact, we compare the original songs and their transformed versions by computing the Kullback-Leibler (KL) divergence [10] between corresponding windowed excerpts (3.5 seconds). Assuming that the MFCCs in a window follow a Gaussian distribution (e.g., as in [41]), the calculation of KL divergence between two windows is given by:

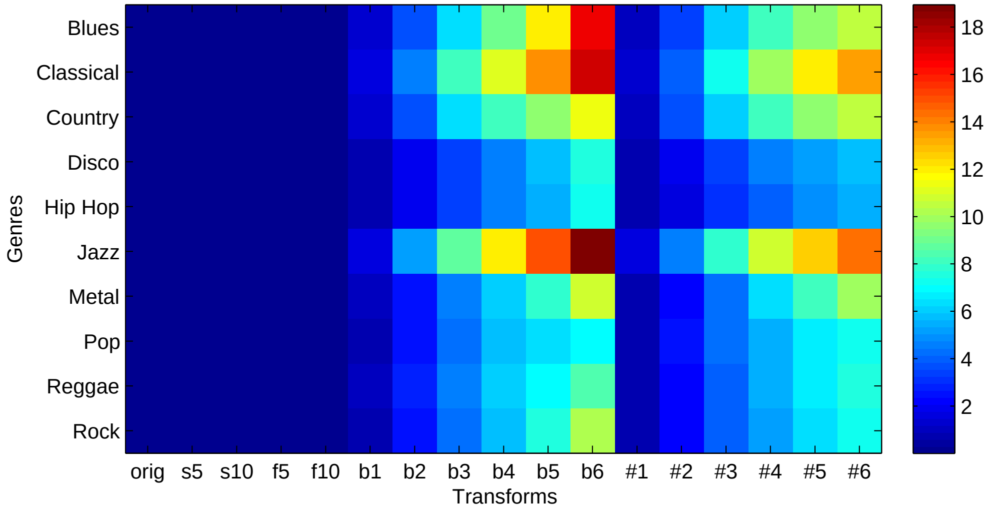

Figure 4.2: MFCC KL-divergence: the horizontal axis represents the key and tempo transforms, from left to right, original, 5% slower, 10% slower, 5% faster, 10% faster, key transform ♭1 to ♭6 and ♯1 to ♯6. The color represents the average KL divergence between corresponding frames in the original and transformed songs.

$$D_{KL}\left(\boldsymbol{N}_0 \parallel \boldsymbol{N}_1\right) = \frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \text{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + \left(\mu_1 - \mu_0\right)^{\top}\Sigma_1^{-1}\left(\mu_0 - \mu_1\right) - d\right) \tag{4.1}$$

where (μ_0, Σ_0) and (μ_1, Σ_1) are the mean and covariance for the two Gaussian distributions, and d is the dimension.

Figure 4.2 shows the KL divergence between different musical transforms of the same songs, averaged over each genre. From the figure, we see that key transforms affect the MFCC distribution, with larger key shifts affecting the distribution more. Interestingly, MFCCs for some genres are more sensitive to the changes in key, such as blues, jazz and metal. This can be explained by the fact that these genres have instruments with richer harmonic structure, and therefore the MFCCs change more since they model timbre. On the other hand, tempo transforms do not have a great effect on the distribution of MFCC values. This is because transforming a song in time does not change the frequency characteristics, but only the number of MFCC frames. Compressing a song subsamples the MFCC frame set, while stretching it adds new MFCC frames by interpolation. In both cases, the distribution of the MFCCs over the window remains about the same.

In the previous, we showed that genres have dominant keys, due to the instrumentation of the genre. On the other hand, in this section, we have shown that MFCCs, which are common features for genre classification, are not invariant to key transformations. This brings forward an interesting dilemma. Because genre is key dependent and MFCCs are not key invariant, then a classifier based on MFCCs may overfit to the dominant keys of the genre. The resulting classifier will then have poor accuracy on songs in the less common keys. In the next section, we look at learning a key-invariant genre classifier, by augmenting the classifier with different musical transforms.

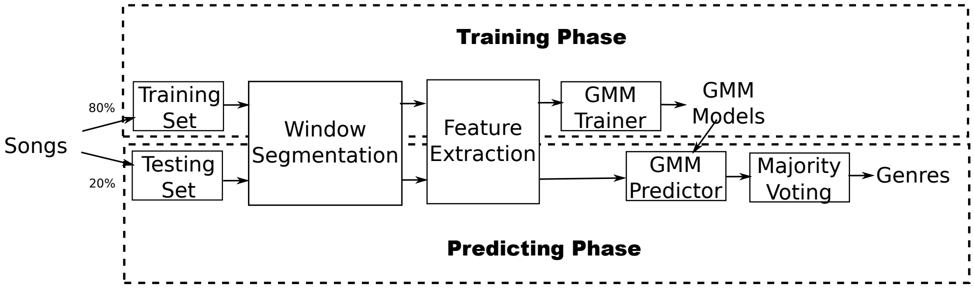

Figure 4.3: System architecture.

4.4 Genre Classification with Musical Transforms

In this paper, we adopt the genre classification system of [41, 6, 28]. Figure 4.3 shows the architecture of the system, which contains four steps. First, the input song is split into non-overlapping windows of equal length (as in [6], we use window length of 3.5 seconds). These windows then go through a feature extraction process, producing feature vectors which are compact representations of those windows. In particular, MFCCs are first extracted from the audio signal, and the mean and standard deviation of the MFCCs over the window are calculated as the feature vector. In the third step, the feature vector is fed to a Gaussian mixture model (GMM) classifier. The parameters of the GMM classifier are learned from the training set using the EM algorithm [12], which iteratively estimates the parameters by maximizing the likelihood of the training set. One GMM is learned for each genre. Given a feature vector extracted from a window, the GMM with the largest likelihood is selected as the genre label for the window. The labels for all the windows in a song are then aggregated with a majority voting process to produce a genre label for the song.

We can modify the genre classification system in two ways to make it invariant to musical transforms. First, in the training phase, we can expand the training set by adding transformed versions of the training songs, hence generating more examples for learning the genre classifier. Second, in the prediction phase, we can augment the classifier by processing the test song along with its transformed versions. The final label for the test song is the majority vote over all windows of all versions of the songs. The data augmentation step can be seen as adding a sample diffusion layer before either the training or the predicting phase of the system.

4.5 Experiments

In this section we present our experimental results on genre classification in the context of key and tempo augmentation.

4.5.1 Dataset and Experimental Setup

In our experiments, we use the GTZAN dataset [41], which contains 1000 song clips of 30 seconds each, with a sampling rate of 22050 Hz at 16 bits. There are 10 musical genres, each with 100 songs: Blues, Classical, Country, Disco, Hip hop, Jazz, Metal, Pop, Reggae, and Rock. We augment the original GTZAN dataset (denoted as the "Orig" dataset) using different combinations of musical transforms. The "Tempo" dataset contains the Orig dataset and its tempo variants, 5% and 10% faster/slower. The "Key" dataset contains the Orig dataset and its key variants from $\flat 6$ to $\sharp 6$. The "Tempokey" dataset is the union of the Tempo and Key datasets. We also augment our dataset with key transforms that are based on the circle of fifths. The "Fifth1" dataset contains the Orig dataset and its key variants with one step on the circle of fifths, i.e. $\flat 5$ and $\sharp 5$, while the “Fifth2” dataset contains variants with one more step, i.e. $\flat 2$ and $\sharp 2$. The circle of fifths augmented datasets are strict subsets of the Key dataset.

We carried out three different sets of experiments in combination with the 6 augmentations listed above. In the first experiment, denoted as AugTrain, the classifiers are trained using the augmented dataset, while genre prediction is performed using only the original songs. In the second experiment, denoted as AugPredict, the classifiers are trained only on the original dataset, while prediction is performed by pooling over the augmented song data. In the final experiment, denoted as AugBoth, both the classifier training and prediction use the augmented song data. Genre classification is evaluated using five random splits of the dataset, with 80% of the songs (and its variants) used for training, and the remaining 20% used for testing. The experiments are carried out on a range of parameters. We use MFCC lengths from 1 to 13 (i.e., the number of DCT coefficients), and vary the number of components in the GMM (K) from 1 to 20. We also assume diagonal covariance matrices in the GMM. In these experiments, we treat the number of GMM components as a tunable parameter. Alternatively, the value of K could be determined using a standard model selection criteria [14] .

4.5.2 Experimental Results

We first examine the effects of the system parameters, such as the size of the GMM and the length of the MFCCs. Figure 4.4a shows the classification accuracy, averaged over all the data augmentations and MFCC lengths, while varying the number of components in the GMM. In general, the classification accuracy increases with K, and there does not seem to be an over-fitting problem for large K, such as 20. Figure 4.4b shows the

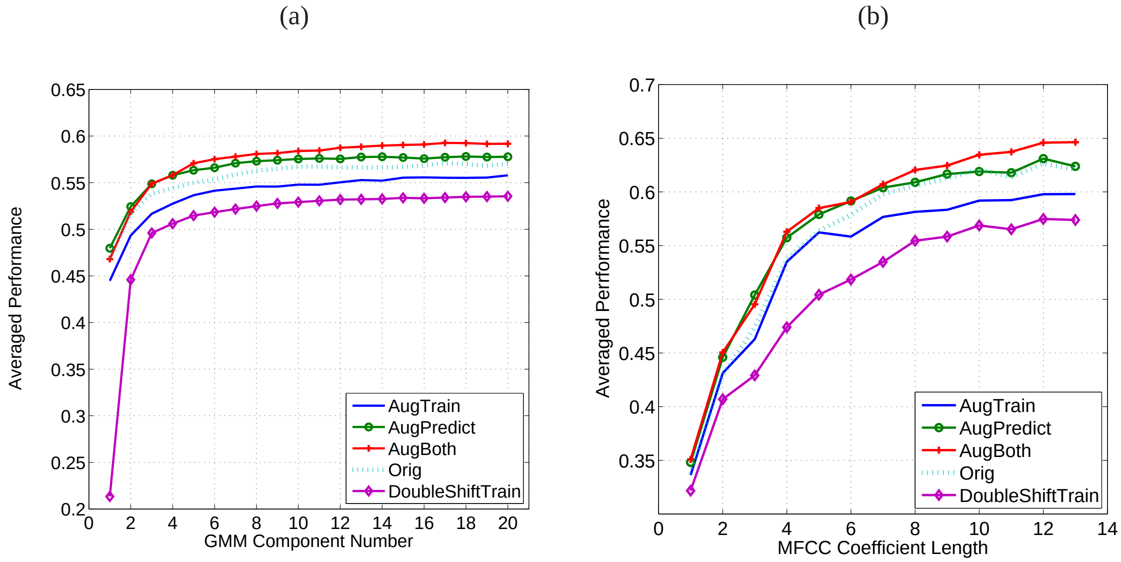

Figure 4.4: (a) Averaged accuracy for all datasets and MFCC lengths, while varying the number of GMM components (K); (b) Averaged accuracy for all datasets and GMM components, while varying the MFCC length.

accuracy, averaged over all data augmentations and GMMs, while varying the length of the MFCCs. Similarly, the accuracy improves as more MFCCs are added. In fact, despite their sensitivity to noise, these high-order coefficients provide useful details for genre classification. As a comparison, [41] limited their system to the first 5 MFCC coefficients and GMMs with K=5, and achieved 61% classification accuracy when using MFCCs with three other types of features. In contrast, our system scores 66.3% on the Orig dataset when using 13 MFCC features.

Next, we look at the effect of signal degradation when using the music transformation. In particular, we add noise to the Orig training set by applying a “double-shift” to each training song. This consists of first shifting the key of the song, and then shifting it back to the original scale. The result is a training song with noise added due to the musical transformation. The double-shifted training set is used to train the genre classifier, which then predicts genres on the Orig test data. This result is denoted as DoubleShiftTrain in Figure 4.4. In particular, using the noisy training data degrades the accuracy, when compared to the Orig performance (e.g, the accuracy drops 5% to 53.5% for K=20). However, in spite of this added noise to the training set, the system is still able to do genre classification, albeit with reduced accuracy.

Finally, we look at the effect of using the proposed data-augmented classifiers. From Figure 4.4, we observe that the AugTrain classifier gives constantly better performance than the DoubleShiftTrain classifier, while its performance is still lower than that of the Orig dataset. This suggests that using augmented training data improves the accuracy, at least compared to the unaugmented classifier using similar noisy training data. This improvement, however, is not enough to overcome the transformation noise. On the other hand, using data-augmented prediction (AugPredict) gives constantly better performance than the Orig dataset. Finally, using both data-augmented classification and prediction (AugBoth) achieves the best accuracy, dominating both AugPredict and Orig. Table 4.1 shows the average classification accuracy using different transformed datasets and data-augmentation schemes for K=20 and MFCC length 13. The best performance achieved for all experiments is 69.3%, using the AugBoth classifier with the Key transformations, K=18 and MFCC length 13.

Table 4.2 shows the classification accuracy for different genres using the AugBoth classifier. Comparing the genres, Classical has the highest accuracy, scoring over 90% on all datasets, followed by Jazz and Metal. In contrast, Disco and Rock are the two worst performing genres. In general, the augmentation of the dataset improves the genre classification. The only exception is the Rock genre, where augmentation always lowers the classification accuracy. Looking at the confusion matrix for AugBoth, we found that

	Tempo	Key	Tempokey	Fifth1	Fifth2	Average
Orig	–	–	–	–	–	64.5%
DoubleShiftTrain	–	–	–	–	–	61.9%
AugTrain	65.1%	62.0%	64.5%	60.5%	62.8%	63.0%
AugPredict	66.2%	63.6%	66.4%	61.0%	63.7%	64.2%
AugBoth	66.6%	67.8%	**68.9%**	67.5%	67.3%	67.6%

Table 4.1: Genre classification accuracy for different data-augmentation schemes and transformed datasets, for K=20 and MFCC length 13.

	Blues	Classical	Country	Disco	Hip-Hop	Jazz	Metal	Pop	Reggae	Rock	Average
Orig	59	92	62	41	64	86	77	58	61	45	64.5
Tempo	64	97	62	46	66	85	75	64	68	39	66.6
Key	62	99	67	55	65	90	83	64	60	33	67.8
Tempokey	63	98	67	55	65	91	87	61	63	39	68.9
Fifth1	61	98	67	52	63	88	83	63	62	38	67.5
Fifth2	64	94	63	58	63	90	79	64	66	32	67.3

Table 4.2: AugBoth Classification Rates for different genres, with K = 20 and MFCC length 13.

more instances of Rock are misclassified as Metal. On the other hand, Disco performs significantly better because less instances are misclassified as Blues, Pop and Rock.

4.5.3 Discussion

From these experimental results we have three conclusions. First, the MFCC feature set is largely a timbral feature set. From the confusion matrices we found that confusable genres have similar instrumentation. Additionally, genres with distinct instrumentation stand out from others easily, e.g., Classical uses orchestral instruments, while Metal has high frequency distorted guitar.

Second, in addition to timbral information, MFCCs also encodes key information, which eventually affects the genre classification accuracy. We observed that the key and tempo augmented classifiers have a significant change in performance over the baseline. Rock and Metal both use guitars and drums as the main instruments, but they have very different key distributions as shown in Figure 4.1. The confusion between Rock and Metal after key augmentation suggest that the classification of Rock music is partly due to musical keys. If we blur the lines between keys for these two genres, we are likely to lose such information, leading to a degradation of classification performance.

Third, making the genre classifier tempo- and key-invariant, via data augmentation, generally improves the classification accuracy. The accuracies of the AugTrain, AugPredict and AugBoth classifiers are significantly better than the noise-added DoubleShiftTrain baseline. Despite the noise from the imperfect musical transforms, the accuracy of the AugPredict and AugBoth classifiers are constantly better than the Orig baseline. These results suggest a method for boosting overall genre classification performance, by artificially generating transformed songs to augment the classifier training and prediction phases, thus strengthening the timbre-orientation of the classifier. However, some genres (e.g. Rock) will suffer from such augmentation since the recognition of that genre is partly due to musical keys.

While the concept of “musical genre” is perceptual and largely based on timbre information, there is still a strong correlation between genre and key, due to instrumentation, which should also be considered. Future work will look at combining timbral and key information, using appropriate machine learning models, to push the performance further. In addition, reducing the noise introduced by the musical transform will also likely improve the classification accuracy.

4.6 Conclusion

MFCCs are widely used audio features in music information retrieval. Extracted over a short-duration frame, MFCCs are typically perceived as a timbral descriptor. In this chapter, we have shown that the MFCCs are not invariant to changes in key, and hence they encode both timbral and key information. On the other hand, we found that musical genres, which should be independent of key, are in fact influenced by the fundamental keys of the instruments involved. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. We suggested an approach to address this problem, which consists of data-augmentation during the classifier training and prediction phases, with key and pitch transformations of the song. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

4.7 Acknowledgement

The text of this chapter, in full, is based on the material as it appears in [19] . The dissertation author was a primary researcher and first author of the cited material.

Chapter 5

Conclusion

In this thesis, we have presented our answers to two critical questions for the development of genre classification systems: 1) how possible is it to attack the musical feature extraction problem from an alternative perspective, an approach different from the traditional audio processing methodologies; 2) is the MFCC feature purely a timbral feature as believed by many, and is genre independent to musical keys and tempo? To address the first question, we employ the convolutional neural networks, a typical image recognition algorithm, as our feature extraction tool. The experiment shows that musical data indeed share many intrinsic characteristics as image data. Provided appropriate modeling techniques from the image information retrieval field, we could develop viable alternative solutions to the musical feature extraction problem. To the second question, we showed that the MFCC feature encodes both timbral and key information, and therefore the feature is not purely timbral. Likewise, we found that genre is related to musical keys for the fundamental keys of the instrumentations. The key inclination of genre and MFCC makes classification systems susceptible to the dominant keys of the genre. To solve this problem, we augment the classification system in the training and predicting phase, tuning the classifier more timbre-oriented. The augmentation improves the overall system performance.

Future works

The experiments in Chapter 3 shows that the CNN model is both difficult to train and not robust enough to generalize the classification to unseen musical data. One possible solution to the generalization problem is providing more training data for the system, while another is improving the overall system architecture with the developments in neural science research. By better understanding of the human auditory system, it is likely to produce fruitful results toward music classification.

To better improve the overall performance of the augmented classification system, we need to eliminate the noise introduced by the audio transform algorithms. One possible strategy is to scale the MFCC windows so as to mimic the key transforms. Doing so removes several intermediate processing in the system, which preserves data integrity. It could also be fruitful using different features than MFCC for the system input, such as the STFT and Mel-bins, to reveal potential features removed during the MFCC extraction process.

Bibliography

[1] Audacity, the free, cross-platform sound editor `http://audacity.sourceforge.net/`,.

[2] The Music Information Retrieval Evaluation eXchange (MIREX) `http://www.music-ir.org/mirex/wiki/2009:main_page`.

[3] R. Basili, A. Serafini, and A. Stellato. Classification of musical genre: a machine learning approach. In *Proceedings of The International Society for Music Information Retrieval*, 2004.

[4] J. Beament. *How we hear music: The relationship between music and the hearing mechanism*. Boydell & Brewer Inc, 2003.

[5] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, 2007.

[6] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and Adaboost for music classification. *Machine Learning*, 65(2):473–484, 2006.

[7] J. Bridle and M. Brown. An experimental automatic word recognition system. *JSRU Report*, 1003, 1974.

[8] Computer audition toolbox `http://cosmal.ucsd.edu/cal/projects/catbox/catbox.htm`.

[9] L. Chittka and A. Brockmann. Perception space: the final frontier. *PLoS Biol*, 3(4):e137, 2005.

[10] T. Cover and J. Thomas. *Elements of information theory*. John Wiley and sons, 2006.

[11] P. de León and J. Inesta. Feature-driven recognition of music styles. *Lecture Notes in Computer Science*, pages 773–781, 2003.

[12] A. Dempster, N. Laird, D. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[13] H. Deshpande, R. Singh, and U. Nam. Classification of music signals in the visual domain. In *Proceedings of the COST-G6 Conference on Digital Audio Effects*, 2001.

[14] R. Duda, P. Hart, and D. Stork. *Pattern classification*. 2001.

[15] D. Ellis. Classifying music audio with timbral and chroma features. In *Int. Symp. on Music Information Retrieval (The International Society for Music Information Retrieval)*, pages 339–340, 2007.

[16] J. Foote and S. Uchihashi. The beat spectrum: a new approach to rhythm analysis. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001*, pages 881–884, 2001.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[18] ISO 16:1975. *Acoustics – Standard tuning frequency (Standard musical pitch)*. ISO, Geneva, Switzerland.

[19] T. Li and A. Chan. Genre Classification and the Invariance of MFCC Features to Key and Tempo. In *Submitted to the 17th International Conference on MultiMedia Modeling*, 2010.

[20] T. Li, A. Chan, and A. Chun. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. In *Proc. International MultiConference of Engineers and Computer Scientists*, 2010.

[21] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 143–146, 2003.

[22] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proc. The International Society for Music Information Retrieval*, pages 34–41, 2005.

[23] T. Lidy, A. Rauber, A. Pertusa, and J. Inesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. *Proc. The International Society for Music Information Retrieval, Vienna, Austria*, 2007.

[24] B. Logan. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval*, volume 28, 2000.

[25] L. Lu, H. Zhang, and S. Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 8(6):482–492, 2003.

[26] M. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *Proc. ISMIR*, pages 594–599, 2005.

[27] C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of the International Conference on Music Information Retrieval*, volume 525, page 30, 2004.

[28] A. Meng, P. Ahrendt, and J. Larsen. Improving music genre classification by short time feature integration. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05)*, volume 5, 2005.

[29] R. Michelson, M. Merzenich, C. Pettit, and R. Schindler. A cochlear prosthesis: Further clinical observations; preliminary results of physiological studies. *The Laryngoscope*, 83(7):1116–1122, 1973.

[30] J. Movshon, I. Thompson, and D. Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *The Journal of Physiology*, 283(1):53, 1978.

[31] C. Müller. Speaker Classification I: Fundamentals, Features, and Methods. *Lecture Notes In Artificial Intelligence; Vol. 4343*, 2007.

[32] F. Pachet and J. Aucouturier. Improving timbre similarity: How high is the sky? *Journal of negative results in speech and audio sciences*, 1(1), 2004.

[33] D. Pearce and H. Hirsch. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Sixth International Conference on Spoken Language Processing*, 2000.

[34] P. Ponce de León and J. Iñesta. Musical style identification using self-organising maps. In *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, page 82. IEEE Computer Society, 2002.

[35] A. Schüz and R. Miller. *Cortical areas: unity and diversity*. CRC, 2002.

[36] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *International Conference on Document Analysis and Recogntion (ICDAR), IEEE Computer Society, Los Alamitos*, pages 958–962, 2003.

[37] H. Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *IEEE International Conference on Acoustics Speech and Signal Processing*, volume 2, 1998.

[38] J. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley and Sons, 2003.

[39] S. Stevens, J. Volkmann, and E. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8:185, 1937.

[40] S. Sundaram and S. Narayanan. Experiments in Automatic Genre Classification of Full-length Music Tracks using Audio Activity Rate. In *IEEE 9th Workshop on Multimedia Signal Processing, 2007. MMSP 2007*, pages 98–102, 2007.

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

[42] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[43] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *IEEE International Conference on Acoustic Speech and Signal Processing*, volume 2. Institute of Electrical Engineers Inc (IEE), 1993.

[44] C. Xu, N. Maddage, X. Shao, F. Cao, and Q. Tian. Musical genre classification using support vector machines. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 5, 2003.