Spectral–Temporal Receptive Field-Based Descriptors and Hierarchical Cascade Deep Belief Network for Guitar Playing Technique Classification

Chien-Yao Wang, Pao-Chi Chang, *Member, IEEE*, Jian-Jiun Ding, *Senior Member, IEEE*, Tzu-Chiang Tai, *Member, IEEE*, Andri Santoso, Yu-Ting Liu, and Jia-Ching Wang, *Senior Member, IEEE*

***Abstract*—Music information retrieval is of great interest in audio signal processing. However, relatively little attention has been paid to the playing techniques of musical instruments. This work proposes an automatic system for classifying guitar playing techniques (GPTs). Automatic classification for GPTs is challenging because some playing techniques differ only slightly from others. This work presents a new framework for GPT classification: it uses a new feature extraction method based on spectral–temporal receptive fields (STRFs) to extract features from guitar sounds. This work applies a supervised deep learning approach to classify GPTs. Specifically, a new deep learning model, called the hierarchical cascade deep belief network (HCDBN), is proposed to perform automatic GPT classification. Several simulations were performed and the datasets of: 1) data on onsets of signals; 2) complete audio signals; and 3) audio signals in a real-world environment are adopted to compare the performance. The proposed system improves upon the F-score by approximately 11.47% in setup 1) and yields an F-score of 96.82% in setup 2). The results in setup 3) demonstrate that the proposed system also works well in a real-world environment. These results show that the proposed system is robust and has very high accuracy in automatic GPT classification.**

***Index Terms*—Deep belief network (DBN), guitar playing technique (GPT) classification, neural network, spectral–temporal receptive fields (STRFs).**

I. Introduction

MUSIC plays an important role in entertainment. Instruments that are used to produce music include string woodwind, brass, piano, and many other instruments. String instruments, such as the guitar, are popular due to their flexibility. A variety of guitar playing techniques (GPTs) is used to produce unique music. Different techniques express different moods and feelings. The automatic classification of GPTs is challenging because some GPTs differ only slightly from others. A small variation of the GPT may lead to music sounds that are quite different.

Automatic GPT classification can be utilized in guitar transcription systems. Moreover, interactive computer-aided musical learning environments are helpful for improving musical skills [1]. An accurate GPT classification system can greatly enhance the effectiveness of an interactive guitar learning system.

Recently, music information retrieval (MIR) systems have been attracting increasing interest in the field of audio signal processing. The literature about MIR focused on key or pitch estimation [2], [3]; music source separation [4]–[6]; music genre classification [7]; the recognition of instruments [8]; and emotion recognition from music [9], [10]. Moreover, several new algorithms [11]–[14] were developed for the classification of GPTs. Automatic GPT classification is interesting because it is still in its early stage. Reboursiere *et al.* [11] presented a scheme for GPT recognition that was based on the successive classification of audio onsets. Su *et al.* [12] investigated the use of sparse coding (SC) to derive useful information from time-frequency representations of audio signals. Chen *et al.* [13] proposed a candidate selection method for classifying electric GPTs. While recent work focused on detecting the technique that is used to play a single note [12], Chen *et al.* [13] expanded this research by proposing an automatic classification of GPTs for solo guitar tracks. Su *et al.* [14] systematically evaluated various audio descriptors and studied the use of sparse modeling to obtain better audio descriptors for automatically classifying violin playing techniques. This work solved the problem of automatic GPT classification for both single notes and solo tracks and considered the following seven GPTs.

1) Normal, which is the basic GPT.
2) Muting, which is the GPT by which the sound is muted to create great attenuation and is generated by pressing the string with the right hand.
3) Vibrato, which is the technique in which a pulsating sound effect is produced by twisting the left-hand finger

Manuscript received September 9, 2018; revised July 12, 2020; accepted July 20, 2020. This article was recommended by Associate Editor B. W. Schuller. *(Corresponding author: Jia-Ching Wang.)*

Chien-Yao Wang is with the Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan.

Pao-Chi Chang and Yu-Ting Liu are with the Department of Communication Engineering, National Central University, Jhongli 320, Taiwan.

Jian-Jiun Ding is with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan.

Tzu-Chiang Tai is with the Department of Computer Science and Information Engineering, Providence University, Taichung 433, Taiwan.

Andri Santoso and Jia-Ching Wang are with the Department of Computer Science and Information Engineering, National Central University, Jhongli 320, Taiwan (e-mail: jcw@csie.ncu.edu.tw).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.3014207

on the string. This technique causes minute and rapid variations in the pitch.

4) Sliding, which is the technique in which the discrete change to the target note is applied by sliding a finger on the left hand over the frets without lifting it. It creates a smooth transition in pitch.
5) Hammer-on, which is the technique of hammering the strings using a finger on the left hand to create a sound that is similar to that created by the normal playing technique but with a smoother attack.
6) Pull-off, which is a technique that produces a sound that is similar to the sound of normal playing technique. However, a smoother attack is created by pulling the finger on the left hand off the string.
7) Bending, which is the bending of a string with fingers on the left hand without an apparent attack. It causes a continuous change in the target note.

In MIR research, various audio descriptors have been proposed to represent musical signal, such as Mel-frequency cepstral coefficients (MFCCs). Su *et al.* [12] utilized 41 descriptors in a GPT classification system, including the spectrogram, the logarithm cepstrum, the group delay function, instantaneous frequency deviation (IFD), and others. Recently, interest in biologically inspired systems were adopted in audio signal processing. Jeon and Juang [15] presented a system that mimics the human auditory system and applies it to process acoustic signal information. Chi *et al.* [16] described a mathematical model for analyzing the early and central stages of the human auditory system, called spectral–temporal response fields (STRFs), and utilized it to transform an acoustic signal into the multiresolution spectral–temporal representation. Wang *et al.* [17], [18] proposed an acoustic descriptor that is based on the STRF-based scale descriptor to recognize speech in a noisy environment.

When the audio descriptors for the guitar sound have been obtained, they are fed into the developed classifier to classify the GPTs. Numerous machine-learning methods have been proposed for solving the classification problem, including the Gaussian mixture model (GMM), the hidden Markov model (HMM), the support vector machine (SVM), and GMM-HMM [19]. In recent years, sparse representation and the deep learning framework became very popular methods for classification. Nam *et al.* [20] utilized the SC feature of musical audio data as an input to the SVM classifier. Humphrey *et al.* [21] used a deep neural network (DNN) framework, called the sparse restricted Boltzmann machine (sparse RBM), to classify SC features. Moreover, Grachten and Krebs [22] proposed RBM modeling expressive dynamics for music analysis. Other DNN frameworks, such as the deep belief network (DBN) [23] and the convolutional neural network (CNN) [24], are also attracting increasing interest in automatic audio signal recognition.

In this work, the STRF model is utilized to develop a new audio descriptor, called the STRF-based rate descriptor, to represent a guitar sound data signal. In the proposed system, the descriptors are obtained by computing the STRF-based scale descriptor and the STRF-based rate descriptor, and combining them with the well-known MFCC audio descriptor. Moreover, this work developed a new DBN framework, called the hierarchical cascade DBN (HCDBN) to classify GPTs automatically.

The main contributions of this work are as follows.

1) A new audio descriptor, called the STRF-based rate descriptor, is proposed.
2) A new framework for deep learning, called the HCDBN, is designed to solve the problem of GPT classification.
3) An advanced feature selection method for GPT classification was proposed.
4) The performance of the proposed approach much outperform those of many state-of-the-art techniques for audio processing.
5) The feasibility of applying the proposed system to solve the GPT classification problem in a real-world environment is examined.

This manuscript is organized as follows. In Section II, the overview of the proposed algorithm is given. In Section III, the proposed STRF-based rate descriptor and its extraction method are introduced. In Section IV, the proposed HCDBN architecture and its design and training methods are described. In Sections V and VI, the experimental setup and the experiment results are presented. A conclusion is given in Section VII.

II. Overview of the Proposed Work

This work proposes a system for automatically classifying GPTs. The inputs to the system are music audio data that are professionally recorded from an electric guitar [12]. The robustness of the system is evaluated using guitar sound data in a GPT dataset recorded in a real-world environment. Fig. 1 presents an overview of the proposed system.

The first step in classifying GPT is to separate guitar music data into numerous clips to be fed into the system. Each clip of guitar music represents a single guitar playing segment. Since the guitar is a plucked string instrument, there is always an onset at the beginning of a musical note. Therefore, one can apply onset detection on the guitar music signal and treat the interval between two onsets as a musical note. Then, one can analyze the GPT within the interval. The onset detection algorithm, introduced by Kehling *et al.* [25], is adopted. The results of which are input to the audio descriptor extraction stage, wherein several audio descriptors, including the STRF-based scale descriptor, the STRF-based rate descriptor, and the conventional MFCC, are applied.

The proposed STRF-based descriptors have two main parts. The first part is the STRF-based scale descriptor, which we developed previously [17]. The present work further develops a new STRF-based descriptor, called the STRF-based rate descriptor. Both the STRF-based scale descriptor and the STRF-based rate descriptor are utilized to construct the final audio descriptor of guitar musical sound data. The final descriptor of every clip is then represented by a concatenation of the proposed STRF-based audio descriptors with the conventional MFCC.

The extracted audio descriptors are fed into the proposed HCDBN framework to perform feature learning and GPT classification. The proposed HCDBN framework is utilized to

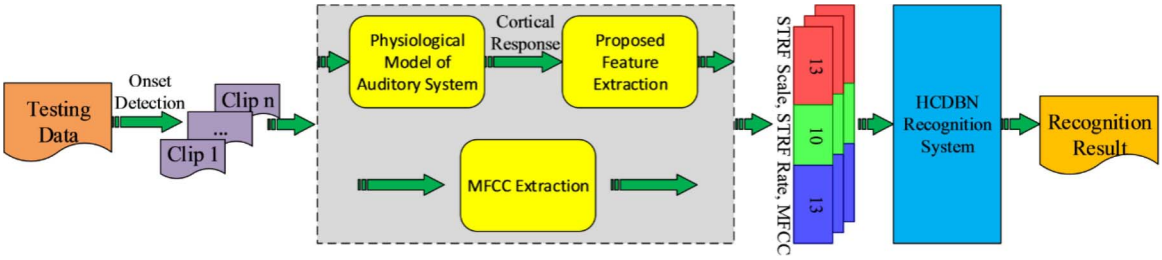


Fig. 1. Proposed system framework.

simulate the functions of human brain by a hierarchical feature learning model. It makes the proposed framework work analogous to the entire human hearing system.

III. Proposed STRF-Based Descriptor Extraction

A. Spectral–Temporal Receptive Fields

When a human being hears a sound, an audio signal is passed through the hearing system. The auditory spectrum is processed by the auditory cortex in the human brain. The STRF [16] is a mathematical model that simulates the two stages of the human hearing system. It comprises the following two models.

1) *Simulating the Early Stage of the Human Hearing System:* The STRF model simulates the first stage of the human hearing system that operates when a human hears a sound. The main functions of this step are to model the cochlea and to generate an auditory spectrogram from the received sound.
2) *Simulating Primary Auditory Cortex (A1):* A mathematical model is utilized to simulate the process in the primary auditory cortex (A1) layer of the human hearing system. The main functions of this step are to perform a multiresolution transformation on the spectral–temporal domain and to capture the formants, harmonics, vocal track, and pitch characteristics.

The first step in simulating the human hearing system is to perform an affine wavelet transform by 128 band-pass filters, whose central frequencies are uniformly distributed along a logarithmic frequency axis. Then, the cochlear output is obtained from

$$y(t,f) = \max\left(\partial_f g(\partial_t y_C(t,f)) *_t \omega(t), 0\right) *_t e^{\frac{-t}{\tau}} u(t) \quad (1)$$

where $y_C(t,f) = s(t) *_t h(t,f)$, which is analogous to the cochlear output, $s(t)$ is the input signal, $h(t, f)$ are the impulse responses of the 128 band-pass filters, and f denotes the central frequency of each band-pass filter. The operator $*_t$ means the convolution operation in the time domain, $g(\,)$ is a nonlinear compression function, $\omega(t)$ denotes a low-pass filter, and $u(t)$ is a unit step function.

To simulate the primary auditory cortex (A1), we apply

$$\mathrm{STRF} = h_S \cdot h_T \quad (2)$$

where h_S denotes the spatial impulse response and h_T denotes the temporal impulse response. For an input of the auditory spectrogram $y(t, f)$, the spectral–temporal response $\mathrm{STRF}(t, f, \Omega, \omega, \varphi, \theta)$ is calculated from

$$\mathrm{STRF}(t,f,\Omega,\omega,\varphi,\theta) = y(t,f) *_{tf} \left[h_S(f,\omega,\theta) \cdot h_T(t,\Omega,\varphi)\right] \quad (3)$$

where $*_{tf}$ means the convolution of the $t-f$ plane, Ω and ω are the spatial density and velocity parameters of the filters, and φ and θ are characteristic phases. $h_S(f, \omega, \theta)$ and $h_T(t,\Omega, \varphi)$ are the spatial impulse response (in cycle/octave) and temporal impulse response (in Hz), respectively

$$h_S(f,\omega,\theta) = h_{\mathrm{scale}}(f,\omega)\cos\theta + \hat{h}_{\mathrm{scale}}(f,\omega)\sin\theta \quad (4)$$

$$h_T(t,\Omega,\varphi) = h_{\mathrm{rate}}(t,\Omega)\cos\varphi + \hat{h}_{\mathrm{rate}}(t,\Omega)\sin\varphi \quad (5)$$

where $h_{\mathrm{scale}}(f)$ and $h_{\mathrm{rate}}(t)$ are approximated by a Gaussian function and a Gamma function, respectively, and $\hat{h}(f)$ denotes the result of the Hilbert transform function.

The scale parameter represents the width of an auditory spectrogram energy that is distributed along the frequency axis. The rate parameter quantifies the velocity at which the spectrogram energy varies along the temporal axis. It can be separated into two directions: 1) the downward rate (positive value) and 2) the upward rate (negative value). Fig. 2 shows examples of the upward moving rate and the downward moving rate.

B. STRF-Based Scale Descriptor

The formants and the harmonics of audio signals can be represented using the STRF descriptor with low-scale and high-scale parameters, respectively.

The first step of our previously proposed STRF-based scale descriptor [17] is to obtain $S(t, \omega)$ by summing the magnitudes of the STRF representations for all f and Ω

$$S(t,\omega) = \sum_f \sum_\Omega |\mathrm{STRF}(t,f,\Omega,\omega,0,0|, \omega = 1,2,\ldots,N_\omega \quad (6)$$

where N_ω is the scale number.

Next, a logarithmic function is used to calculate S_L

$$S_L(t,\omega) = \log(S(t,\omega)). \quad (7)$$

Then, an N_k-point discrete cosine transform (DCT) [26] is performed on $S_L(t, \omega)$, where N_k is the feature dimension

$$S_{DL}(t,k) = \sum_{\omega=1}^{N_\omega} S_L(t,\omega)\cos\left(\frac{2\pi\omega k}{N_\omega}\right), \quad k = 1,2,\ldots,N_k. \quad (8)$$

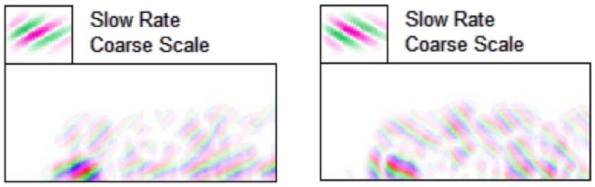


Fig. 2. (a) Upward and (b) downward moving rate. The x-axis is time; and the y-axis is frequency.

TABLE I
FREQUENCY RANGE SETTING FOR FEATURE EXTRACTION

Index Number	Frequency Range (Hz)
1	0-500
2	501-1000
3	1001-2000
4	2001-4000
5	4001-8000

C. STRF-Based Rate Descriptor

Following the STRF-based scale descriptor, the STRF-based rate descriptor is also proposed. The latter one is used to capture the characteristic of rate parameters over a period of $2m + 1$ frames. The first process in the STRF-based rate descriptor is given by

$$R_t(f, \Omega) = \frac{1}{2m+1} \sum_{i=t-m}^{t+m} \sum_{\omega} |\mathrm{STRF}(i, f, \Omega, \omega, 0, 0)|. \quad (9)$$

Equations (10) and (11) yield the positive rate $R_t{}^+$and the negative rate $R_t{}^-$, respectively

$$R_t^+(j) = \sum_{L_j \le f \le H_j} \sum_{\forall \Omega > 0} R_t(f, \Omega) \quad (10)$$

$$R_t^-(j) = \sum_{L_j \le f \le H_j} \sum_{\forall \Omega < 0} R_t(f, \Omega) \quad (11)$$

where j is the frequency range index, and L_j and H_j, respectively, represent the lower and the higher frequencies in the jth frequency range.

Based on the concept of the critical band [27], this work extracts features of audio data in certain frequency ranges, which are shown in Table I. The audio frequency is split into five ranges based on the frequency selectivity of the human hearing system. Better resolution is used for the low-frequency part so that the low-frequency part can be emphasized.

Finally, the total rate $R_t(j)$ can be obtained from a combination of the positive rate $R_t{}^+(j)$ and the negative rate $R_t{}^-(j)$ together with a logarithmic function

$$R_t(j) = \left[R_t^+(j), R_t^-(j)\right] \quad (12)$$

$$R_t^L(j) = \left[\log\left(R_t^+(j)\right), \log\left(R_t^-(j)\right)\right]. \quad (13)$$

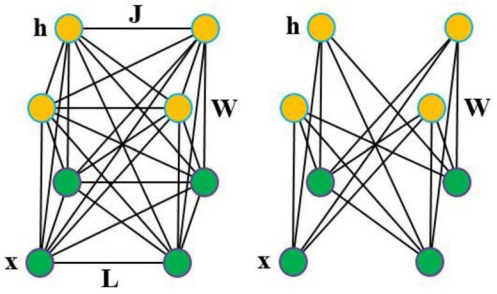


Fig. 3. Basic unit of DBN. Left: BM model; right: RBM model.

IV. HIERARCHICAL CASCADE DEEP BELIEF NETWORK

A. Deep Belief Network

Hinton and Salakhutdinov [28] introduced the framework of a DBN, which has recently become one of the most popular frameworks in machine learning—especially in deep learning. The DBN is constructed from several organized restricted Boltzmann machines (RBMs). The RBM model, presented in Fig. 3, is a generative model. It is a modified version of the Boltzmann machine (BM).

The energy functions for the BM and the RBM are shown in (14) and (15), respectively

$$E(\mathbf{x}, \mathbf{h}) = -\tfrac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{L}\mathbf{x} - \tfrac{1}{2}\mathbf{h}^{\mathbf{T}}\mathbf{J}\mathbf{h} - \mathbf{x}^{\mathbf{T}}\mathbf{W}\mathbf{h} \quad (14)$$

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{x}^{\mathbf{T}}\mathbf{W}\mathbf{h} - \mathbf{a}^{\mathbf{T}}\mathbf{x} - \mathbf{b}^{\mathbf{T}}\mathbf{h} \quad (15)$$

where $\mathbf{x}$ is the input unit, $\mathbf{h}$ is the hidden unit, $\mathbf{L}$ is the weight among input units, $\mathbf{J}$ is the weight among hidden units, $\mathbf{W}$ is the weight between the input layer and the hidden layer, $\mathbf{a}$ is the bias of the input layer, and $\mathbf{b}$ is the bias of the hidden layer. Then, the joint probability $P(\mathbf{x}, \mathbf{h})$ of the energy function is defined as

$$P(\mathbf{x}, \mathbf{h}) = \tfrac{1}{Z} e^{-E(\mathbf{x}, \mathbf{h})} \quad (16)$$

$$\text{where } Z = \sum_{\mathbf{x}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}. \quad (17)$$

Then, the RBM is applied to maximize $P(\mathbf{x})$ in (18)

$$P(\mathbf{x}) = \tfrac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})}. \quad (18)$$

Mnih and Hinton [29] presented the contrastive divergence (CD) algorithm, which is shown as follows, to perform RBM learning:

$$\Delta w_{ij} = \varepsilon\left(\langle x_i h_j \rangle_{\text{data}} - \langle x_i h_j \rangle_{\text{recon}}\right). \quad (19)$$

The framework of the RBM can also be utilized to perform discrimination and is then referred to as a discriminative RBM framework. The joint probability of the discriminative RBM $P(\mathbf{y}, \mathbf{x}, \mathbf{h})$ is given as follows:

$$P(\mathbf{y}, \mathbf{x}, \mathbf{h}) = \tfrac{1}{Z} e^{-E(\mathbf{y}, \mathbf{x}, \mathbf{h})} \quad (20)$$

$$\text{where } Z = \sum_{\mathbf{y}, \mathbf{x}, \mathbf{h}} e^{-E(\mathbf{y}, \mathbf{x}, \mathbf{h})} \quad (21)$$

and $\mathbf{y}$ is the label of data. The goal of the discriminative RBM is to maximize the joint probability $P(\mathbf{y}, \mathbf{x})$, which is given as

$$P(\mathbf{y}, \mathbf{x}) = \tfrac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{y}, \mathbf{x}, \mathbf{h})}. \quad (22)$$

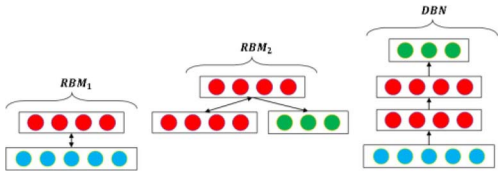


Fig. 4. Network architecture of the DBN. The DBN is formed by cascading several generative RBMs and a discriminative RBM. The blue part denotes the input, the red part denotes the neuron of the hidden layer, and the green part denotes the label.

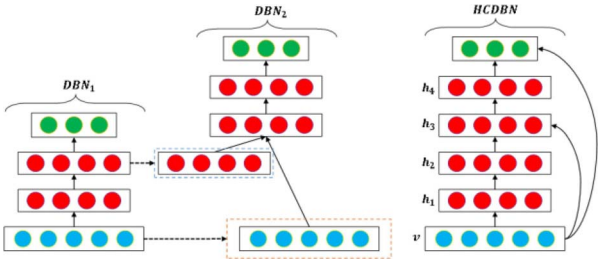


Fig. 5. Network architecture of the HCDBN. The HCDBN is formed by cascading several DBNs. The input layer of the ith DBN is connected to the input layer of the first DBN. The blue part denotes the input, the red part denotes the neuron of the hidden layer, and the green part denotes the label.

Discriminative RBM can be trained using a CD algorithm as described above, or using another discriminative algorithm, such as the gradient descent algorithm. In the DBN framework, RBMs perform layerwise pretraining to initialize the weights of the DBN. A DBN is composed of layerwise pretrained RBMs, as presented in Fig. 4. In the HCDBN herein, the Softmax with the cross-entropy layer is utilized and added to the top of the architecture as another layer for fine-tuning.

B. HCDBN Architecture

In this work, the proposed HCDBN framework comprises several layerwise fine-tuned DBNs, as in Fig. 5. Its main purpose is to solve the problem of information loss that may occur in a high layer DBN when the layerwise fine-tuning process is carried out in the architecture of the cascade DBN framework.

The design of the proposed HCDBN considers the gap between layerwise unsupervised pretraining and layerwise supervised learning. Since every DBN layer in the HCDBN framework performs layerwise fine-tuning, some properties of the original input data are lost during discrimination learning. Therefore, in the proposed HCDBN, an additional connection from the input layer to some high layers of the HCDBN is defined. For example, in Fig. 5, the layer v is connected to the layer h_3, ensuring not only that every layer of the DBN maintains its original properties during training but also that both low-level information and high-level information are learned in an integrated fashion.

In this work, a pretraining stage that involves the discriminative RBM or the generative RBM for hidden layers is utilized. Since the input signal in the experiment is a real-valued vector, the energy function of a Gaussian–Bernoulli RBM, given by $(1/2)\mathbf{x}^2 - \mathbf{x}^\mathbf{T}\mathbf{W}\mathbf{h} - \mathbf{a}^\mathbf{T}\mathbf{x} - \mathbf{b}^\mathbf{T}\mathbf{h}$, is used to model the connection between the input layer and the hidden layer of the HCDBN architecture.

Then, the energy function of the Bernoulli–Bernoulli RBM given by $-\mathbf{g}^\mathbf{T}\mathbf{W}\mathbf{h} - \mathbf{a}^\mathbf{T}\mathbf{g} - \mathbf{b}^\mathbf{T}\mathbf{h}$ is utilized to model the connection between hidden layers without the income from the input layer. Finally, the energy function of the hybrid RBM, given by $(1/2)\mathbf{x}^2 - [\mathbf{x}, \mathbf{v}]^\mathbf{T}\mathbf{W}\mathbf{h} - \mathbf{a}^\mathbf{T}[\mathbf{x}, \mathbf{v}] - \mathbf{b}^\mathbf{T}\mathbf{h}$, is used to model the connection from the input layer to a particular layer of the HCDBN architecture. This particular layer is identified as the visible layer. Here, $\mathbf{x}$ is the input data, $\mathbf{v}$ is the visible layer, $\mathbf{g}$ and $\mathbf{h}$ are the hidden layers, $\mathbf{W}$ is the weight, $\mathbf{a}$ is the bias of the input data and the visible layer, and $\mathbf{b}$ is the bias of the hidden layer.

After the discriminative RBM has been trained, a Softmax with the cross-entropy loss layer is added as the first stage of the deep learning architecture to perform fine-tuning. Another layer, the Hinge loss layer, is then added to enhance the discrimination ability of the Softmax with the cross-entropy loss layer by maximizing the margin among the extracted audio descriptors of the GPT data from different classes.

The discriminative ability of the HCDBN framework arises from the fact that each DBN can maintain the information of the input data during the hierarchical training process. To maintain this information, we connect the input layer to some higher layers in the HCDBN framework. As in Fig. 5, the proposed HCDBN is constructed from several DBNs, in which each layer is connected to the next layer and some connections are from the input layer to the high-level layers in the HCDBN.

C. Weight Initialization

Typically, a deep neural framework initializes the weights of a network randomly or according to a particular distribution, Some deep learning method uses the Gaussian distribution described in (23) with a fixed standard deviation to initialize the weight of a network [30]

$$W_{i,j} \sim N(0, \sigma) \tag{23}$$

where σ is a standard deviation, such as 0.001. He *et al.* [31] and Simonyan and Zisserman [32] proposed a random weight initialization method that is based on the number of neurons in the network. Glorot and Bengio [33] presented a weight initialization method that assumes that the activation function is linear as

$$W_{i,j} \sim U\left(\frac{-1}{\sqrt{n_1}}, \frac{1}{\sqrt{n_1}}\right). \tag{24}$$

He *et al.* [31] proposed an initialization method and defined the parametric rectified linear unit (PReLU) activation function as

$$W_{i,j} \sim N\left(0, \tfrac{2}{n_l}\right) \tag{25}$$

where n_l is the number of neurons in layer l.

A traditional DBN uses a logistic function as its activation function, which is unlike the activation function described above. Therefore, a new initialization method for the proposed HCDBN framework that uses various activation functions for various layer connections is required. Specifically, the architecture of the HCDBN is separated into the following five subarchitectures.

1) The Gaussian–Bernoulli RBM, whose input layer is connected to a higher hidden layer.

2) The Bernoulli–Bernoulli RBM, in which a hidden layer is connected to a higher hidden layer.
3) The Hybrid RBM, whose input layer is connected to a high-level layer.
4) The top layer of the network, which is the Softmax with the cross-entropy loss layer, for primary fine-tuning.
5) The final layer of the network, which is the Hinge loss layer, for use in the last fine-tuning step.

Different initialization methods are applied to different subarchitectures based on its characteristics. The initialization method in (25) is utilized to initialize the network weight in subarchitectures 2) and 3). The initialization method in (23) is used to initialize the weight of subarchitectures 2) and 4). Finally, the initialization method in (24) is utilized to initialize the weight of subarchitecture 5).

D. Loss Function

The Softmax function as follows is commonly used in the output layer of the DBN:

$$l(\mathbf{y}, \mathbf{z}) = \frac{e^{z_{\mathbf{y}}}}{\sum_{i=1}^{C} e^{z_i}}. \tag{26}$$

Its objective is to transform the prediction results of the DBN into a probability. Moreover, in this work, an additional layer on the top of the Softmax layer is utilized as the final layer of the neural-network architecture. This final layer is the Hinge loss layer, whose loss function is given by

$$l(\mathbf{y}, \mathbf{z}) = \max(0, 1 - \mathbf{y} \cdot \mathbf{z}) \tag{27}$$

where $\mathbf{z}$ is a decision function, which satisfies $\mathbf{z} = \mathbf{wx} + \mathbf{b}$ for the linear classifier and $\mathbf{z} = k(\mathbf{w}, \mathbf{x})$ for the nonlinear classifier, and k is a kernel function. The term $\mathbf{y}$ is the target value and C is the number of classes.

The advantage of utilizing the Softmax with the cross-entropy loss layer in the first stage of fine-tuning and adding the Hinge loss layer in the subsequent fine-tuning stage is to maximize the margin between different guitar music data from different classes, and thereby to improve the discriminative ability of the framework. The Hinge loss function is used to maximize the margin between different guitar music data from different classes. Therefore, the data classified into the margin region probably contain the characteristics owned by different classes, this approach helps the classifier to process any silent part of the sound data.

V. Experimental Setup

This section describes the experimental setup for evaluating the performance of the proposed framework. The GPT dataset from the work of Su *et al.* [12] was utilized. This dataset comprises seven playing techniques of the electrical guitar that is composed of 19 subclasses of GPT. Table II describes the GPT database used in the experiments. There are two sets of data: 1) a split dataset, which includes data on the onsets of sounds and only portions of the waveform signals, obtained by clipping them from 0.1 s before the onset to 0.2 s after the onset and 2) a complete dataset, which includes complete audio signals of guitar sounds.

TABLE II
Number of Sound Clips in the GPT Database

7 classes	19 sub-classes in 7 main classes	Split	Complete
normal	normal normal half step down normal half step up normal complete step down normal complete step up	3659	1974
muting	mute	367	378
vibrato	trill	622	630
pull-off	pulling half step pulling complete step	515	511
hammer-on	hamming half step hamming complete step	559	567
sliding	slide half step down slide half step up slide complete step down slide complete step up	1061	1134
bending	bending up down half bending up down complete bending up half bending up complete	874	1253

The experimental settings of Su *et al.* [12] were used to perform five-fold cross-validation. The performance of the system is evaluated using the mean normalized F-score across seven main playing techniques. The performance of the proposed HCDBN framework is compared with that of the DBN framework proposed by Keyvanrad and Homayounpour [34]. Each HCDBN and hierarchical advanced DBN (HADBN) contains two advanced DBNs (ADBNs), and each ADBN and DBN has two hidden layers, with every hidden layer having 300 hidden units. In the input layer, the number of parameters is given by *input_feature_dim**300; in the hidden layer, the number of parameters is given by 300*300; and in the output layer, the number of parameters is given by 300**number_of_classes*. After RBM stacking, all models undergo 2000 iterations of backpropagation for fine-tuning. The experimental results of Su *et al.* [12] are taken as the baseline, and those obtained for the DBN scheme of Keyvanrad and Homayounpour [34] are labeled as "DBN" results.

We proposed not only the HCDBN framework but also other improved versions of the DBN framework, including the ADBN and the HADBN. The architecture of the ADBN is similar to that of the traditional DBN but with the improvements of the proposed hybrid weight initialization algorithm, (see Section IV-C), and the proposed training scheme that uses the Hinge loss function (see Section IV-D). The HADBN framework is utilized to evaluate a new strategy for fine-tuning the neural network. The HADBN hierarchically cascades the network of the DBN to fine-tune the neural network and considers the data from both the input layer and the learned feature. The two improvement methods of the ADBN framework are also applied to the HADBN framework. Experiments are performed to confirm the performances of the ADBN and the HADBN.

To evaluate the performance of the proposed audio descriptors, several descriptors, such as the traditional MFCC, the

TABLE III
RESULTING AVERAGE F-SCORES USING THE SPLIT DATASET (SEVEN MAIN CLASSES)

Split/7	DBN [34]	ADBN	HADBN	HCDBN
MFCC13	66.91%	69.74%	69.02%	69.93%
S	69.75%	71.76%	72.67%	72.12%
S + R	76.00%	77.56%	78.83%	77.04%
MFCC13+ S+R	72.64%	78.86%	78.36%	80.23%

STRF-based scale (S) descriptor, the STRF-based rate (R) descriptor, and a concatenation of those three feature descriptors (MFCC+S+R), are examined. The evaluation results are reported in terms of F-score at the clip-level, and the F-score is the harmonic mean of precision and recall which is calculated by

$$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{28}$$

VI. EXPERIMENTAL RESULTS

This section describes five experiments that are performed to evaluate the proposed framework. They are as follows.

1) Evaluating the performance of both the proposed STRF-based descriptor and the proposed HCDBN framework using the settings of Su *et al.* [12] with the split dataset.
2) Analyzing the behavior of audio signals in the internote experiment, where musical audio clips from a GPT class are compared with audio clips from other classes.
3) Verifying the performance of the proposed HCDBN framework using complete audio clip signals from the dataset.
4) Analyzing the results of the intranote experiment to compare musical audio clips from a GPT class with audio clips associated with the same GPT.
5) Evaluating the performance of the proposed system to solve the GPT classification problem in the real-world environment.

Each of the five experiments will be described in the following sections.

A. Results Using the Split Dataset

First, the performances of the proposed STRF-based descriptor and the proposed HCDBN framework are evaluated in a split dataset experiment. The audio descriptors MFCC13, S, S + R, and MFCC13 + S + R, are used. Table III presents the performances (in terms of F-scores) of all configurations of the audio descriptor and the DBN framework. The audio descriptors S, S+R, and MFCC13+S+R outperform the traditional MFCC13 descriptor with F-scores that are 2.84%, 9.09%, and 5.73% better, respectively. These results demonstrate that the proposed STRF-based audio descriptors provide a high F-score in the split dataset. The proposed audio descriptors capture the transitions of pitches or harmonics in audio clips and are useful for extracting features in the onset parts of the clips.

Moreover, Table III shows that the proposed frameworks of the ADBN, the HADBN, and the HCDBN all outperform the DBN with all audio descriptors. When the audio descriptor of MFCC13 + S + R is applied, the F-scores of the ADBN, the HADBN, and the HCDBN are 6.22%, 5.72%, and 7.59% better than that of the DBN. The HCDBN framework yields the best F-score of 80.23% when the MFCC13+S+R audio descriptor is used. The proposed audio descriptors and the proposed HCDBN architecture totally improve the average F-score of the DBN by 13.32%.

B. Observation 1: Internote Behavior of GPT

In this section, the behavior of waveform signals of split musical data associated with various GPTs is analyzed. Such analysis is called internote analysis. Since the technique that is used to play a musical instrument depends on the temporal information, the variance of the spectrogram on the time scale, which is shown in (29), is used to analyze the internote behavior of the GPT musical data

$$\text{Var}(t) = \frac{1}{2n} \sum_{f} \sum_{i=t-n}^{t+n} \left(x_{t,f} - x_{i,f}\right)^2 \tag{29}$$

where t is time; f is frequency, x is the spectrogram of the audio data, and n is the number of frames. The $(t-n)$th frames to the $(t+n)$th frames are used in the analysis of the t-th frame.

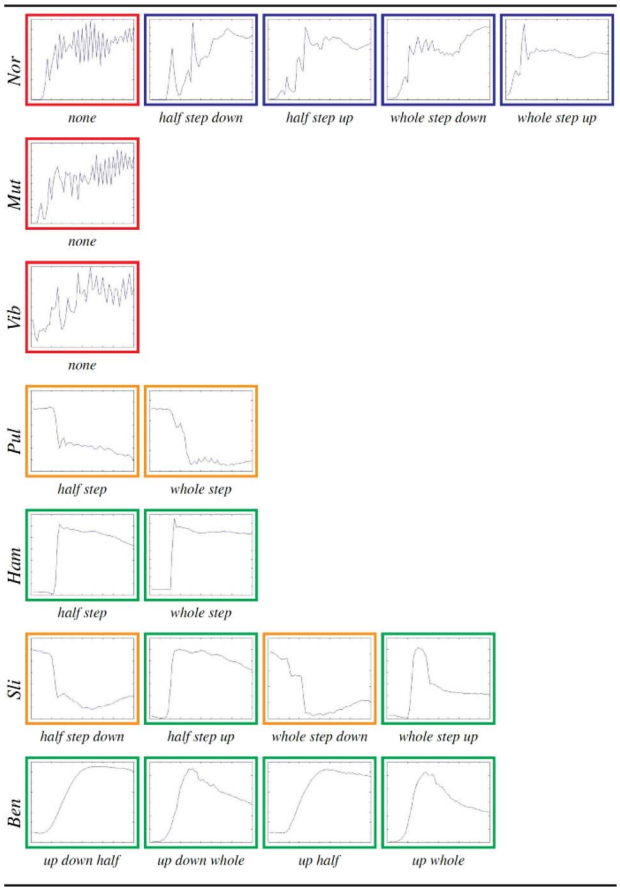
Table IV plots the graph of spectrogram variances that are associated with seven GPTs in different classes obtained using the split dataset. Frames of the same color are strongly similar. As shown in Table IV, even though some graphs are of different GPT classes, they exhibit very similar variations along the time axis. Some graphs associated with normal, mute, and vibrato techniques (red frames) are very similar to one another. Some graphs associated with hammer-on, sliding, and bending techniques look very similar (green frames). Moreover, the graphs associated with the pull-off and sliding techniques are also very similar (orange frames).

Some GPTs are hard to differentiate at the onset. For example, normal, mute, and vibrato have similar onsets. To differentiate them, the information after the onset is required. For the GPT of "normal," the signal remains unchanged after the onset. For "mute" and "vibrato," due to fretting and picking on the string, the signal changes significantly. Therefore, if the proposed STRF-based features are applied only on the onset part, the performance of GPT classification may be limited.

Therefore, as presented in Fig. 6, using the variation trend of the spectrogram can only separate the GPT into two groups—[normal, mute, vibrato] and [pull-off, hammer-on, sliding, bending], even there are actually seven classes. To provide a deeper understanding of the internote behavior, Table V presents the confusion matrix of audio clips from various GPT classes on the split dataset.

Tables IV and V and Fig. 6 indicate that using a split dataset is challenging because some data associated with different GPT classes are not well distinguished. As will be explained in the following section, using the complete audio clip can improve the recognition result of the system.

TABLE IV
INTERNOTE ANALYSIS OF THE SPLIT DATASET. EACH PLOT REPRESENTS THE VARIANCE IN THE SIGNALS FOR DIFFERENT GPTS, WHICH IS COMPUTED BY (29). THE PLOTS ALONG EACH ROW DENOTE THE SIGNAL VARIANCE OF DIFFERENT SUBCLASSES IN EACH GPT ON THE TIME AXIS



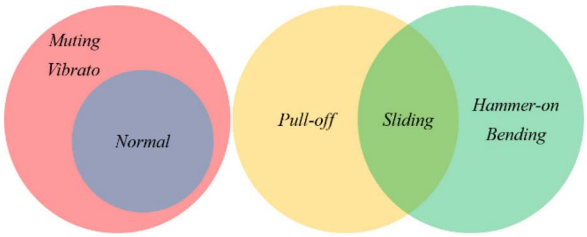


Fig. 6. Visualization of variance on the split dataset. There are four different variation trends in Table IV, including red, blue, orange, and green frames. Normal clips contain red and blue frames; muting clips contain red frames; vibrato clips contain red frames; pull-off clips contain orange frames; hammer-on clips contain green frames; sliding clips contain orange and green frames; and bending clips contain green frames.

C. Results Obtained Using the Complete Dataset

The experiments in this section evaluate the performance of the proposed framework using the complete dataset. The experiments were performed with three audio descriptors, which were MFCC13, MFCC13 + S, and MFCC13 + S + R. Table VI shows the results concerning the performance of the proposed system using the complete dataset. As expected, the F-score that was obtained using the complete audio clip signal exceeds that obtained using the split dataset. In the split dataset, the highest F-score is 80.23%. In contrast, when the complete audio clip signal is applied, the highest F-score is 93.38%, which was obtained when the HADBN framework and the MFCC13+S descriptor were used. Using the complete dataset increased the average F-score by 13.15%.

TABLE V
CONFUSION MATRIX BASED ON THE SPLIT DATASET OF SEVEN MAIN CLASSES

	nor	*mut*	*vib*	*pul*	*ham*	*sli*	*ben*
nor	-	0.14	2.60	0.96	0.41	0.14	0.14
mut	4.11	-	2.74	-	-	-	-
vib	37.90	0.81	-	-	-	-	0.81
pul	4.85	-	-	-	3.88	2.30	1.94
ham	0.90	-	0.88	8.11	-	10.81	3.60
sli	0.94	-	0.44	9.91	13.68	-	11.32
ben	1.15	-	-	-	2.30	9.20	-

TABLE VI
RESULTING AVERAGE F-SCORES USING THE COMPLETE DATASET (SEVEN MAIN CLASSES)

Complete/7	ADBN	HADBN	HCDBN
MFCC13	93.26%	93.27%	92.83%
MFCC13+S	90.00%	93.38%	89.25%
MFCC13+S+R	88.40%	92.74%	92.62%

D. Observation 2: Intranote Behavior of GPT

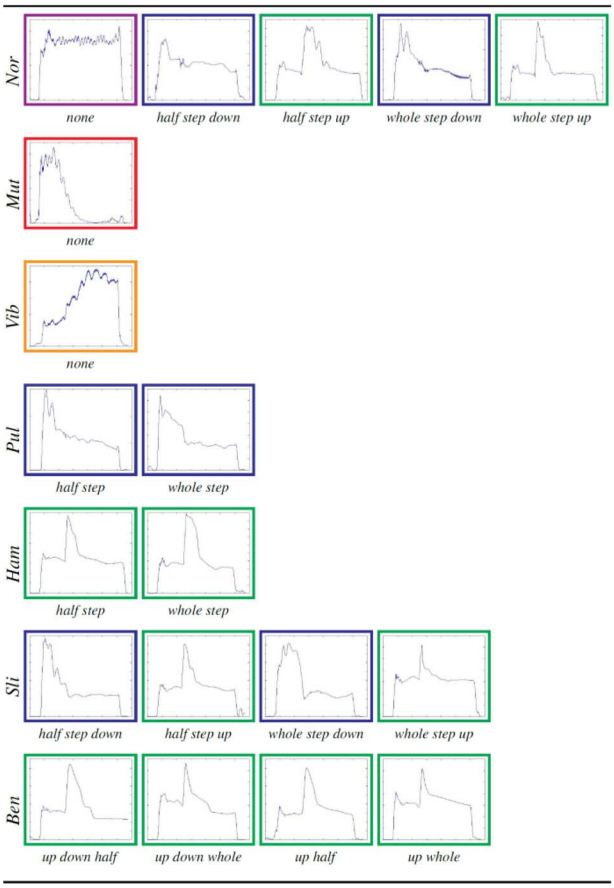
In this section, the behavior of sound clips of the complete GPT musical data associated with the same class is analyzed. Such analysis is called intranote analysis. The GPTs in the dataset comprise 19 subtechniques, as mentioned in Table II. We utilize information about these 19 subtechniques to elucidate the intranote behavior.

Table VII shows the variances of the spectrograms of distinct audio clips associated with seven GPTs and 19 subtechniques using the complete dataset. From Table VII, one can see that the differences among the spectrogram variances for different GPTs or subtechniques are more obvious than those in Table IV. In Table VII, the subtechniques of step-up and step-down produce quite different variation trends. Table VIII presents the confusion matrix corresponding to the complete dataset. It shows that confusion arises only in a small number of cases.

Fig. 7 shows a visualization of the complete dataset. It reveals that, with the complete dataset, the GPTs can be easily separated into three groups—[mute], [vibrato], and [normal, pull-off, hammer-on, sliding, bending]. Almost all classification errors in Table VIII match the areas of confusion in Fig. 7.

Compared to the internote analysis of the split dataset, the intranote analysis of the complete dataset reveals the global variation of a GPT data signal, whereas the split dataset yields a local variation. Considering global and local information

TABLE VII
INTRANOTE ANALYSIS OF THE COMPLETE DATASET. EACH PLOT REPRESENTS THE VARIANCE IN THE SIGNALS FOR DIFFERENT GPTS, WHICH IS COMPUTED BY (29). THE PLOTS ALONG EACH ROW DENOTE THE SIGNAL VARIANCE OF DIFFERENT SUBCLASSES IN EACH GPT ON THE TIME AXIS



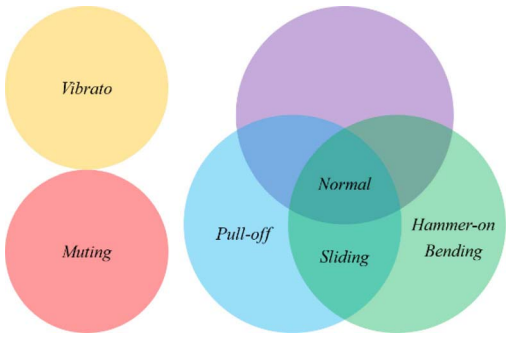


Fig. 7. Visualization of variance on the complete dataset. There are five different variation trends in Table VII, including red, orange, purple, blue, and green frames. Normal clips contain purple, blue, and green frames; muting clips contain red frames; vibrato clips contain orange frames; pull-off clips contain blue frames; hammer-on clips contain green frames; sliding clips contain blue and green frames; and bending clips contain green frames.

simultaneously can separate seven GPT class better than considering only global or local information, as in Fig. 8.

Given the audio data associated with 19 subtechniques, another approach to GPT classification is developed. It first identifies the subtechniques from audio data and then maps the result of that identification onto seven main GPTs. The following section considers this alternative approach.

TABLE VIII
CONFUSION MATRIX BASED ON THE COMPLETE DATASET OF SEVEN MAIN CLASSES

	nor	*mut*	*vib*	*pul*	*ham*	*sli*	*ben*
nor	-	-	1.52	0.76	0.51	1.27	0.51
mut	-	-	-	-	-	-	-
vib	-	-	-	-	-	-	0.7
pul	2.49	-	-	-	-	4.90	-
ham	0.88	-	0.88	-	-	1.77	5.31
sli	1.33	-	0.44	0.44	2.21	-	1.33
ben	-	-	-	-	0.40	0.80	-

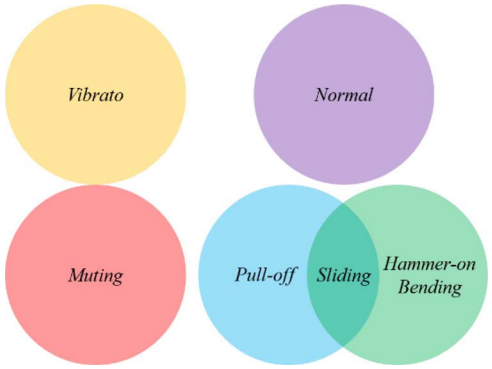


Fig. 8. Visualization of variance on the mixture of the split and the complete datasets. Normal can be separated from pull-off, sliding, hammer-on, and bending when considering global and local information simultaneously.

E. Overall Comparison

The experiments in this section compares the performance of the proposed system with those of the baseline systems in [12] and the DBN in [34]. The baseline system in [12] used 41 kinds of features to perform SC and utilized the SVM as a classifier. Beyond directly classifying seven main GPTs, the proposed three frameworks utilize an indirect strategy of classifying guitar audio data into 19 subtechniques and then mapping the recognition result onto seven main GPTs. Following the comparison in the preceding experiments, two datasets are used in this experiment: 1) the split dataset and 2) the complete dataset.

Table IX compares the performances of the proposed systems (ADBN, HADBN, and HCDBN with the descriptors of S+R or MFCC13+S+R) with that of the baseline system [12] and the DBN [34] using the split dataset. In the framework names, the number 7 refers to that the system classifies the musical data into seven main GPTs and 19 refers classifying the musical data into 19 subtechniques.

Table IX shows that, for the split dataset, the proposed ADBN-7 framework yields an F-score of 78.86%, the HADBN-7 yields an F-score of 78.83%, and the HCDBN-7 yields the highest F-score of 80.23%. The average F-score of the baseline system of [12] is 68.76%. All three proposed systems are more than 10% more accurate than the baseline system. The F-scores of the proposed ADBN, HADBN, and HCDBN systems are about 12% better than that of the DBN.

TABLE IX
RESULTING AVERAGE F-SCORES OF GPT CLASSIFICATION USING THE SPLIT DATASET

Scheme	Feature	Avg. F-score
Baseline-7 [12]	SC	68.76%
DBN-7 [34]	MFCC13	66.91%
ADBN-7	MFCC13+S+R	78.86%
ADBN-19	S+R	79.16%
HADBN-7	S+R	78.83%
HADBN-19	S+R	76.42%
HCDBN-7	MFCC13+S+R	80.23%
HCDBN-19	S+R	79.20%

TABLE X
RESULTING AVERAGE F-SCORES OF GPT CLASSIFICATION USING THE COMPLETE DATASET

Scheme	Feature	Avg. F-score
DBN-7 [34]	MFCC13	85.71%
ADBN-7	MFCC13	93.26%
ADBN-19	MFCC13	94.50%
HADBN-7	MFCC13+S	93.38%
HADBN-19	MFCC13+S+R	96.81%
HCDBN-7	MFCC13	92.83%
HCDBN-19	MFCC13+S+R	96.82%

Table X compares the performances of the proposed three frameworks and the DBN framework [34] using the complete dataset. This experiment involved three audio descriptors, which were MFCC13, MFCC13 + S, and MFCC13 + S + R. The table identifies the audio descriptor that generated the best average F-score. The experimental results reveal that all three proposed frameworks yielded an F-score that was at least 7% better than that of the DBN. The frameworks of the DBN, the proposed ADBN, the proposed HADBN, and the proposed HCDBN yield average F-scores of 85.71%, 93.26%, 93.38%, and 92.83%, respectively. The best result was obtained when using the HCDBN to classify the musical data into 19 subtechniques, which yielded an average F-score of 96.82%.

F. Applied Proposed System on Real-World Applications

In this section, an experiment is performed to test the performance of the proposed framework in the real-world environment. The test data are real-world audio clips of guitar music signals.

The main differences between these test data and the test data in the GPT dataset are that the playing techniques were not applied continuously, the existence of noise, and that the audio clips have unequal durations. The real-world guitar playing audio clips have various lengths, from 0.1 to 5 s. Therefore, it is difficult to determine the starting and ending points of each guitar audio clip.

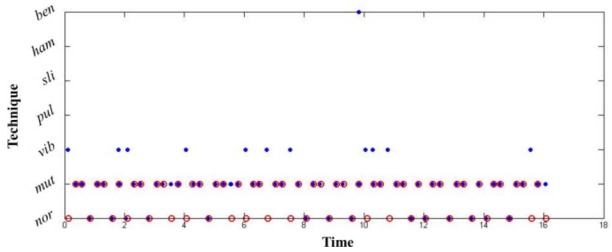


Fig. 9. GPT classification results for riff.wav.

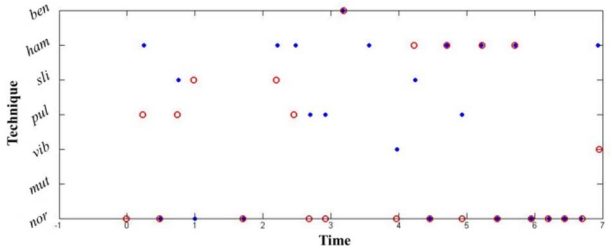


Fig. 10. GPT classification results for solo1.wav.

In this experiment, first, the onset time of the audio clip is detected using the method proposed by Kehling *et al.* [25]. Then, the audio descriptors, such as the MFCC and the STRF-based audio descriptors, are acquired from the audio data between consecutive onsets. Finally, the descriptors are fed into the trained DNN to identify the playing techniques in the audio clip.

Fig. 9 presents the results of onset detection and the recognition of the GPT in a real-world audio clip (riff.wav). The x-axis represents the time and the y-axis represents the class of the playing technique (1: normal, 2: muting, 3: vibrato, 4: pull off, 5: hammer-on, 6: sliding, and 7: bending). Red circles represent the ground truth, and blue dots represent the GPT classification results. The GPT recognition rate is $53/65\times100\%=81.54\%$.

Another experiment is conducted using another audio clip file, solo1.wav. Fig. 10 presents the results of onset detection and the recognition of the GPT in a real-world audio clip (solo1.wav). The x-axis represents the time and the y-axis represents the class of the playing technique (1: normal, 2: muting, 3: vibrato, 4: pull off, 5: hammer on, 6: sliding, and 7: bending). Red circles represent the ground truth, and blue dots represent the GPT classification results. The GPT recognition rate is $12/22\times100\%=54.55\%$.

The performance of the proposed system in the real-world demonstrates that the system can perform GPT classification accurately even in a real-world environment.

VII. CONCLUSION

This work proposed a system for identifying GPTs using an STRF-based scale descriptor, an STRF-based rate descriptor, and a new DNN, called an HCDBN. Simulations show that the proposed framework has yielded very high recognition rates. With the split signal of the guitar music data, the proposed framework yielded 11.47% and 13.32% higher

average F-scores than the baseline system in [12] and the DBN baseline [34], respectively. The proposed framework also yielded a much higher F-score than using the DBN when a complete signal of guitar music data was used. The proposed GPT classification system yielded an average F-score of 80.23% with the split signal and of 96.82% with the complete signal. Moreover, audio clips of guitar music signals in a real-word environment were used to evaluate the performance of the proposed system. The experimental results demonstrate that the proposed framework can work well even in a real-world environment. Finally, the proposed STRF-based descriptors are able to capture audio characteristics, such as formants and harmonics. Therefore, it is especially suitable for analyzing the techniques of various string instruments. Apart from plucked string instruments, such as the guitar, the proposed technique classification can also be applied to bowing string instruments, such as the violin.

REFERENCES

[1] K. Takano and S. Sasaki, "An Interactive music learning system in ensemble performance class," in *Proc. IEEE Int. Conf. Broadband Wireless Comput. Commun. Appl.*, Barcelona, Spain, 2011, pp. 65–74.

[2] G. Burlet and A. Hindle, "Isolated guitar transcription using a deep belief network," *PeerJ Comput. Sci.*, vol. 3, p. e109, Mar. 2017.

[3] J. K. Dhiman, N. Adiga, and C. S. Seelamantula, "A spectro-temporal demodulation technique for pitch estimation," in *Proc. Interspeech*, 2017, pp. 2306–2310.

[4] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech, Language Process. (TASLP)*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[5] Y.-S. Lee, C.-Y. Wang, S.-F. Wang, J.-C. Wang, and C.-H. Wu, "Fully complex deep neural network for phase-incorporating monaural source separation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, 2017, pp. 281–285.

[6] Y. S. Lee, K. Yu, S. H. Chen, and J. C. Wang, "Discriminative training of complex-valued deep recurrent neural network for singing voice separation," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1327–1335.

[7] S. C. Lim, J. S. Lee, S. J. Jang, S. P. Lee, and M. Y. Kim, "Music-genre classification system based on spectro-temporal features and feature selection," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1262–1268, Nov. 2012.

[8] Y. Han, J. Kim, K. Lee, Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Trans. Audio, Speech, Language Process. (TASLP)*, vol. 25, no. 1, pp. 208–221, Jan. 2017.

[9] Y. E. Kim *et al.*, "Music emotion recognition: A state of the art review," in *Proc. Int. Conf. Music Inf. Retrieval*, 2010, pp. 255–266.

[10] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music into emotions," in *Proc. Int. Conf. Music Inf. Retrieval*, 2008, pp. 325–330.

[11] L. Reboursiere, O. Lahdeoja, T. Drugman, S. Dupont, C. Picard-Limpens, and N. Riche, "Left and right-hand guitar playing techniques detection," in *Proc. Int. Conf. New Interfaces Music. Exp.*, 2012, pp. 7–10.

[12] L. Su, L. F. Yu, and Y. H. Yang, "Sparse cepstral and phase codes for guitar playing technique classification," in *Proc. 15th Int. Conf. Music Inf. Retrieval*, 2014, pp. 9–14.

[13] Y. P. Chen, L. Su, and Y. H. Yang, "Electric guitar playing technique detection in real-world recording based on F0 sequence pattern recognition," in *Proc. Int. Conf. Music Inf. Retrieval*, 2015, pp. 708–714.

[14] L. Su, H.-M. Lin, and Y.-H. Yang, "Sparse modeling of magnitude and phase-derived spectra for playing technique classification," *IEEE/ACM Trans. Audio, Speech, Language Process. (TASLP)*, vol. 22, no. 12, pp. 2122–2132, Dec. 2014.

[15] W. Jeon and B. H. Juang, "Speech analysis in a model of the central auditory system," *IEEE/ACM Trans. Audio, Speech, Language Process. (TASLP)*, vol. 15, no. 6, pp. 1802–1817, Aug. 2007.

[16] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *J. Acoust. Soc. Amer.*, vol. 118, no. 2, pp. 887–906, Aug. 2005.

[17] J.-C. Wang, C.-H. Lin, E.-T. Chen, and P.-C. Chang, "Spectral-temporal receptive fields and MFCC balanced feature extraction for noisy speech recognition," in *Proc. Asia–Pac. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Siem Reap, Cambodia, 2014, pp. 1–4.

[18] J. C. Wang, C. Y. Wang, Y. H. Chin, Y.-T. Liu, E. T. Chen, and P. C. Chang, "Spectral-temporal receptive fields and MFCC balanced feature extraction for robust speaker recognition," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 4055–4068, Feb. 2017.

[19] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, Apr. 2011.

[20] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proc. Int. Conf. Music Inf. Retrieval*, 2012, pp. 565–570.

[21] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: New directions for music informatics," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 461–481, Dec. 2013.

[22] M. Grachten and F. Krebs, "An assessment of learned score features for modeling expressive dynamics in music," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1211–1218, Aug. 2014.

[23] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. Int. Conf. Music Inf. Retrieval*, 2010, pp. 339–344.

[24] G. Gwardys and D. Grzywczak, "Deep image features in music information retrieval," *Int. J. Electron. Telecommun.*, vol. 60, no. 4, pp. 321–326, Dec. 2014.

[25] C. Kehling, J. Abe, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters," in *Proc. 17th Int. Conf. Digit. Audio Effects*, 2014, pp. 1–8.

[26] B. Juang and T. Chen, "The past, present, and future of speech processing," *IEEE Signal Process. Mag.*, vol. 15, no. 3, pp. 24–48, May 1998.

[27] T. Petersen and S. Boll, "Critical band analysis-synthesis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 3, pp. 656–663, Jun. 1983.

[28] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[29] A. Mnih and G. Hinton, "Learning nonlinear constraints with contrastive backpropagation," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2. Montreal, QC, Canada, 2005, pp. 1302–1307.

[30] A. Krizhevsky, S. Ilya, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25. Red Hook, NY, USA: Curran, 2012, pp. 1097–1105.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 1026–1034.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Sep. 2014, pp. 1–14. [Online]. Available: arXiv:1409.1556

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 9, 2010, pp. 249–256.

[34] M. A. Keyvanrad and M. M. Homayounpour, "A brief survey on deep belief networks and introducing a new object oriented MATLAB toolbox (DeeBNet V2.2)," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Aug. 2014, pp. 1–27. [Online]. Available: arXiv:1408.3264

Chien-Yao Wang received the B.S. degree in computer science and information engineering and the Ph.D. degree from National Central University, Taoyuan, Taiwan, in 2013 and 2017, respectively.

He is currently a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests include signal processing, deep learning, and machine learning.

Dr. Wang is an Honorary Member of the Phi Tau Phi Scholastic Honor Society.

Pao-Chi Chang (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1986.

From 1986 to 1993, he was a Research Staff Member of the Department of Communications, IBM T. J. Watson Research Center, Hawthorne, NY, USA. At Watson, his work centered on high-speed switching systems, efficient network design algorithms, and multimedia conferencing. In 1993, he joined the Faculty of National Central University, Taoyuan, Taiwan, where he is currently a Professor with the Department of Communication Engineering. Since 1994, he has been established and has headed the Video–Audio Processing Laboratory, Electrical Engineering Department and the Communication Department, National Central University. He is the Principle Investigator for many joint projects with National Science Council, Institute of Information Industry, Chung Hwa Telecommunication Laboratories, and many other companies. His research interests include speech/audio coding, video/image compression, scalable coding, error resilient coding, digital watermarking and data hiding, and multimedia delivery over packet and wireless networks.

Jian-Jiun Ding (Senior Member, IEEE) was born in Taiwan in 1973. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, in 1995, 1997, and 2001, respectively.

From 2001 to 2006, he was a Postdoctoral Researcher with the Department of Electrical Engineering, NTU, where he is currently a Professor with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering. His current research interests include time-frequency analysis, fractional Fourier transforms, linear canonical transforms, wavelet transforms, image processing, image compression, orthogonal polynomials, fast algorithms, integer transforms, quaternion algebra, pattern recognition, and filter design.

Tzu-Chiang Tai (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2010.

He is an Associate Professor with the Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan. His current research interests include reconfigurable computing, VLSI design automation, and algorithm design and analysis.

Andri Santoso received the B.S. degree in computer science from Brawijaya University, Malang, Indonesia, in 2012, and the M.S. degree in computer science and information engineering from National Central University, Taoyuan, Taiwan, in 2016.

His main areas of research interest are image processing, signal processing, and deep learning.

Yu-Ting Liu received the M.S. degree in communication engineering from the Video–Audio Processing Laboratory, NCU, Taoyuan, Taiwan.

Her research interests include speech recognition, audio signal processing, and deep learning.

Jia-Ching Wang (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2002.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Central University, Jhongli, Taiwan. He was an Honorary Fellow with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI, USA, from 2008 to 2009. His research interests include signal processing, machine learning, deep learning, and VLSI architecture design.

Prof. Wang is an Honorary Member of the Phi Tau Phi Scholastic Honor Society and a member of ACM and IEICE.