Han and Lee *EURASIP Journal on Audio, Speech, and Music Processing* (2016) 2016:2
DOI 10.1186/s13636-015-0079-0

EURASIP Journal on
Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

Detecting fingering of overblown flute sound using sparse feature learning

Yoonchang Han[1] and Kyogu Lee[1,2*]

Abstract

In woodwind instruments such as a flute, producing a higher-pitched tone than a standard tone by increasing the blowing pressure is called overblowing, and this allows several distinct fingerings for the same notes. This article presents a method that attempts to learn acoustic features that are more appropriate than conventional features such as mel-frequency cepstral coefficients (MFCCs) in detecting the fingering from a flute sound using unsupervised feature learning. To do so, we first extract a spectrogram from the audio and convert it to a mel scale. Then, we concatenate four consecutive mel-spectrogram frames to include short temporal information and use it as a front end for the sparse filtering algorithm. The learned feature is then max-pooled, resulting in a final feature vector for the classifier that has extra robustness. We demonstrate the advantages of the proposed method in a twofold manner: we first visualize and analyze the differences in the learned features between the tones generated by standard and overblown fingerings. We then perform a quantitative evaluation through classification tasks on six selected pitches with up to five different fingerings that include a variety of octave-related and non-octave-related fingerings. The results confirm that the learned features using the proposed method significantly outperform the conventional MFCCs and the residual noise spectrum in every experimental condition for the classification tasks.

Keywords: Musical instrument, Flute overblowing detection, Unsupervised feature learning, Feed-forward neural networks, Sparse filtering, Timbre analysis, Music information retrieval

1 Introduction

Tone production on a flute involves the speed of an air jet across the embouchure and the fingering used [1]. Every pitch on the flute has its own standard fingering with an appropriate blowing pressure. However, it is possible to generate a tone that is higher in frequency than the usual tone by increasing the blowing pressure [2]; this is referred to as “overblowing” or “harmonic” by flutists. This implies that it is also possible to generate the same-pitched sound with different fingerings. For instance, a C6 tone can be produced using a C4 or C5 fingering as well as a C6 fingering. However, each fingering produces a sound with a slightly different timbre.

Although standard fingering generates a brighter and more stable tone than an overblown sound, alternative fingerings are frequently used to minimize mistakes in note transition or to add a special color to the tone, especially in contemporary repertoire [3]. On the other hand, novice players unintentionally produce overblown sounds because the fingering rules of a flute are not always consistent. In particular, most of the octave-4 and octave-5 fingerings are identical except for C, C#, D, and D#, and thus, many beginner flute players use octave-4 fingerings for octave-5 notes.

Since the timbral differences among the same-pitched notes with distinct fingerings are not obvious, music transcription systems typically focus only on detecting the onset and pitch of a note, discarding the fingering information. However, it is valuable to figure out which fingering is used to generate a specific note, particularly for musical training purposes. Handcrafted audio features such as mel-frequency cepstral coefficients (MFCCs) are commonly used for musical timbre analysis, but there is an increasing interest in learning features from data in an unsupervised manner.

This paper proposes a method that attempts to estimate the fingering of a flute player from an audio signal

* Correspondence: kglee@snu.ac.kr
[1]Music and Audio Research Group (MARG), Graduate School of Convergence Science and Technology, Seoul National University, 599 Gwanak-ro, Seoul, Republic of Korea
[2]Advanced Institutes of Convergence Technology (AICT), Suwon, Republic of Korea

based on sparse filtering (SF), which is an unsupervised feature-learning algorithm [4]. In our previous work, we demonstrated that the learned features could be used to detect the mistakes of beginner flute players [5]. However, the performance of the learned features was nearly identical to that of the MFCCs, which are among the most widely used handmade features for timbre analysis. Furthermore, the evaluation was limited in that only octave-related fingerings were considered for binary classification, and the same flute was used to create all of the sound samples, thus failing to generalize for all types of overblowing.

In this paper, we extend our previous work with a more complete evaluation by developing a fingering dataset that contains flute sound samples from various materials and makers. A major motivation for this work is to determine whether the performance of learned features shows a consistent increase in performance over conventional features under various types of real-world situations, because learned sparse features have been shown to outperform handmade features such as MFCCs for various music information tasks such as genre classification and automatic annotation [6, 7].

To avoid classification errors caused by the blowing skill of the performer, we recorded sound samples from players with a minimum of 3 years' experience. In addition, we expanded the dataset to include more target pitches up to octave 6 and experimented with up to five different fingerings for each selected note. In terms of our algorithm, we changed the input for feature learning from linear-scale spectrogram and MFCCs to a mel-scale spectrogram. We also added max-pooling to make the system more robust.

The remainder of the paper is organized as follows. We present related works in Section 2 and describe the spectral characteristics of an overblown tone in Section 3. In Section 4, we present the overall structure of the proposed system, including the preprocessing, feature-learning, max-pooling, and classification steps. In the following section, we explain the evaluation procedure, including the process of dataset construction, in detail. In Section 6, we present the results of the experiments and discussions, followed by the conclusions and directions for future work in Section 7.

The term "flute" is widely used across cultures and can refer to one of the various types of woodwind instruments. In this paper, "flute" means a modern open-hole Boehm flute, which is the most common member of the flute family and a regular member of symphony orchestras in the West [8].

2 Related work

A study by Kereliuk et al. [9] and Verfaille et al. [10] attempted to detect overblown flute fingering using the residual noise spectrum with principal component analysis (PCA) and linear discriminant analysis (LDA). This approach uses energy measurements of multiples of the fundamental frequency submultiples F0/l where ($l = \{2,3,4,5,6\}$) in the first octave and a half (i.e., between F0 and 1.5F0). This is where, in their observation, the most noticeable differences appear [10]. The spectrum energy is measured using a Hann window centered on the region of interest. In addition, the researchers use a comb-summed energy measure, which simply sums the energies of the same harmonic comb, to reduce dimensionality in the experiment. In addition, the researchers recorded a flute sound from a microphone attached to the flute head joint, and only the attack segments of the notes were used in the experiment. Their proposed method allows for a detection error below 1.3 % for notes with two and three possible fingerings. However, this error dramatically increased up to 14 % for four and five possible fingerings. Although this system is specifically designed for detecting overblown flute sounds by measuring energy in the region of interest, from the experimental results, it is possible to say that the system does not capture all existing spectral differences between the sounds from each fingering. We plan to solve this limitation by replacing hand-designed features with sparse feature learning.

It is reported that using learned features has matched or outperformed handmade features in a range of different modalities [4]. Recently, there have been an increasing number of attempts to apply various feature-learning methods to the music information retrieval field. Henaff et al. applied a sparse coding algorithm to a single frame of a constant-Q transform spectrogram for musical genre classification [11], and Schülter et al. applied restricted Boltzmann machines (RBMs) to similarity-based music classification [12]. In addition, Nam et al. applied sparse RBMs to music annotation and piano transcription [13, 14], and Lee et al. used convolutional deep-belief networks for music genre and artist classification [7].

As shown in the works listed above, feature learning shows promising results in many audio and music information retrieval fields when appropriate input data, learning algorithms, and pre/postprocessing methods are used.

3 Spectral characteristics of overblown tone

Timbre is often described by musicians with adjectives such as full, rich, dull, mellow, and round [15] because it is associated with the perceived feeling of the sound derived from various spectral characteristics. A timbral difference between the sound generated by standard fingering and overblown fingering is actually very small; thus, it is difficult for non-musicians to spot the difference. As shown in Fig. 1,

spectrograms of a D5 tone with proper D5 fingering and D4 fingering look quite alike. However, flute experts can easily distinguish the difference. Usually, a standard tone is described as "clearer," and an overblown flute sound is slightly more "airy" than the original tone. The sound of a regular flute tone is highly sine-wave-like, and only multiples of F0 are visibly strong, while the other part of the spectrum has very low energy. The "airy" timbre of the overblown tone is mainly caused by the spectral energy existing at multiples of other than F0. It is difficult, but still possible, to observe from Fig. 1 that the spectrum of the overblown sound has strong peaks at multiples of F0 (587 Hz) as well as minor energy around multiples of F0 of the original fingering (293 Hz) between F0 and 2F0, and 2F0 and 3F0. Interestingly, this tendency is not clear between 3F0 and 4F0 but appears again between 4F0 and 5F0 for a D5 tone. As shown above, it is certain that an airy timbre is caused by residual noise at multiples of other than F0, but it is difficult to set a concrete handmade rule for the spectral characteristics to distinguish every fingering.

4 System architecture

We perform flute-fingering detection using the system architecture presented in Fig. 2. The details of each block in the diagram, including the parameter values, are explained in this section.

4.1 Preprocessing

Although sparse feature learning effectively learns the feature from the data, appropriate preprocessing is a prerequisite to obtaining a good feature. In addition, we need to decrease the dimensionality as much as possible without losing important parts of the information. We perform several steps for preprocessing, as described below.

4.1.1 Downsampling

In the first preprocessing step, the input audio is downsampled to 22,050 Hz from the original 44,100-Hz sampling rate. Since now the Nyquist frequency of the input signal is 11,025 Hz, we can get rid of unnecessary noisy information above this frequency while retaining enough numbers of harmonics for the highest note of the flute. This experiment covers frequencies up to the seventh harmonic of G6.

4.1.2 Time-frequency representation

We compute a discrete Fourier transform (DFT) for the spectrogram of the input audio with a 25-ms window and a 10-ms hop size. Then, its linear frequency scale is converted to a mel scale, and the spectral magnitude is compressed by a natural logarithm. We chose 128 for the number of mel-frequency bins, following Hamel's [16] and Nam's work [14]. By using a moderate number of mel-frequency bins instead of a linear-scale DFT result, it is possible to reduce the input dimension

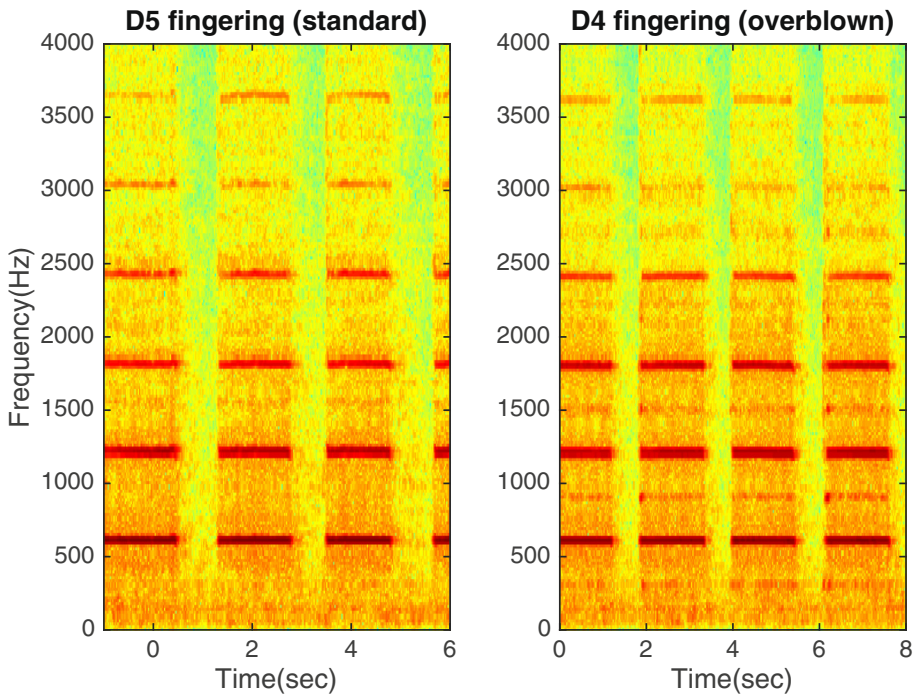

Fig. 1 Log spectrogram example of standard and overblown tone. Standard tone is D5 tone played with regular D5 fingering, and overblown tone is D5 tone played with D4 fingering with a sharper and stronger air jet

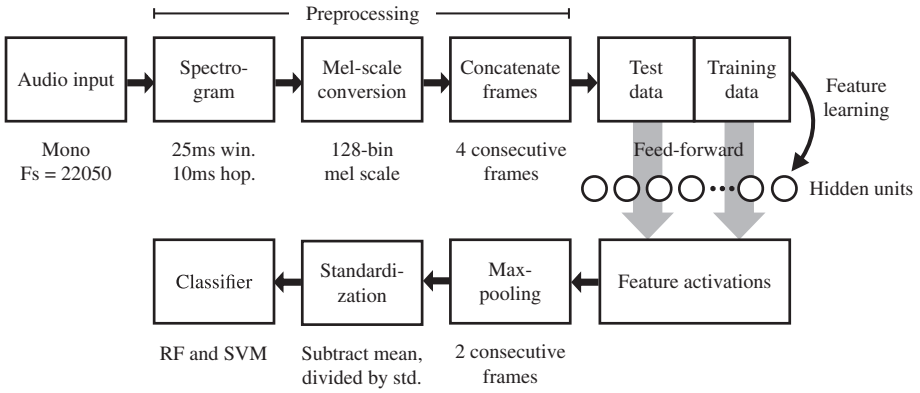

Fig. 2 Overall schematic of proposed flute fingering detection system. The system takes an audio waveform as an input, and four consecutive frames of a 128-bin mel-scale spectrogram are concatenated to learn timbral features. Note that features are learned only from training data. Obtained feature activations are max-pooled and standardized prior to training the classifier

significantly while preserving the audio content effectively enough. This is very helpful for decreasing the overall computational complexity of the system because the DFT of an audio signal generates a high-dimensional vector, and the next frame-concatenation step multiplies the number of feature dimensions.

4.1.3 Frame concatenation

As a final stage of preprocessing, we concatenate several consecutive frames as a single input for feature learning. This process can be seen as learning temporal information within the size of a concatenation for feature learning and to make the system more robust than when we use a single frame. We concatenate four frames as a single-input example. Thus, the time resolution of the data in this step is now 70 ms according to the window and hop size mentioned above.

4.2 Feature learning and representation

We learn sparse features from the preprocessed data described above. This section explains the sparse filtering algorithm, activation function, and details about how we generated the sparse feature representation.

4.2.1 Unsupervised feature learning

Recent efforts in machine learning have sought to define a method to automatically learn higher-level representations of input data [17]. Such an approach is extremely valuable, particularly when designing features by hand is challenging, and the timbre is an example where it is difficult to determine a distinctive difference in a sound spectrum by observation. Many choices are available for unsupervised feature learning, such as restricted Boltzmann machines [18], autoencoder [19], and sparse coding [20]. These approaches have been successfully applied to a variety of modalities, but they require extensive tuning of the hyperparameters [4].

From these approaches, we chose to employ sparse filtering as a feature-learning algorithm in the proposed system because it has only one hyperparameter (the number of features to learn) and converges more quickly than other algorithms, especially when the number of input data and the feature dimension are large. This characteristic is suitable for our task because it can be easily implemented in a real-time score-following system for mobile applications without fine-parameter tuning for each device and environment. Furthermore, a short-time Fourier analysis of audio wave inherently generates a significant number of input data.

4.2.2 Sparse filtering algorithm

Sparse filtering first normalizes each feature to be uniformly active in total by dividing each feature by its ℓ_2-norm throughout all examples as follows:

$$f_j' = f_j / \left\| f_j \right\|_2 \tag{1}$$

where f_j represents the jth feature value. In a similar manner, it then normalizes each example by dividing each example by its ℓ_2-norm across all features as follows:

$$\tilde{f}^{(i)} = f'(i) / \left\| f'(i) \right\|_2 \tag{2}$$

where $f^{(i)}$ represents the ith example. By computing Eq. 1 and Eq. 2, now all values lie in the unit-ℓ_2 hypersphere. This feature normalization introduces competition between features. For example, if one component of $f^{(i)}$ is increased, other components will decrease because of the normalization. Finally, the normalized features are

optimized for sparseness using an ℓ_1 penalty, and it can be written as

$$\text{minimize} \sum_{i=1}^{N} \left\| \tilde{f}^{(i)} \right\|_1 = \sum_{i=1}^{N} \left\| \frac{f'(i)}{\left\| f'(i) \right\|_2} \right\|_1 \quad (3)$$

for a dataset of N examples. Consequently, each example is represented by a small number of active sparse units, and each feature is active only for a small number of examples at the same time. A more detailed description of sparse filtering can be found in [4].

As an activation function, we used t soft-absolute function shown below:

$$f_j^{(i)} = \sqrt{\epsilon + \left(w_j^T x^{(i)} \right)^2} \approx \left| w_j^T x^{(i)} \right| \quad (4)$$

where w is the weight matrix, x is the visible node (i.e., input data), and we set $= 10^{-8}$. An off-the-shelf L-BFGS [21] package is used to optimize the sparse filtering objective until convergence.

4.2.3 Learning strategy and feature representation

We obtained a weight matrix from training data such that the feature activation of the test data can be simply obtained by a feed-forward process. As mentioned above, sparse filtering only has one parameter to tune: the number of features to learn. To determine the effect of hidden unit size on the overall detection accuracy, we used 39, 512, and 1024 for the number of hidden units. The top 10 most active feature bases for six different notes (C5, D5, C6, C#6, E6, and G6) are shown in Fig. 3. It is possible to observe that harmonic partials of the mel spectrum are distributed into bases.

4.3 Max-pooling

Using short-term features provides high time resolution but is prone to local short-term errors. Max-pooling is the process that takes the largest value in the region of interest such that only the most responsive units are retained. We apply max-pooling over the time domain to every two non-overlapping consecutive frames. This temporal maximum value-selection process is included to add an extra robustness to the system and to reduce the computational complexity of the next step, a classifier.

It is important to note that max-pooling is a nonlinear downsampling; hence, the time resolution becomes half of the original resolution when it is pooled every two frames. In the proposed system, the original time resolution of the spectrogram is 25 ms, and it becomes 70 ms after we concatenate four frames with a hop size of 10 ms. Since we are pooling over two frames, the time resolution after max-pooling is 140 ms. Increasing the size of the max-pooling region might increase the detection accuracy further, but this is an appropriate resolution for flute transcription because a typical sixteenth note with 100 beats per minute (BPM) is 150 ms, which is still greater than our frame size.

4.4 Classification

There are a variety of choices available for classifier. The support vector machine (SVM) and random forest (RF) are both highly popular classifiers across various applications. The underlying idea behind the SVM is to calculate a maximal margin hyperplane that performs a binary classification of the data [22]. By contrast, RF is an algorithm that uses a combination of decision trees that have a randomly selected subset of variables [23].

The performances of the SVM and RF are a highly arguable topic, and there is significant variability in their problems and metrics [24]. We use both SVM and RF as classifiers to compare the performance. We first standardize the values by subtracting the mean and divide them by the standard deviation prior to feeding the data into the classifier. Thus, the data have their mean at zero and standard deviation at one. We use a radial basis function (RBF) kernel for SVM, and the number of trees for RF is set as 500.

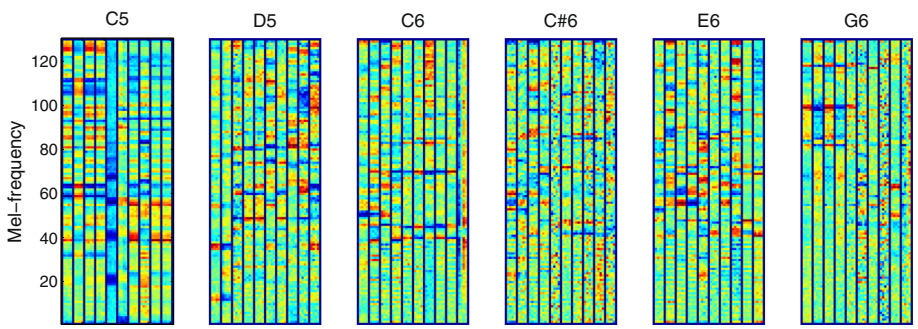

Fig. 3 Top 10 most active feature bases of each note. It is possible to observe that harmonic partials are distributed into bases. The bases are learned from sparse filtering with 39 hidden units

Table 1 Selected fingering set

Played note	Fingerings
C5	C4, C5
D5	D4, D5
C6	C4, C6, F4
C#6	C#4, C#6, F#4
E6	A4, C4, E4, E6
G6	C4, C5, E♭4, G4, G6

Played note is a pitch of the note, and each pitch is played with different fingerings by altering only the blowing pressure. The evaluation is composed of six independent classification problems, and the number of classes is the number of fingerings

5 Evaluation

5.1 Data specification

We created a dataset that resembles the one in a previous study by Verfaille et al. [10]. This includes diverse overblown fingering cases such as octave-related to non-octave-related fingerings and "few keys changing" to "many keys changing" positions, as illustrated in Table 1. We recorded flute sounds from two expert performers with more than 20 years of experience and two intermediate performers with 3 years of experience. Each performer played six pitches with all possible fingerings as they appeared in Table 1. The average length of each tone was 16 s, excluding noises and silences. The length of the audio recorded from each performer was 305 s on average, and the total audio length was 1218 s. For flute tones, performers sustained the target tone for each fingering without vibrato, special articulations, or melodic phrases. We extracted 122 K frames of spectrogram in total using a 25-ms window and 10-ms hop size before frame concatenation and the max-pooling step.

The flutes used for the recording were modern Boehm flutes with B foot joints made by multiple flute makers and of different materials. Two intermediate players used silver heads with nickel body joints (Yamaha). One expert used a sterling silver flute (Powell), and another expert used a rose gold flute (Brannen-Cooper). Note that only the first nickel body flute had a "split E" mechanism, which facilitates the production of E6. Every flute used in the experiment was an open-hole flute without any cylindrical acrylic plugs.

Flutes made of several different materials are used for the experiment because not every flutist uses a golden flute, and the general timbre of the flute changes according to the material used in its construction. For instance, the flute made of gold is often described as having a fuller, richer, and more liquid timbre, while the silver flute is more delicate and shrill in the loud and high tones [25].

Although we used the same fingering set as [10], our data is recorded without attaching a microphone to the flute head joint. To simulate a real-world situation, audio samples were recorded using the built-in microphone of a Samsung NT900 laptop rather than by using high-end studio microphones. Flute sounds from four different performers were all recorded in different locations. Three performers played their flutes in their apartment living rooms, and one performer played the flute in a small office. The dataset is available to download from (http://marg.snu.ac.kr/DFFD), which includes the audio files with annotations.

5.2 Experiments

A stratified eightfold cross validation was used to evaluate the performance of the proposed method. As shown in Table 1, our experiment is composed of six independent classification problems, and the number of classes for each note is the number of fingerings. First, we collected flute tones from every performer and partitioned the data into eightfold that were proportionally representative of each class. Then, onefold was used for a test; the remaining folds were used for training. Note that this procedure was performed frame-wise, and the order of the frames was random. In addition, each fold includes the audio recorded from flutes of various materials and in various recording environments. The mean and standard deviation of the detection error probability P_f (i.e., inverse accuracy) was calculated eight times via repeated cross validation for each fingering of each selected pitch.

We also evaluated MFCCs to compare them with our proposed feature representation. To make a fair comparison, we used the same experiment settings, such as identical frame and hop sizes, concatenating four frames as a single example and standardizing to have a zero mean and unit variance. Thirteen-dimensional MFCCs were used with delta and double delta for a total of 39 dimensions. As mentioned previously, we evaluated the performance of sparse features with 39 hidden units for a fair comparison with MFCCs, and 512 and 1024 units are also evaluated in order to determine the effects of increased hidden unit size on the overall classification accuracy.

6 Results and discussion

The detection accuracy of the proposed flute-fingering detection method and the effect of parameters, classifiers, and max-pooling are presented in this section. We compare the performance of sparse feature representation with existing approaches such as MFCCs and the PCA/LDA method.

6.1 Comparison with MFCCs

In general, the proposed algorithm outperformed MFCCs for every fingering set in the same 39-dimensional setting as indicated in Table 2. The worst detection error rate of

Table 2 Eightfold cross-validation result of the fingering detection system

		SVM		Random forest	
Note	Feature (dim.)	$P_f \pm \sigma^2_{Pf}$	$P_f \pm \sigma^2_{Pf}$ (Mp)	$P_f \pm \sigma^2_{Pf}$	$P_f \pm \sigma^2_{Pf}$ (Mp)
C5	MFCCs (39)	9.87 ± 1.57	8.07 ± 1.00	14.91 ± 1.91	12.20 ± 1.05
	SF (39)	0.55 ± 0.47	0.20 ± 0.27	0.68 ± 0.52	0.13 ± 0.24
	SF (1024)	0.26 ± 0.24	0.00 ± 0.00	0.36 ± 0.24	0.07 ± 0.19
D5	MFCCs (39)	7.43 ± 1.30	8.21 ± 0.82	10.16 ± 1.59	10.20 ± 2.42
	SF (39)	2.20 ± 0.78	1.41 ± 1.12	1.48 ± 0.54	0.40 ± 0.51
	SF (1024)	1.67 ± 0.72	1.18 ± 0.68	1.00 ± 0.36	0.39 ± 0.38
C6	MFCCs (39)	8.47 ± 0.86	6.91 ± 1.48	11.78 ± 1.59	8.91 ± 0.82
	SF (39)	2.08 ± 0.88	1.50 ± 0.80	1.30 ± 0.50	0.67 ± 0.40
	SF (1024)	0.76 ± 0.42	0.13 ± 0.17	0.56 ± 0.34	0.04 ± 0.12
C#6	MFCCs (39)	7.01 ± 0.62	5.47 ± 0.90	9.31 ± 1.01	6.02 ± 0.98
	SF (39)	1.39 ± 0.72	1.07 ± 0.82	1.26 ± 0.52	0.56 ± 0.33
	SF (1024)	0.31 ± 0.29	0.08 ± 0.15	0.29 ± 0.27	0.04 ± 0.11
E6	MFCCs (39)	6.36 ± 0.78	4.56 ± 0.90	7.43 ± 0.96	4.80 ± 1.27
	SF (39)	2.35 ± 0.79	1.74 ± 0.76	2.03 ± 0.46	1.31 ± 0.51
	SF (1024)	1.06 ± 0.48	1.11 ± 0.30	1.02 ± 0.37	0.34 ± 0.24
G6	MFCCs (39)	13.80 ± 1.14	12.77 ± 1.28	13.76 ± 1.12	10.63 ± 0.96
	SF (39)	5.71 ± 1.30	4.49 ± 0.76	4.62 ± 1.26	3.22 ± 0.58
	SF (1024)	1.53 ± 0.53	1.06 ± 0.54	1.55 ± 0.36	1.06 ± 0.44

Mean and standard deviations of detection error ($P_f \pm \sigma^2_{Pf}$) probabilities (%) using MFCCs and sparse filtering (SF) for 39 and 1024 units. SVM and random forest were used for the classifications, and probabilities are given for features without and with max-pooling (Mp)

the MFCCs occurred when $P_f = 14.91 \pm 1.91$ % for C5 with RF; the highest error rate of the proposed method occurred when $P_f = 5.71 \pm 1.30$ % for G6 with SVM.

The proposed system does not use PCA for orthogonalization or dimension reduction. However, the first and second highest eigenvalues of PCA are computed from MFCCs and sparse features in order to visualize two-dimensional feature representation, as shown in Fig. 4. In addition, we also present a visualization using the t-SNE algorithm described in [26] in Fig. 4. This algorithm is capable of learning high-order dependencies. From this figure, it is possible to observe that MFCCs overlap significantly for all fingerings, whereas the generated sparse feature, by comparison, overlaps much less.

6.2 Effect of max-pooling

Performing max-pooling on the obtained feature activation visibly improved the performance for both MFCCs and sparse features, as shown in Table 2. It can also be observed from Fig. 4 that adding max-pooling effectively removes a significant amount of feature noise. We perform max-pooling over two consecutive frames. Increasing the pooling size might increase the performance further, but we decided not to increase the pooling size in order to keep the time resolution smaller than the length of a sixteenth note at 100 BPM (150 ms).

6.3 Effect of the number of hidden units

Although we used 39 dimensions for a fair comparison with the MFCCs, we evaluated higher values for the number of hidden units to determine their effects. Using additional hidden units means that more bases are used to describe the input signal. This effectively decreased the detection error rate. The classification performance nearly approached saturation at 256 units with a slight improvement up to 1024 units, as shown in Fig. 5.

6.4 Effect of classifier

It is interesting to note that RF demonstrated generally matched or better performance for sparse features, and SVM exhibited superior performance for MFCCs, as shown in Fig. 6. For example, RF showed a lower error rate for D5, C6, C#6, E6, and G6 for SF, while SVM performed better for C5, D5, C6, C#6, and E6 for MFCCs. However, the performance gap between RF and SVM for MFCCs becomes smaller as the number of possible fingerings increases, while RF constantly returns a better performance for sparse features. We can conclude from this result that RF is more suitable for classifying sparse features, especially for multiclass tasks, and that using SVM is marginally better for MFCCs when there exist only a few possible fingerings.

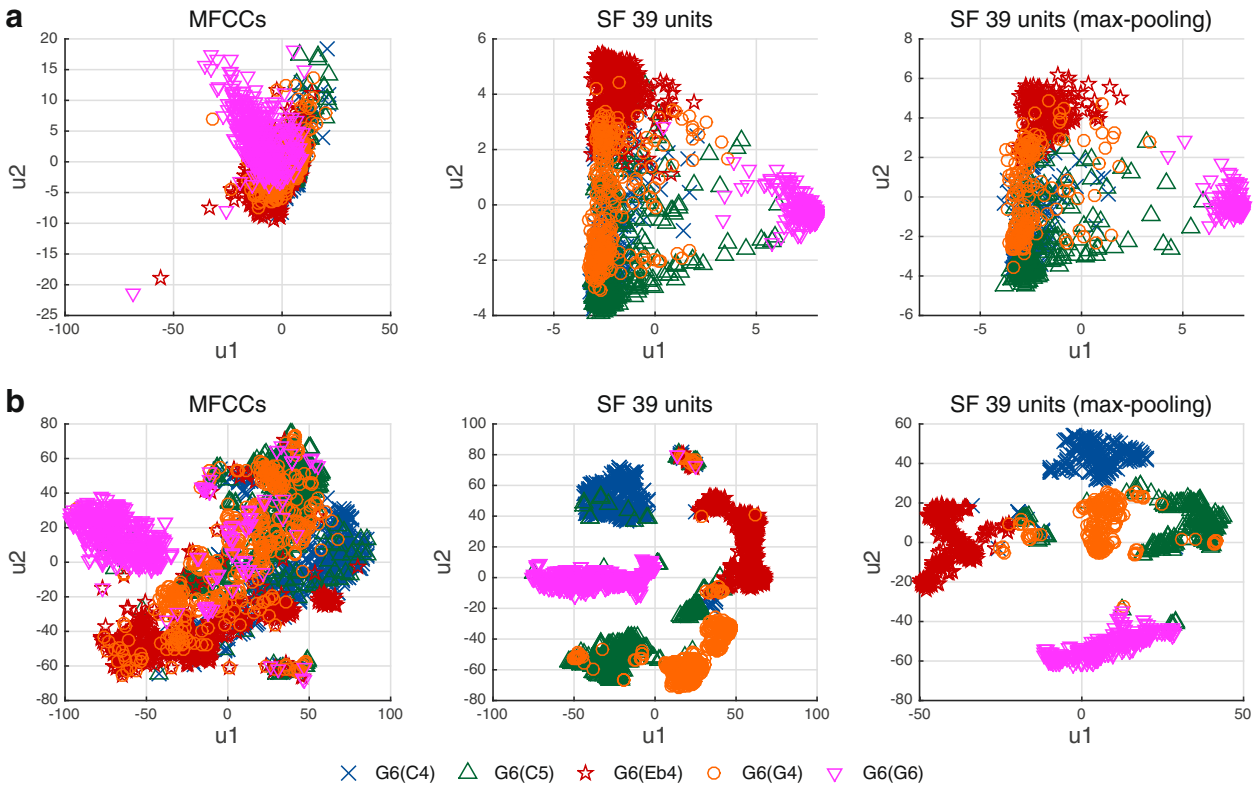

Fig. 4 MFCCs and sparse features with and without max-pooling projected onto a factorial map. The factorial map is built from the first and second highest eigenvalues of PCA and t-SNE (u_1, u_2) for visualization purposes. **a** PCA factorial map of five fingerings (C4, C5, E♭4, G4, G6) producing a G6 tone. **b** t-SNE factorial map of the same tones. It is possible to observe that sparse features are "cleaner" features for flute fingerings than the MFCCs. In addition, max-pooling removes a considerable amount of noise from the feature

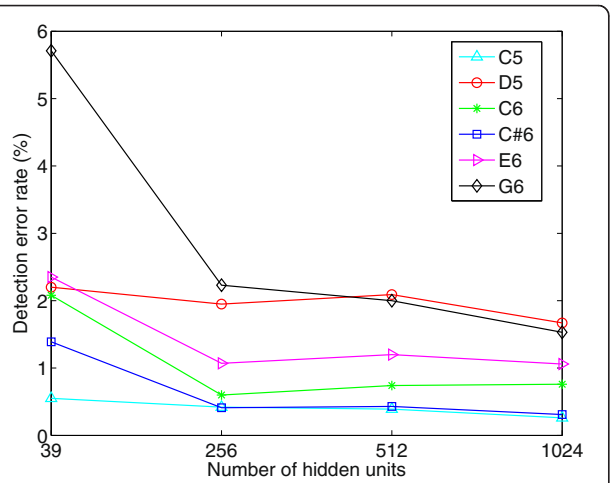

Fig. 5 Effect of number of hidden units. It is possible to observe that the detection error rate decreases as the number of hidden units increases. This nearly approaches saturation at 256 units, with minor improvement up to 1024 units

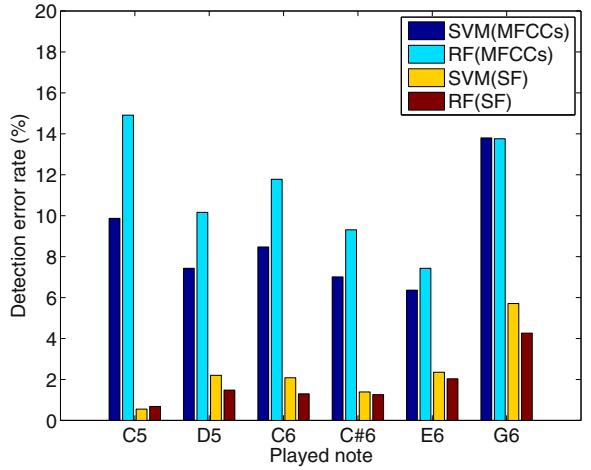

Fig. 6 Effect of classifier. In general, SVM and RF showed better performance for MFCCs and SF, respectively. However, using RF returned an improved result for both features as the number of possible fingerings increase

6.5 Comparison with PCA/LDA method

A direct comparison of the PCA/LDA method from Verfaille et al. [10] with the proposed method would be inappropriate because Verfaille recorded sounds from a microphone attached to the flute head joint and used the mean energy measured from the initial 100 ms of the attack segment only. In our proposed system, we used entire notes rather than attack segments because attack parts do not exist when the notes are played with a slur during the actual flute performance. Further, to simulate a real-world smart-device application, we used a laptop's built-in microphone at a distance to record, rather than attaching an extra microphone to the flute. However, it is meaningful to observe that the error probabilities of the proposed method were less than 5.71 % for 39 sparse feature units and less than 1.11 % for 1024 units with max-pooling applied to four- and five-fingering configurations. The error rates of the PCA/LDA method were below 1.3 % for two and three fingerings; however, these rates dramatically increased to 13.3 % for four and five fingerings in the eightfold cross validation.

This performance improvement was a consequence of the spectral differences between signals successfully captured by sparse feature learning, as visualized in Fig. 7. From the magnitude response plot of C#6 in Fig. 7, it can be observed that the eighth basis is activated for C#4 fingering; however, the eighth basis was not activated for C#6 and F#4. We can see that this eighth basis effectively captures the spectral characteristics of the C#4 fingering between the first and third octave and is particularly clear around 1348 and 1905 Hz, which are annotated with blue circles. Similarly, 1855 Hz of the 32nd basis, annotated with a red circle, is significantly activated for C#4 and F#4 fingering; however, it is not significantly activated for C#6 fingering. This occurs because the sounds from C#4 and F#4 fingerings have strong energy in this region with a peak at 1905 and 1806 Hz, respectively, whereas C#6 has no significant energy in this region.

The handmade feature of [10] uses energy measurements of the multiples of the fundamental frequency submultiples between the first octave and a half (i.e., F0/l where $l = \{2,3,4,5,6\}$ between F0 and 1.5F0) because this is where different acoustical characteristics appear the most significant in their observation, as mentioned above. The proposed system achieved improved performance by learning spectral characteristics with sparse features and describing the sound spectrum with activations of learned bases rather than by restricting a region of interest and measuring energies at certain points.

7 Conclusions

We designed a flute-fingering detection system based on the sparse feature learning method. The results obtained in this study indicate that the learned sparse features delivered improved performance compared with other conventional features for flute-fingering detection, especially as the number of possible fingerings increased. The performance gap between the MFCCs and sparse features for flute-fingering detection was not significant in our previous study [5]; however, the more complicated task with a larger dataset described in this paper confirmed that the learned sparse features were able to capture the

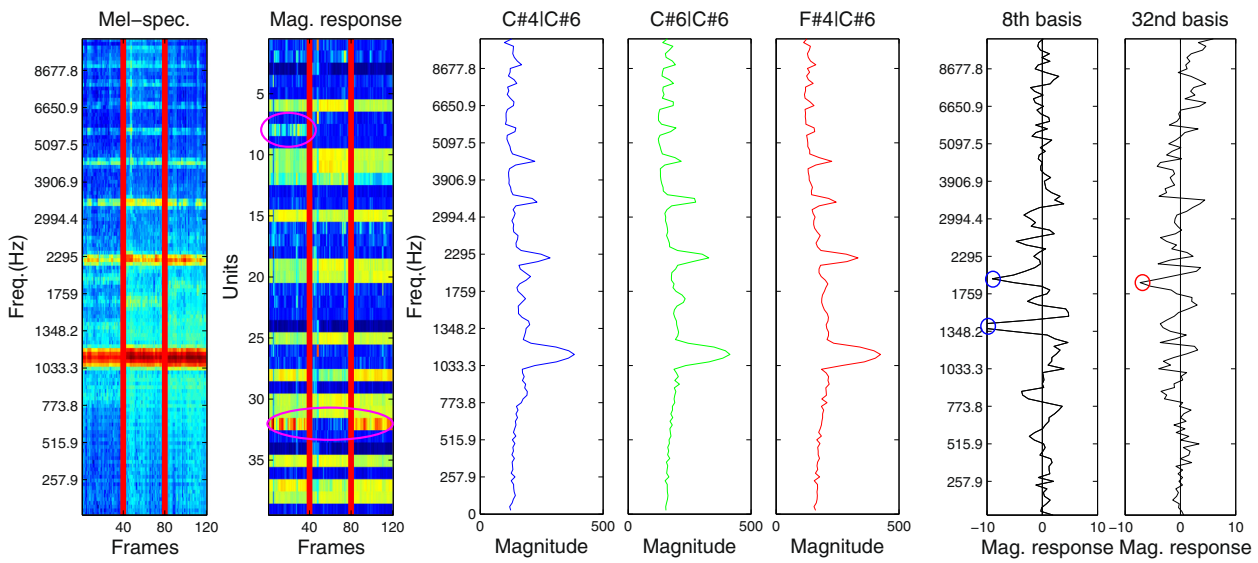

Fig. 7 Mel spectrogram, feature activations, mean mel spectrum, and selected basis of all possible C#6 fingerings. Mel-spectrogram and feature activations of C#4, C#6, and F#4 fingerings are separated by red vertical lines. Unique feature activations for each fingering and important spectral peaks of basis are annotated with circles

spectral differences among the same-pitched notes with distinct fingerings and outperformed the conventional features such as MFCCs that focus on the spectral envelope.

The proposed method achieved a detection error rate of less than 5.71 % for all cases, while the error rate of the existing PCA/LDA method dramatically increased to 13.3 % for four and five fingerings. Increasing the number of units and adding max-pooling to the generated feature further improved the performance and achieved error rates of up to 1.18 % for all cases. In addition, a comparison of classifier performance between SVM and RF showed that RF is generally more suitable for sparse features and is more robust against the number of classes.

The proposed system is well suited for potential use as a flute-fingering detection application for flute education. The window/hop size for the spectrogram and the max-pooling size for the feature activation were determined while considering real-time flute transcriptions. This could be used in multiple recording environments with mobile devices because the system does not require excessive computational power or extensive parameter tuning related to the recording environment. In addition, this framework can be easily applied to other instruments or timbre analysis tasks with minor changes because it does not use the deterministic rule but learns the differences from the input signal.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was supported partly by the Ministry of Science, ICT (Information and Communication Technologies) and Future Planning by the Korean Government (NRF-2014K2A2A6049835), and partly by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. 2014R1A2A2A04002619).

Received: 12 March 2015 Accepted: 3 December 2015
Published online: 21 January 2016

References
1. JW Coltman, Sounding mechanism of the flute and organ pipe. J. Acoust. Soc. Am. **44**, 983–992 (1968)
2. J Wolfe, J Smith, J Tann, NH Fletcher, Acoustic impedances of classical and modern flutes. J. Sound Vib. **243**, 127–144 (2001)
3. JR Smith, N Henrich, J Wolfe, The acoustic impedance of the Boehm flute: standard and some non-standard fingerings. Proc. Inst. Acoust. **19**, 315–330 (1997)
4. J Ngiam, PW Koh, Z Chen, S Bhaskar, AY Ng, Sparse filtering, in *Proceedings of the Advances in Neural Information Processing Systems 18 (NIPS'11)*, 2011, pp. 1125–1133
5. Y Han, K Lee, Hierarchical approach to detect common mistakes of beginner flute players, in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR'14)*, 2014, pp. 77–82
6. P Hamel, D Eck, Learning features from music audio with deep belief networks, in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR'10)*, 2010, pp. 339–344
7. H Lee, P Pham, Y Largman, AY Ng, *Unsupervised feature learning for audio classification using convolutional deep belief networks*. Adv. Neural Inf. Process Syst, 2009, pp. 1096–1104
8. SJ Maclagan, *A dictionary for the modern flutist* (Scarecrow Press, Plymouth, 2009), p. 19
9. C Kereliuk, B Scherrer, V Verfaille, P Depalle, MM Wanderley, Indirect acquisition of fingerings of harmonic notes on the flute, in *Proceedings of the International Computer Music Conference (ICMC'07)*, 2007, pp. 263–266
10. V Verfaille, P Depalle, MM Wanderley, Detecting overblown flute fingerings from the residual noise spectrum. J. Acoust. Soc. Am. **127**, 534–541 (2010)
11. H Henaff, K Jarrett, K Kavukcuoglu, Y LeCun, Unsupervised learning of sparse features for scalable audio classification, in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, 2011, pp. 77–82
12. J Schülter, C Osendorfer, Music similarity estimation with the mean-covariance restricted Boltzmann machine, in *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA'11)*, 2011, pp. 118–123
13. J Nam, J Ngiam, H Lee, M Slaney, A classification-based polyphonic piano transcription approach using learned feature representations, in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, 2011, pp. 175–180
14. J Nam, J Herrera, M Slaney, JO Smith, Learning sparse feature representations for music annotation and retrieval, in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12)*, 2012, pp. 565–570
15. KW Berger, Some factors in the recognition of timbre. J. Acoust. Soc. Am. **36.10**, 1888–1891 (1964)
16. P Hamel, S Lemieux, Y Bengio, D Eck, Temporal pooling and multiscale learning for automatic annotation and ranking of music audio, in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, 2011, pp. 729–734
17. A Coates, B Carpenter, C Case, S Satheesh, B Suresh, T Wang, DJ Wu, AY Ng, Text detection and character recognition in scene images with unsupervised feature learning, in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'11)*, 2011, pp. 440–445
18. GE Hinton, S Osindero, YW Teh, A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
19. P Vincent, H Larochelle, Y Bengio, PA Manzagol, Extracting composing robust features with denoising autoencoders, in *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, 2008, pp. 1096–1103
20. BA Olshausen, DJ Field, Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis. Res. **37**, 3311–3325 (1997)
21. M Schmidt. minFunc. http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html, 2005.
22. A Statnikov, L Wang, CF Aliferis, A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinf. **9.1**, 319 (2008)
23. L Breiman, Random forests. Mach. Learn. **45.1**, 5–32 (2001)
24. R Caruana, A Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, 2006, pp. 161–168
25. DC Miller, The influence of the material of wind-instruments on the tone quality. Science **29**, 161–171 (1909)
26. LJP van der Maaten, GE Hinton, Visualizing high dimensional data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)