electronics

MDPI

Article

Individual Violin Recognition Method Combining Tonal and Nontonal Features

Qi Wang and Changchun Bao *

Speech and Audio Signal Processing Laboratory, Faculty of Information Technology,
Beijing University of Technology, Beijing 100124, China; wangqi91@bjut.edu.cn
* Correspondence: baochch@bjut.edu.cn

Received: 29 March 2020; Accepted: 4 June 2020; Published: 8 June 2020

Abstract: Individual recognition among instruments of the same type is a challenging problem and it has been rarely investigated. In this study, the individual recognition of violins is explored. Based on the source–filter model, the spectrum can be divided into tonal content and nontonal content, which reflects the timbre from complementary aspects. The tonal/nontonal gammatone frequency cepstral coefficients (GFCC) are combined to describe the corresponding spectrum contents in this study. In the recognition system, Gaussian mixture models–universal background model (GMM–UBM) is employed to parameterize the distribution of the combined features. In order to evaluate the recognition task of violin individuals, a solo dataset including 86 violins is developed in this study. Compared with other features, the combined features show a better performance in both individual violin recognition and violin grade classification. Experimental results also show the GMM–UBM outperforms the CNN, especially when the training data are limited. Finally, the effect of players on the individual violin recognition is investigated.

Keywords: individual violin recognition; tonal/nontonal content; Gaussian mixture models–universal background model; violin grade classification

1. Introduction

Musical instrument recognition is a process to identify the type of musical instrument from the audio, which can be achieved through timbre analysis between different instruments. It has many practical applications, such as music information retrieval, audio content analysis and automatic music transcription.

In developing musical instrument recognition, two promising and challenging directions have emerged recently. One is applying the musical instrument recognition to the real-life situation. Instead of the isolated notes [1,2], more recent research deals with solos and multi-instrument music [3–5]. Another is restricting the instrument models to specific types and making a refined classification. Much research has been done to distinguish similar instruments within a family. For example, Banerjee et al. examined features and classification strategies for identifying four instruments of the string family [6]. Fragoulis et al. explored the discrimination between piano notes and guitar notes [7]. Avci et al. studied the machine learning-based classification of violin and viola sounds in the same notes [8]. Lukasik investigated the identification of individual instruments of the violin family [9].

The aim of this study is to tackle the problem of individual instrument recognition of the same type on the solo phrases. Owing to the same structure and shape, these individuals have similar timbre that is hard to be distinguished even by the experienced musicians. According to the results of individual recognition, the targeted individual will be identified among many instruments with identical appearance. For example, the solos played by the world-famous individual can be retrieved in the music database automatically. This can also help people search for the lost instruments through the audio. Furthermore, the similarity of timbre between individuals can be derived from the recognition results, which is an essential basis to classify quality grade of the instruments.

We choose violins for the individual recognition research. The violin was first known in 16th-century, and it is an important instrument in a wide variety of musical genres. Many researchers have sought explanations for the difference of the violins by investigating varnish and wood properties, plate tuning systems and the spectral balance of the radiated sound [10]. Violin performers were also required to rate violins for playability, articulation and projection [11]. However, acoustics and players are unable to provide an absolutely reliable standard for identifying and evaluating the violins. For example, even the experienced players cannot distinguish the individual of the Stradivari from other violins under the double-blind conditions [12]. The individual recognition of violin in our research may help address these issues.

In this study, the timbre analysis of the violin is investigated based on the source–filter model. A new set of the features are proposed, which are extracted using the tonal content and nontonal content. Inspired by speaker verification [13], a system based on Gaussian mixture models–universal background model (GMM–UBM) is built to recognize different violins. A solo dataset of violin is created in order to evaluate the recognition task of violin individuals. In addition to individual violin recognition, the experiments on grade classification of violins are conducted and players' influence on individual recognition is discussed as well.

The remainder of this study is organized as follows: Section 2 reviews the related works. Section 3 describes the source–filter model of the violin. Based on the model, the tonal and nontonal features are extracted in Section 4. Section 5 introduces the GMM–UBM recognition system. In Section 6, the solo dataset created for violin recognition is described. The experimental results and performance analysis are presented in Section 7. Finally, we draw the conclusions in Section 8.

2. Related Works

The traditional approach of instrument recognition is to extract acoustic features from the music signal and classify them using pattern recognition algorithms [14]. The existing feature sets include perception-based, temporal, spectral and timbral features [15]. For example, Essid et al. dealt with real music using autocorrelation coefficients, amplitude modulation features, mel-frequency cepstral coefficients (MFCC), spectral centroid, spectral slope, frequency derivative of the constant-Q coefficients, etc. [16]. In [17], the performance of 13 features was compared, including MFCCs, MPEG-7 features [18] and perception-based features. Among the individual feature schemes, the MFCC feature scheme gave the best classification performance. Experiments in [19,20] also favored the MFCCs over other features. Inspired by the MFCC, Duan et al. proposed the mel-scale uniform discrete cepstrum as the feature, which can model the timbre of mixture music [21]. For the audio signals with multiple sources, Costa et al. presented a sparse time–frequency feature by combining different instances of the fan–chirp transform [22].

In addition to the feature extraction module, there has been considerable interest in the classification module. For example, Diment et al. [23] trained a Gaussian mixture model (GMM) for the combination of the MFCCs and phase-related features. As the GMM cannot model dynamic evolution of the features, Zlatintsi et al. explored the hidden Markov model (HMM) in classification task [24]. In [25], a pairwise strategy was used in the classification task and the SVM was shown to outperform the GMM. Joder et al. conducted a large number of experiments to assess the impact of temporal integration on instrument recognition systems, and the classifiers consisted of the SVM, GMM and HMM [26]. Recently, Yip and Bittner made an open-source solo classifier using random forests for better performance [27].

More recently, deep learning techniques have been increasingly used owing to their superior performance. Han et al. used a convolutional neural network (CNN) to recognize predominant instrument in mixture music, which achieved higher accuracy than the conventional methods using SVMs [4]. In [28], Hilbert spectral feature and MFCCs were employed as inputs of a CNN to classify the predominant instrument at different time intervals. Koszewski et al. proposed a CNN-based automatic instrument tagging method, which provided promising recognition scores even for noisy recordings [29]. Hung et al. explored the CNNs with constant-Q coefficients and harmonic series feature for frame-level instrument recognition and pitch estimation [30,31]. Yu et al. also proposed to construct a CNN in a pattern of multitask learning, which used the auxiliary classification to assist the instrument classification [32]. Deep architectures can also implement feature extraction and classification in an end-to-end manner, which outperformed the traditional two-stage architectures in many tasks of recognition. For example, Li et al. showed that feeding raw audio waveforms to a CNN achieved 72% F-micro in discriminating instruments, whereas the MFCCs and random forest only achieved 64% [33].

The performance of instrument recognition approaches depends a lot on the annotated data, especially the deep learning methods. There are some efforts in the direction of datasets, such as the MUMS [34], RWC [35], ParisTech [26], UIOWA [36], MedleyDB [37], IRMAS [38] and MusicNet [39]. Currently, the annotated dataset for individual recognition is quite limited. In this study, we focus on individual violin recognition and create a dataset of violin individuals.

3. Source-Filter Model of Violin

The source–filter model originated from speech production has been used for decades in speech coding and synthesis. Similar to the speech signal, the music signal can be modeled as a combination of excitation and resonator [40]:

$$x(t) = p(t) \times h(t), \tag{1}$$

where t denotes time, $x(t)$ stands for the music signal, $p(t)$ the excitation friction, $h(t)$ the impulse response of resonator. Here "source" refers to vibrating strings and "filter" represents the resonance structure of the rest of violin. When a string vibrates, the bridge rocks and transmits the vibration to the resonance box. Then each component is amplified according to the resonance generated at that frequency. In the source–filter model, the excitation and resonator both determine the unique timbre of each violin.

The harmonic sequence is a typical structure in the spectrum of the music signal. The pitch is what is perceived as the tone, and its value can be determined according to the positions of harmonics for the violin. If two violins are playing the same note, the positions of two harmonic sequences will be the same. Under this simplified assumption, it will therefore be the envelope that makes the two sounds different. Hence, the timbre feature is often obtained through extracting the shape of the spectral envelope.

According to the sinusoids plus noise model, the pitched sound can be modeled as a sum of harmonics and noise residual [41]. The residual part is often ignored because of its low energy. In the "source" of the violin, the stable harmonics result from the main modes of vibration of the strings. The residual noise is generated by the sliding of the bow against the string, plus by other nonlinear behavior in excitation. Therefore, an improved source–filter model based on the sinusoidal plus residual decomposition is proposed [42]. The music signal $x(t)$ can be modeled by

$$x(t) = \{\sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)] + e(t)\} \times h(t), \tag{2}$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the rth sinusoid in the excitation, respectively, $e(t)$ is the noise component in the excitation, and $h(t)$ is the impulse response of resonator. According to Equation (2), the music signal can be divided into two parts, the tonal component $x_T(t)$ and the nontonal component $x_N(t)$. As shown in Equations (3) and (4), each part is produced by its own excitation, but the resonator is same.

$$x_T(t) = \{\sum_{r=1}^{R} A_r(t)\cos[\theta_r(t)]\} \times h(t), \tag{3}$$

$$x_N(t) = e(t) \times h(t), \tag{4}$$

The approximate process in frequency domain is illustrated in Figure 1. Considering the significance of the harmonics, the spectral envelope extracted from the music signal $x(t)$ is similar to the envelope of the tonal part in Figure 1d. As shown in Figure 1d, the shape of the spectral envelope is determined at the positions of partials, which is equivalent to a sampled envelope. The noise $e(t)$ is a stochastic signal and the spectral envelope of the nontonal component contains more comprehensive information about resonator. In Figure 1, the envelope shape of the nontonal part has big difference with that of the tonal part, which captures complementary information of timbre.

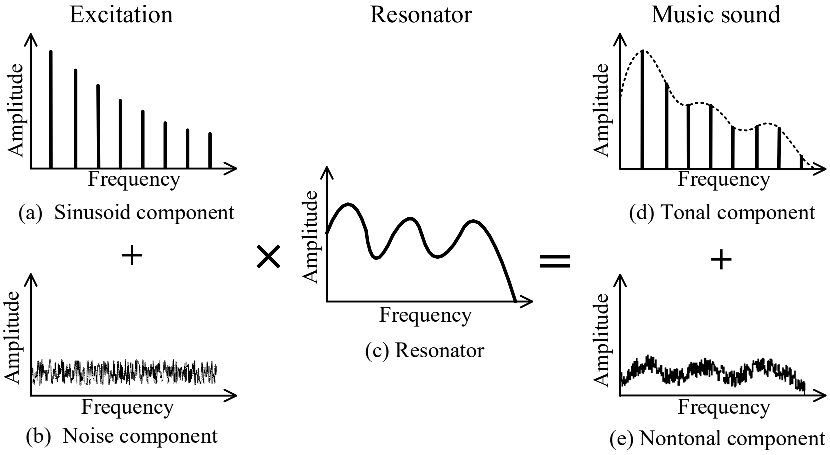


Figure 1. Improved source–filter model of the violin.

4. Feature Extraction

4.1. Tonal/Nontonal Content Extraction

The tonal part has been widely used in musical instrument classification by identifying the harmonic series from spectrum [17,20,43]. On the contrary, the research on nontonal content is quite limited [44,45]. The nontonal part refers to the non-harmonic residual, which is often used to describe the frequency components located between the partials [8].

We extract the tonal and nontonal contents in the frequency domain. The multiple pitch estimation algorithm based on harmonic product spectrum [46] is used to obtain the fundamental frequency of the music signal. For the fundamental frequency f_0, the position of the nth harmonic is around nf_0. Around each of these harmonic peaks, a region is defined to cover the peak. Here the region width of each peak is set to $0.3f_0$, so the nth harmonic peak interval is $[(n - 0.15)f_0, (n + 0.15)f_0]$. Figure 2 shows an isolated tonal lobe of a violin note, whose fundamental frequency is about 329 Hz. By resaving the spectral regions in each harmonic peak interval, all the tonal lobes are obtained.

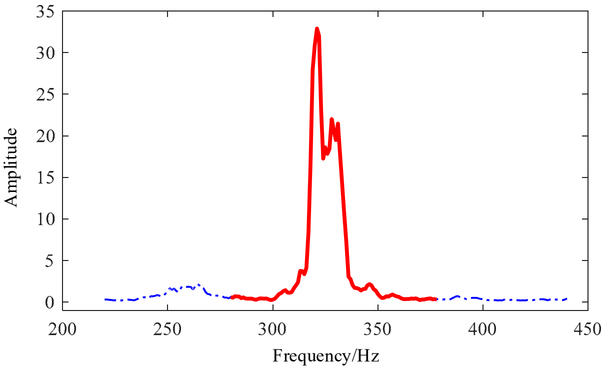


Figure 2. Isolated tonal lobe (solid line).

The nontonal content can be extracted by eliminating the tonal lobes from the spectrum. First, all the amplitude values in each harmonic peak interval are zeroed. Then the zero-magnitude points in spectrum will be spline interpolated. After the moving operation on the whole spectrum, a curve of nontonal content is obtained.

Figure 3 shows the original spectrum of a violin note E4 and its envelopes. The fundamental frequency of note E4 is about 329 Hz and the corresponding harmonic structure is obvious in Figure 3. The envelope of tonal content is similar to that of original spectrum. There is a difference between the envelopes of original spectrum and nontonal content. Compared with the envelopes of original spectrum and tonal content, several new peaks appear in the nontonal content, such as an obvious one located at 2000 Hz. Therefore, it is feasible to extract complementary timbre information from tonal content and nontonal content.

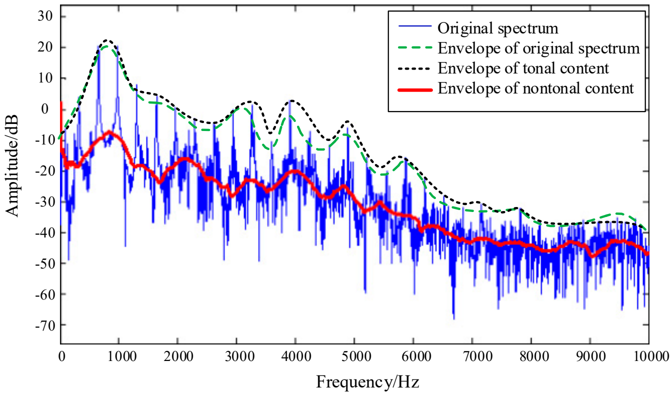


Figure 3. Spectrum and envelopes of the violin note E4.

4.2. Feature Extraction

Based on the source–filter model, the cepstral coefficients are usually used to parameterize the spectral envelope. The MFCC emphasizes perceptually meaningful frequencies using Mel scale and provides a more compact representation than the cepstral coefficients.

Similar to the MFCC, the gammatone frequency cepstral coefficient (GFCC) is derived using gammatone filters with equivalent rectangular bandwidth bands (ERB) [47]. The gammatone filterbank could model the human cochlear filtering better than the Mel filterbank. In speech recognition [48], speaker identification [49] and music retrieval [50], the GFCC performs substantially better than the conventional MFCC. Many researches also indicate that the GFCC exhibits superior noise robustness to the MFCC [51,52].

The impulse response of a gammatone filter centered at frequency f is given by

$$g(f,t) = \begin{cases} t^{a-1}e^{-2\pi bt}\cos(2\pi ft), & t \geq 0 \\ 0, & else \end{cases}, \tag{5}$$

where t denotes time, a is the filter order; rectangular bandwidth b increases with the center frequency f. With the designed gammatone filterbank, a time–frequency representation can be obtained from the outputs of the filterbank. Taking the log operation on the power spectrum and applying discrete cosine transform (DCT) on the log spectrum, the GFCC extraction is finished.

In this study, two timbre features named tonal GFCC and nontonal GFCC are introduced. These two features are extracted as in Figure 4. A tonal/nontonal content extraction block is inserted after the traditional FFT. Here the gammatone filterbank is modified by attenuating the "tails" of the response further away from the filter's center frequency.

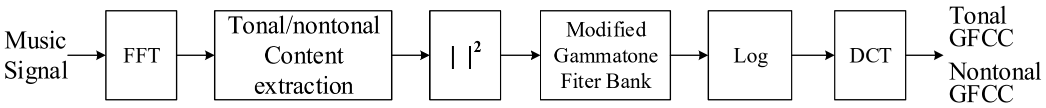

Music Signal
FFT
Tonal/nontonal Content extraction
| |2
Modified Gammatone Fiter Bank
Log
DCT
Tonal GFCC
Nontonal GFCC

Figure 4. Extraction process of the tonal gammatone frequency cepstral coefficients (GFCC) and nontonal GFCC.

Figure 5 shows the MFCC, tonal GFCC and nontonal GFCC consisting of vectors with dimension 40, respectively. All these features are extracted from violin A and violin B. Each subfigure contains features of 300 frames from solo *Humoreske* and the bold line corresponds to the average value. An excellent feature should show a significant difference between different individuals while keep stable in one violin. As shown in Figure 4, the tonal and nontonal features tend to better satisfy the above-mentioned requirements than the MFCC. The variances of the MFCC are 256.4, 283.6 for the violins A and B in the left column of Figure 5, whereas the variances of tonal GFCC are 163.7, 140.9 and the variances of nontonal GFCC drop to 102.5, 68.9 for the violins A and B, respectively. In the comparison of two rows, the MFCCs of two violins are similar while the nontonal features of A and B are easy to distinguish. For example, there is a valley around the seventh coefficient in the nontonal GFCC's average line of the violin A, whereas the counterpart of the violin B is in an increasing trend. To extract the timbre information more comprehensively, the tonal GFCC and nontonal GFCC are concatenated as the combined features. Among the combined feature vector, the first 40 components come from the tonal GFCC and the last 40 components come from the tonal GFCC.

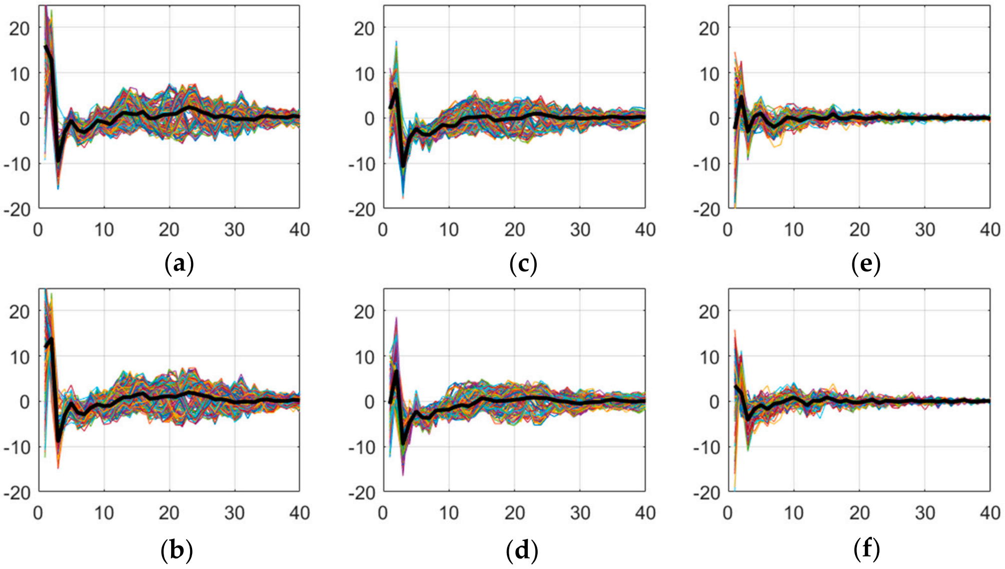


Figure 5. Features of two violins: (**a**) mel-frequency cepstral coefficients (MFCC) of violin A; (**b**) MFCC of violin B; (**c**) Tonal GFCC of violin A; (**d**) tonal GFCC of violin B; (**e**) nontonal GFCC of violin A; (**f**) nontonal GFCC of violin B.

5. Individual Violin Recognition System

The recognition system is built using a UBM for general violin representation and maximum a posteriori (MAP) adaption to derive individual violin models from the UBM. The basic components of the GMM–UBM recognition system are shown in Figure 6.

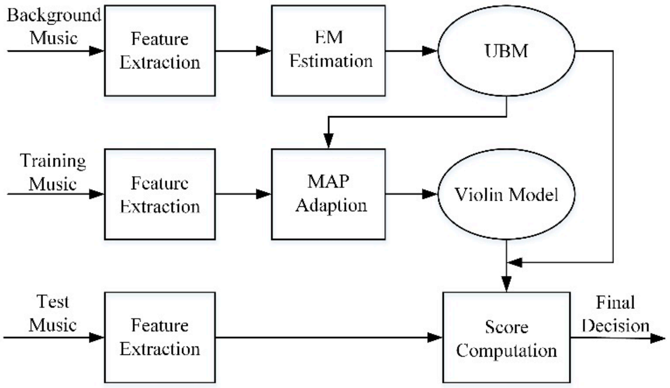


Figure 6. GMM–UBM based recognition system.

As a front-end processing stage in the system, the music is segmented into frames by a short window of 93 ms and features are extracted from the frames. The UBM is a single GMM trained to represent the individual-independent distribution of features. A number of solos from different violins are pooled as the background music data and a sequence of feature vectors are extracted for training a UBM. For the sequence of feature vectors $\boldsymbol{Y} = \{y_1, y_2, \ldots, y_T\}$, the likelihood function of the UBM is defined as

$$p(\boldsymbol{Y}|\lambda^{UBM}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\lambda^{UBM}), \tag{6}$$

where $p(\boldsymbol{y}_t|\lambda^{UBM})$ is the weighted linear combination of M unimodal Gaussian densities:

$$p(\boldsymbol{y}_t|\lambda^{UBM}) = \sum_{m=1}^{M} c_m^{UBM} N(\boldsymbol{y}_t|\mu_m^{UBM}, \Sigma_m^{UBM}), \tag{7}$$

In the density, c_m^{UBM} is the weight of mixture m. The mth Gaussian component is parameterized by a mean vector μ_m^{UBM} and a covariance matrix Σ_m^{UBM}. Denoted as $\lambda^{UBM} = \left\{c_m^{UBM}, \mu_m^{UBM}, \Sigma_m^{UBM} | m = 1, 2, \ldots, M\right\}$, all the parameters of the UBM are estimated via the iterative expectation–maximization (EM) algorithm [53].

Then a hypothesized violin model is trained using training data of the specified individual. Unlike the standard training of a model for the specified violin directly, our basic idea is to derive the violin's model by updating the parameters in the UBM via a MAP adaption. In this case, the components of the adapted GMM retain correspondence with the mixtures of the UBM.

Given the training feature vectors $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_T\}$ from the specified individual, the details of adaptation are as follows. Similar to the EM algorithm, we first compute the posterior probability for the mixture m in the UBM:

$$p(l = m|z_t; \lambda^{UBM}) = \frac{c_m^{UBM} N(z_t|\mu_m^{UBM}, \Sigma_m^{UBM})}{\sum\limits_{n=1}^{M} c_n^{UBM} N(z_t|\mu_n^{UBM}, \Sigma_n^{UBM})}, \tag{8}$$

where l is a latent variable to represent the index of Gaussian components. Then the sufficient statistics for the weight, mean and variance of the mth Gaussian component of the feature vector z_t are computed as follows:

$$N_m = \sum_{t=1}^{T'} p(l = m|z_t; \lambda^{UBM}), \tag{9}$$

$$F_m = \sum_{t=1}^{T'} p(l = m|z_t; \lambda^{UBM}) z_t, \tag{10}$$

$$S_m = \sum_{t=1}^{T'} p(l = m|\boldsymbol{z}_t; \lambda^{UBM})\boldsymbol{z}_t^2, \tag{11}$$

Next, the hypothesized violin model is derived from the UBM. The parameters of the *m*th mixture are updated using these new sufficient statistics:

$$c_m^{vn} = [\alpha_m N_m / T' + (1 - \alpha_m) c_m^{UBM}]\rho, \tag{12}$$

$$\mu_m^{vn} = \beta_m F_m + (1 - \beta_m)\mu_m^{UBM}, \tag{13}$$

$$\Sigma_m^{vn} = \gamma_m S_m + (1 - \gamma_m)[\Sigma_m^{UBM} + (\mu_m^{UBM})^2] - (\mu_m^{vn})^2, \tag{14}$$

where α_m, β_m, γ_m are adaption coefficients for the weight, mean and variance, respectively. Coefficient ρ is a scale factor computed over all adapted weights c_m^{vn} to ensure they are summed to unity.

To simplify the process of adaption, all the Gaussian components in the individual violin model share the same variance and weight parameters with the UBM. Only the mean μ_m^{vn} is updated here. Defining the mean adaption coefficient β_m as:

$$\beta_m = \frac{N_m}{N_m + r}, \tag{15}$$

where *r* is a relevance factor in the range of 8 to 20, the mean vectors of the hypothesized violin model can be updated as:

$$\mu_m^{vn} = \frac{N_m F_m + r\mu_m^{UBM}}{N_m + r}, \; m = 1, 2, \ldots, M \tag{16}$$

For the hypothesized violin model $\lambda^{vn} = \left\{c_m^{UBM}, \mu_m^{vn}, \Sigma_m^{UBM} | m = 1, 2, \ldots, M\right\}$, the log-likelihood ratio for test sequence $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{T''}\}$ is computed as the test score:

$$Score = \frac{1}{T''} \sum_{t=1}^{T''} \ln p(\boldsymbol{w}_t \big| \lambda^{vn}) - \ln p(\boldsymbol{w}_t \big| \lambda^{UBM}) \tag{17}$$

Finally, a decision threshold is determined for accepting or rejecting the hypothesized violin model. The corresponding violin will be selected as a candidate when its score exceeds the threshold. The scores of all candidate models will be sorted and the highest one is the optimal recognition result.

6. Violin Dataset for Individual Recognition

A solo dataset of violin is developed in order to evaluate the individual violin recognition system. The dataset consists of solo recordings from 86 violins with various characteristics. According to the raw material, production process and tonal quality, the violins are divided into low-grade, medium-grade and high-grade, respectively. The dataset includes violins of the three grades, whose prices range from 100 to 20,000$. More details of the violin dataset are given in Table 1.

Table 1. Details of the violin dataset.

Grade of Violins	**High**	**Medium**	**Low**
Price range [$]	3000–20,000	1000–2000	100–500
Number of violins (NV)	10	60	16
Number of performers per violin (NPV)	1	3	5
Number of solo contents (NSC)	3	4	4
Number of solos per violin (NSV = NPV × NSC)	3	12	20
Total number of solos (TNS = NV × NSV)	30	720	320

A total of 68 violin performers participated in the recoding. The performers have 3 to 55 years of experience on playing violin. Each performer played 3 or 4 excerpts of different content using the same violin. The music excerpts cover the classical music, popular music and Chinese folk music. For each violin, the solos of same content were performed by different players. For example, there are

60 medium-grade violins and each violin was played by 3 performers. In this case, each player performed 4 music excerpts, including *Turkish March*, *Humoreske*, *Can You Feel My Love* and *Early Spring*. It should be noted that the number of performers for high-grade violins is quite limited, because each valuable violin was only played by its owner.

All the solo excerpts were played indoor in a quiet environment. The audio files were recorded at 44.1 kHz sampling frequency with a resolution of 32 bits per sample. After the recording sessions, all the music data were reviewed and processed to remove silence at the beginning and end of the recordings. Each solo excerpt lasts about 3 min and the 1070 files contain more than 50 h of audio signals.

7. Experiments and Results

7.1. Individual Violin Recognition

In order to obtain a general violin representation independent of the dataset, 140 solos of unknown violins were downloaded from the Internet as the background data. All files in the violin dataset were divided into two groups referred to as the training data and test data. Three partitioning ways of data are shown in Table 2. For each violin, at least one solo file was randomly chosen as the training set. In this way, 86 violin models were derived. The remaining solo excerpts for each individual were collected as the test set.

Table 2. Data partition of the training and test sets.

		Train1	**Train2**	**Train3**
Training set	Number of solos per violin	1	2	≥1
	Total number of solos	86	172	535
Test set	Total number of solos	984	898	535

All the music data were segmented into the frames by a 93 ms window progressing at a 46.5 ms frame rate. The 40-dimensional tonal GFCC and 40-dimensional nontonal GFCC were concatenated as the combined features, which were fed into the recognition system. In the GMM–UBM system, the models of 64 Gaussian components were trained. The relevance factor r was fixed at 16 in the process of adaption.

We also carried out the experiments using other features and classifiers. The features for comparison contain the MPEG-7, linear prediction cepstral coefficients (LPCC) and MFCC, which are widely used in instrument classification. Specifically, the MPEG-7 set includes seven feature descriptors: harmonic centroid, harmonic deviation, harmonic spread, harmonic variation, spectral centroid, log attack time and temporal centroid.

The classifiers used for comparison are GMM and CNN. For each violin, a GMM of 64 Gaussian components was trained via the EM algorithm. The CNN model was based on the Inception-v3 [54]. Its architecture and parameters are shown in Appendix A. The CNN took 15-frame features as the input. For the n-dimensional feature, the dimension of input was $n \times 15 \times 1$. The CNN had 86 units in the output layer, corresponding the 86 violins. We also trained a CNN in an end-to-end manner, which employed the raw audio waveforms as input. The CNN had the same architecture and parameters as the sample-level network given in [55]. The raw waveform input was set to 59,049 samples (2678 ms at 22.05 kHz sampling frequency), and the dimension of output was 86.

Considering that the training set were chosen randomly, all the experiments were repeated 10 times to obtain the average. The accuracy of individual violin recognition is shown in Table 3. The accuracy is the fraction of correct solo excerpts among all the test excerpts, which is equal the micro-averaged measure F-micro in this multi-label problem. We also used the macro-averaged measure F-macro [56] to evaluate the performance of recognition across 86 classes. The F-macro of individual violin recognition is presented in Table 4.

Table 3. Accuracy of individual violin recognition (%).

	Classifier	MPEG-7	LPCC	MFCC	Combined Features	Raw Waveform
Train1	GMM–UBM	55.04	58.35	62.05	63.98	–
	GMM	29.96	34.48	38.16	45.50	–
	CNN	24.82	30.79	36.81	40.06	39.32
Train2	GMM–UBM	62.72	66.81	70.23	73.41	–
	GMM	41.73	49.41	55.27	60.61	–
	CNN	29.74	40.98	51.65	56.37	57.64
Train3	GMM–UBM	67.94	69.83	78.22	82.35	–
	GMM	43.91	58.53	64.37	74.57	–
	CNN	36.93	42.72	59.28	66.47	66.87

Table 4. F-macro of individual violin recognition (%).

	Classifier	MPEG-7	LPCC	MFCC	Combined Features	Raw Waveform
Train1	GMM–UBM	53.82	57.09	60.51	62.97	–
	GMM	28.16	33.17	35.83	45.29	–
	CNN	21.33	29.64	35.99	38.44	37.51
Train2	GMM–UBM	61.51	66.03	69.67	72.91	–
	GMM	39.89	46.88	53.53	58.47	–
	CNN	25.07	36.70	48.45	54.32	55.87
Train3	GMM–UBM	67.01	68.21	77.02	79.36	–
	GMM	41.14	52.86	60.74	72.89	–
	CNN	28.47	33.43	52.50	62.12	62.91

Among all the extracted features in Tables 3 and 4, the combined features perform best regardless of the type of classifier and the amount of the training data. The performance of the GMM–UBM system is superior to that of the GMM system. The UBM can model the individual-independent distribution of the features using background data, so the GMM–UBM performs better. As the number of training samples increases, the differences between the performance of the GMM and GMM–UBM become smaller. When the training data are more abundant, the superiority of the UBM is less obvious. In "Train1" and "Train2", the performance of the CNN system is inferior to that of the two GMM-based systems. This is largely due to the small amount of training data limits the performance of the CNN. In "Train3", the performance of the CNN system is still unsatisfactory. One of the reasons is that the training data of "Train 3" is class-imbalanced. For example, there are only 3 solos for each high-grade violin in the dataset. The training data for each high-grade violin are less than other violins. We can also observe that the end-to-end CNN system's performance is similar to or even superior to that of the CNN with the combined features. This indicates that the CNN could learn some effective representations from the raw waveform audio signals.

On the metrics of accuracy and F-macro, we further performed the paired t-test to compare the performance of proposed features and MFCC. The p-values are presented in Table 5. Most of the p-values are smaller than 0.05, which demonstrates that the superiority of the combined features to the MFCC is statistically significant.

Table 5. *p*-value between the combined features and MFCC in individual violin recognition.

		p-Value (Accuracy)	*p*-Value (F-macro)
Train1	GMM–UBM	0.0753	0.0496
	GMM	0.0027	0.0019
	CNN	0.0375	0.0520
Train2	GMM–UBM	0.0199	0.0260
	GMM	0.0165	0.0121
	CNN	0.0114	0.0098
Train3	GMM–UBM	0.0487	0.0561
	GMM	0.0004	<0.0001
	CNN	<0.0001	<0.0001

7.2. Violin Grade Classification

Considering the grade information instead of the individual label, the classification of violin grades could be implemented similarly. The system's output was three grades rather than 86 individuals. With the same partitioning of the training and test sets given in Table 2, we carried out the classification experiments. The accuracy and F-macro of violin grade classification are presented in Tables 6 and 7, respectively.

Table 6. Accuracy of violin grade classification (%).

	Classifier	MPEG-7	LPCC	MFCC	Combined Features	Raw Waveform
Train1	GMM–UBM	81.13	83.83	86.34	89.35	–
	GMM	65.94	70.84	76.62	80.97	–
	CNN	65.67	70.53	71.46	77.56	71.87
Train2	GMM–UBM	84.99	88.16	92.04	94.18	–
	GMM	72.20	74.38	86.61	90.35	–
	CNN	72.87	78.91	82.75	86.83	86.93
Train3	GMM–UBM	88.32	91.42	96.77	97.96	–
	GMM	75.36	80.89	91.08	94.49	–
	CNN	76.05	78.64	88.42	93.81	92.02

Table 7. F-macro of violin grade classification (%).

	Classifier	MPEG-7	LPCC	MFCC	Combined Features	Raw Waveform
Train1	GMM–UBM	63.35	65.42	74.69	80.82	–
	GMM	41.42	50.01	59.56	64.20	–
	CNN	41.70	45.48	52.57	60.23	50.67
Train2	GMM–UBM	68.37	72.94	76.98	84.09	
	GMM	51.32	55.77	68.18	77.64	–
	CNN	50.88	59.09	66.68	73.54	73.75
Train3	GMM–UBM	74.73	82.69	92.45	93.90	–
	GMM	57.79	64.26	78.64	86.98	–
	CNN	50.09	56.38	79.52	83.59	82.89

According to the results in Tables 6 and 7, we can draw a conclusion similar to that in individual violin recognition. For each classifier, the combined features lead to better performance than other features. The GMM–UBM's superiority to the GMM and CNN is still obvious. Compared with the results shown in Tables 3 and 4, all the systems perform better in violin grade classification than individual violin recognition. The grade of violins is a broader category than individual, which is responsible for better performance. We could also observe a big difference between the values of accuracy and F-macro in Tables 6 and 7. The accuracy aggregates the contributions of all test solo excerpts to compute a metric, whereas the F-macro computes the metric independently for each class and then take the average. When the classification results of the three grades differ a lot, there is an obvious difference between the overall accuracy and F-macro.

In the violin dataset, 60 violins belong to the medium grade and the solos in this grade account for 67% of the total data. For the majority grade, the accuracy of violin grade classification is presented in Table 8. Compared with the results shown in Table 6, all the systems obtain higher accuracy metrics on the medium-grade data. This is largely due to the training data of the medium grade are more sufficient than other grades. In "Train3", the CNN performs better than the two GMM-based systems on the classification of medium-grade violins. We believe that the CNN can be a promising model when the training data are sufficient and balanced.

Table 8. Classification accuracy of the medium-grade violins (%).

	Classifier	MPEG-7	LPCC	MFCC	Combined Features	Raw Waveform
Train1	GMM–UBM	81.79	84.34	94.14	96.14	–
	GMM	79.93	81.32	84.87	89.51	–
	CNN	79.63	84.10	81.48	88.79	75.62
Train2	GMM–UBM	88.09	91.15	96.94	98.31	–
	GMM	82.65	85.51	92.34	95.23	–
	CNN	83.16	88.27	91.16	91.66	90.29
Train3	GMM–UBM	89.27	91.51	98.59	98.86	–
	GMM	86.74	89.97	95.51	96.19	–
	CNN	89.58	91.98	98.61	98.96	98.26

A paired t-test was also conducted to compare the performance of the classification systems using the combined features and MFCC. As shown in Table 9, the p-values demonstrate that the superiority of the combined features to MFCC is statistically significant.

Table 9. p-value between the combined features and MFCC in violin grade classification.

		p-Value (Accuracy)	p-Value (F-macro)
Train1	GMM–UBM	0.0363	0.0351
	GMM	0.0018	0.0048
	CNN	0.0057	0.0056
Train2	GMM–UBM	0.0590	0.0082
	GMM	0.0086	0.0037
	CNN	0.0096	0.0187
Train3	GMM–UBM	0.0240	0.0423
	GMM	0.0207	0.0058
	CNN	<0.0001	0.0008

7.3. Effect of Performer

The effect of the violin performer was also discussed in the individual violin recognition systems using the combined features. According to the label information of the player, two set of experiments were conducted. In the same-performer scheme, the training data and test data for each violin were performed by the same player. In the different-performer scheme, the training data and test data for each violin were from different players. The corresponding ways of data partitioning are shown in Table 10.

Table 10. Data partition of the two performer-based schemes.

		Same-Performer		**Different-Performer**	
		Train1	**Train2**	**Train1**	**Train2**
Training set	Number of solos per violin	1	2	1	2
	Total number of solos	86	172	86	172
Test set	Total number of solos	248	162	736	736

The recognition accuracy rate (%) of the two schemes are presented in Table 11. There is a huge divide between the results of the same-performer scheme and the different-performer scheme. The accuracy rate has a prominent improvement when all the training data and test data are played by the same person. This indicates that the classifiers are overfitted to some extent. The classifiers are too closely fit to the limited training set, so the same-performer scheme obtains a better performance. In the comparison of the results between the two performer-based schemes, the difference of the GMM–UBM system is smaller than that of the GMM system. This is due to that the UBM could model the performer-independent distribution of the features using background data, which mitigates the overfitting of GMM. For each violin in the dataset, all solos played by the specific performer are of different contents. The high accuracy of the same-performer scheme also shows that, the influence of the performer is more prominent than the solo content in the recognition experiments.

Table 11. Comparison of performance between the two performer-based schemes.

	Same-Performer			**Different-Performer**		
	GMM–UBM	**GMM**	**CNN**	**GMM–UBM**	**GMM**	**CNN**
Train1	92.72	86.72	64.23	56.39	33.66	33.23
Train2	96.91	89.49	76.49	64.95	47.42	50.27

8. Conclusions

This paper proposed the combined features consisting of tonal GFCC and nontonal GFCC, which could capture more comprehensive information about timbre. Utilizing the proposed features, a framework based on the GMM–UBM was built to identify different violins. The framework employed a UBM to represent the violin-independent distribution of the combined features and a MAP adaption to derive individual violin models.

In order to evaluate the performance of individual violin recognition, a solo dataset consisting of 86 violins was developed. Among all the extracted features in this paper, the proposed features performed best in both individual violin recognition and violin grade classification. The GMM–UBM's superiority to the CNN was more obvious when the size of training set was smaller. The success of deep learning often hinges on the availability of sufficient training data, and the small amount of training data limited the performance of the CNN. The UBM could model the individual-independent distribution of the features using background data, so the GMM–UBM performed better with limited training data. Considering the great influence of players, the performer-independent individual recognition and a more robust model are promising in the future.

Author Contributions: Conceptualization, Q.W. and C.B.; methodology, Q.W.; software, Q.W.; investigation, Q.W.; resources, Q.W.; data curation, Q.W.; writing—original draft preparation, Q.W.; writing—review and editing, C.B.; visualization, Q.W.; supervision, C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Beijing Postdoctoral Research Foundation, Grant Number ZZ2019-70.

Acknowledgments: The authors are grateful to the thorough reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

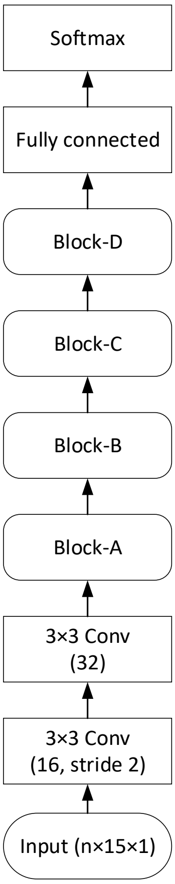
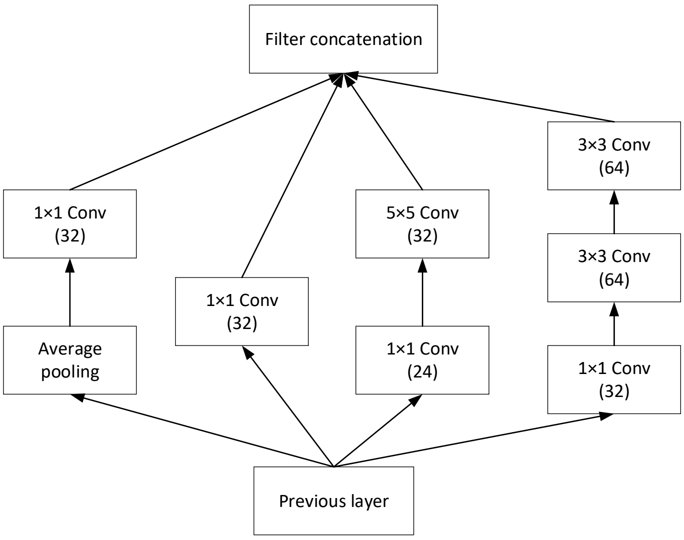
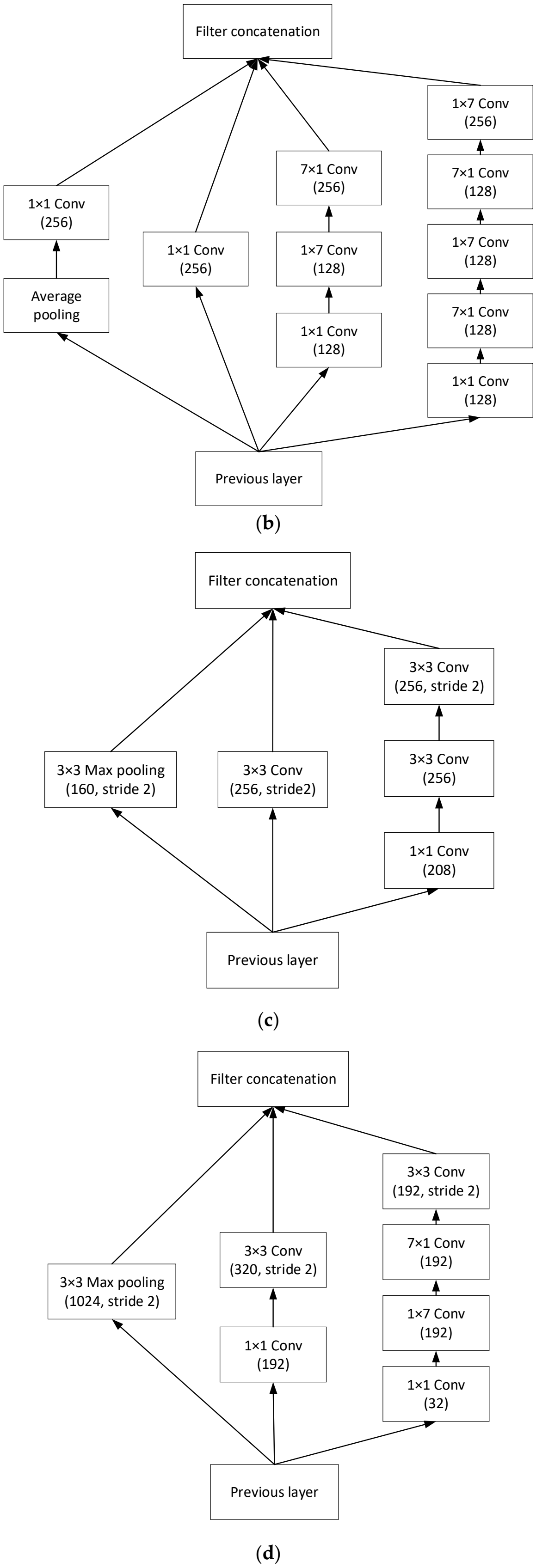


Figure A1. Diagram of the Inception network.



(**a**)

Figure A2. *Cont.*



(**b**)

(**c**)

(**d**)

Figure A2. Block structure in the Inception network. (**a**) block A, (**b**) block B, (**c**) block C, (**d**) block D.

References

1. Kaminskyj, I.; Czaszejko, T. Automatic recognition of isolated monophonic musical instrument sounds using kNNC. *J. Intell. Inf. Syst.* **2005**, *24*, 199–221. [CrossRef]
2. Heittola, T.; Klapuri, A.; Virtanen, T. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Kobe, Japan, 26–30 October 2009; pp. 327–332.
3. Yu, L.; Su, L.; Yang, Y. Sparse cepstral codes and power scale for instrument identification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 7460–7464.
4. Han, Y.; Kim, J.; Lee, K. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 208–221. [CrossRef]
5. Gururani, S.; Sharma, M.; Lerch, A. An attention mechanism for musical instrument recognition. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 4–8 November 2019; pp. 83–90.
6. Banerjee, A.; Ghosh, A.; Palit, S.; Ballester, M.A.F. A novel approach to string instrument recognition. In Proceedings of the International Conference on Image and Signal Processing (ICISP), Taipei, Taiwan, 22–25 September 2018; pp. 165–175.
7. Fragoulis, D.; Papaodysseus, C.; Exarhos, M.; Roussopoulos, G.; Panagopoulos, T.; Kamarotos, D. Automated classification of piano-guitar notes. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1040–1050. [CrossRef]
8. Avci, K.; Arican, M.; Polat, K. Machine learning based classification of violin and viola instrument sounds for the same notes. In Proceedings of the IEEE Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; pp. 1–4.
9. Lukasik, E. Long term cepstral coefficients for violin identification. In Proceedings of the 128 AES Convention, London, UK, 22–25 May 2010.
10. Wollman, I.; Fritz, C.; Poitevineau, J. Influence of vibrotactile feedback on some perceptual features of violins. *J. Acoust. Soc. Am.* **2014**, *136*, 910–921. [CrossRef]
11. Saitis, C. Evaluating Violin Quality: Player Reliability and Verbalization. Ph.D. Thesis, McGill University, Montréal, QC, Canada, 2013.
12. Fritz, C.; Curtin, J.; Poitevineau, J.; Borsarello, H.; Wollman, I.; Tao, F.-C.; Ghasarossian, T. Soloist evaluations of six Old Italian and six new violins. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 7224–7229. [CrossRef]
13. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker verification using adapted Gaussian mixture models. *Digit. Signal Process.* **2000**, *10*, 19–41. [CrossRef]
14. Bai, M.R.; Chen, M.-C. Intelligent preprocessing and classification of audio signals. *J. Audio Eng. Soc.* **2007**, *55*, 372–384.
15. Brown, J.C.; Houix, O.; McAdams, S. Feature dependence in the automatic identification of musical woodwind instruments. *J. Acoust. Soc. Am.* **2001**, *109*, 1064–1072. [CrossRef]
16. Essid, S.; Richard, G.; David, B. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Trans. Audio Speech Lang. Process.* **2005**, *14*, 68–80. [CrossRef]
17. Deng, J.; Simmermacher, C.; Cranefield, S. A study on feature analysis for musical instrument classification. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2008**, *38*, 429–438. [CrossRef]
18. Peeters, G.; Mcadams, S.; Herrera, P. Instrument sound description in the context of MPEG-7. In Proceedings of the International Computer Music Conference (ICMC), Berlin, Germany, 27 August–1 September 2000; pp. 166–169.
19. Eronen, A. Comparison of features for musical instrument recognition. In Proceedings of the IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA), New York, NY, USA, 21–24 October 2001; pp. 19–22.
20. Nielsen, A.; Sigurdsson, S.; Hansen, L.; Arenas-Garcia, J. On the relevance of spectral features for instrument classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, HI, USA, 15–20 April 2007; pp. 485–488.
21. Duan, Z.; Pardo, B.; Daudet, L. A novel cepstral representation for timbre modeling of sound sources in polyphonic mixtures. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 7495–7499.

22. Costa, M.V.M.; Apolinário, I.F.; Biscainho, L.W.P. Sparse time-frequency representations for polyphonic audio based on combined efficient fan-chirp transforms. *J. Audio Eng. Soc.* **2019**, *67*, 894–905. [CrossRef]
23. Diment, A.; Rajan, P.; Heittola, T.; Virtanen, T. Modified group delay feature for musical instrument recognition. In Proceedings of the International Symposium on Computer Music Multidisciplinary Research, Marseille, France, 15–18 October 2013; pp. 431–438.
24. Zlatintsi, A.; Maragos, P. Multiscale fractal analysis of musical instrument signals with application to recognition. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *21*, 737–748. [CrossRef]
25. Essid, S.; Richard, G.; David, B. Musical instrument recognition by pairwise classification strategies. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1401–1412. [CrossRef]
26. Joder, C.; Essid, S.; Richard, G. Temporal integration for audio classification with application to musical instrument classification. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 174–186. [CrossRef]
27. Yip, H.; Bittner, R.M. An accurate open-source solo musical instrument classifier. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Suzhou, China, 23–28 October 2017.
28. Kim, D.; Sung, T.T.; Cho, S.Y.; Lee, G.; Sohn, C.B. A single predominant instrument recognition of polyphonic music using CNN-based timbre analysis. *Int. J. Eng. Technol.* **2018**, *7*, 590–593. [CrossRef]
29. Koszewski, D.; Kostek, B. Musical instrument tagging using data augmentation and effective noisy data processing. *J. Audio Eng. Soc.* **2020**, *68*, 57–65. [CrossRef]
30. Hung, Y.N.; Yang, Y.H. Frame-level instrument recognition by timbre and pitch. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 23–27 September 2018; pp. 135–142.
31. Hung, Y.N.; Chen, Y.A.; Yang, Y.H. Multitask learning for frame-level instrument recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 381–385.
32. Yu, D.; Duan, H.; Fang, J.; Zeng, B. Predominant instrument recognition based on deep neural network with auxiliary classification. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 852–861. [CrossRef]
33. Li, P.; Qian, J.; Wang, T. Automatic instrument recognition in polyphonic music using convolutional neural network. *arXiv* **2015**, arXiv:1511.05520.
34. Eerola, T.; Ferrer, R.; Flores, R.F. Instrument Library (MUMS) Revised. *Music Percept. Interdiscip. J.* **2008**, *25*, 253–255. [CrossRef]
35. Goto, M.; Hashiguchi, H.; Nishimura, T.; Oka, R. RWC music database: Popular, classical and jazz music databases. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 13–17 October 2002; pp. 287–288.
36. University of Iowa Musical Instrument Samples. Available online: http://theremin.music.uiowa.edu/MIS.html (accessed on 4 May 2019).
37. Bittner, R.M.; Salamon, J.; Tierney, M.; Mauch, M.; Cannam, C.; Bello, J.P. MedleyDB: A multitrack dataset for annotation intensive MIR research. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 155–160.
38. Bosch, J.; Janer, J.; Fuhrmann, F.; Herrera, P. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 8–12 October 2012; pp. 559–564.
39. Thickstun, J.; Harchaoui, Z.; Kakade, S.M. Learning features of music from scratch. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017; pp. 1–14.
40. Valimaki, V.; Pakarinen, J.; Erkut, C.; Karjalainen, M. Discrete-time modelling of musical instruments. *Rep. Prog. Phys.* **2005**, *69*, 1–78. [CrossRef]
41. Serra, X. Musical sound modeling with sinusoids plus noise. In *Musical Signal Processing*; Routledge: New York, NY, USA, 1997; pp. 91–122.
42. Caetano, M.; Rodet, X. A source-filter model for musical instrument sound transformation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 137–140.
43. Barbedo, J.G.A.; Tzanetakis, G. Musical instrument classification using individual partials. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *19*, 111–122. [CrossRef]

44. Livshin, A.; Rodet, X. The importance of the non-harmonic residual for automatic musical instrument recognition of pitched instruments. In Proceedings of the AES 120th Convention, Paris, France, 20–23 May 2006; pp. 1–5.
45. Wu, Y.; Wang, Q.; Liu, R. Music instrument classification using nontonal MFCC. In Proceedings of the International Conference on Frontiers of Manufacturing Science and Measuring Technology, Taiyuan, China, 24–25 June 2017; pp. 417–420.
46. Chen, X.; Liu, R. Multiple pitch estimation based on modified harmonic product spectrum. *Lect. Notes Electr. Eng.* **2012**, *211*, 271–279. [CrossRef]
47. Qi, J.; Wang, D.; Jiang, Y.; Liu, R. Auditory features based on Gammatone filters for robust speech recognition. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013; pp. 305–308.
48. Meng, X.T.; Yin, S. Speech recognition algorithm based on nonlinear partition and GFCC features. *Appl. Mech. Mater.* **2014**, *556*, 3069–3073. [CrossRef]
49. Shi, X.; Yang, H.; Zhou, P. Robust speaker recognition based on improved GFCC. In Proceedings of the IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 27–29 July 2016; pp. 1927–1931.
50. Ren, Z.; Fan, C.; Ming, Y. Music retrieval based on rhythm content and dynamic time warping method. In Proceedings of the IEEE International Conference on Signal Processing (ICSP), Chengdu, China, 6–9 November 2016; pp. 989–992.
51. Shao, Y.; Jin, Z.; Wang, D.L.; Srinivasan, S. An auditory-based feature for robust speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 19–24 April 2009; pp. 4625–4628.
52. Zhao, X.; Wang, D.L. Analyzing noise robustness of MFCC and GFCC features in speaker identification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–30 May 2013; pp. 7204–7208.
53. Neal, R.; Hinton, G. A view of the EM algorithm that justifies incremental, sparse and other variants. In *Learning in Graphical Models*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1998; pp. 355–368.
54. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
55. Lee, J.; Park, J.; Kim, K.; Nam, J. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. In Proceedings of the Sound & Music Computing Conference, Espoo, Finland, 5–8 July 2017; pp. 220–226.
56. Asch, V.V. *Macro- and Micro-Averaged Evaluation Measures*; CLiPS: Antwerp, Belgium, 2013; pp. 1–27.