# Learning temporal features using a deep neural network and its application to music genre classification

**Conference Paper** · August 2016

CITATIONS
31

READS
1,679

**2 authors:**

Il-Young Jeong
Seoul National University
**9** PUBLICATIONS **128** CITATIONS

SEE PROFILE

Kyogu Lee
Seoul National University
**157** PUBLICATIONS **1,627** CITATIONS

SEE PROFILE

# LEARNING TEMPORAL FEATURES USING A DEEP NEURAL NETWORK AND ITS APPLICATION TO MUSIC GENRE CLASSIFICATION

**Il-Young Jeong and Kyogu Lee**
Music and Audio Research Group
Graduate School of Convergence Science and Technology, Seoul National University, Korea
{finejuly, kglee}@snu.ac.kr

## ABSTRACT

In this paper, we describe a framework for temporal feature learning from audio with a deep neural network, and apply it to music genre classification. To this end, we revisit the conventional spectral feature learning framework, and reformulate it in the cepstral modulation spectrum domain, which has been successfully used in many speech and music-related applications for temporal feature extraction. Experimental results using the GTZAN dataset show that the temporal features learned from the proposed method are able to obtain classification accuracy comparable to that of the learned spectral features.

## 1. INTRODUCTION

Extracting features from audio that are relevant to the task at hand is a very important step in many music information retrieval (MIR) applications, and the choice of features has a huge impact on the performance. For the past decades, numerous features have been introduced and successfully applied to many different kinds of MIR systems. These audio features can be broadly categorized into two groups: 1) spectral and 2) temporal features.

Spectral features (SFs) represent the spectral characteristics of music in a relatively short period of time. In a musical sense, it can be said to reveal the timbre or tonal characteristics of music. Some of popular SFs include: spectral centroid, spectral spread, spectral flux, spectral flatness measure, mel-frequency cepstral coefficients (MFCCs) and chroma. On the other hand, temporal features (TFs) describe the relatively long-term dynamics of a music signal over time such as temporal transition or rhythmic characteristics. These include zero-crossing rate (ZCR), temporal envelope, tempo histogram, and so on. The two groups are not mutually exclusive, however, and many MIR applications use a combination of many different features.

The abovementioned features - be it spectral or temporal - have one thing in common: they are all 'hand-crafted' features, which are highly based on the domain knowledge or signal processing techniques. With the rapid advances in the field of machine learning and deep learning in particular, however, more recent works have become less dependent of using the standard audio features but instead try to 'learn' optimal features [4]. These approaches usually take no preprocessing step [1] or least, such as a magnitude spectrum [2,12] or mel-scale filter banks [1,10], but just let the machine learn the optimal features for a given task. Although a number of feature learning approaches have been proposed so far for many MIR-related applications, most of them have focused on learning SFs for a short-time signal [2, 12]. In case of TFs, on the other hand, few studies tried to apply deep learning models but it was limited to training the classification model from the high-level features [11, 14].

In this paper, we endeavor to learn TFs using a deep neural network (DNN) from a low-level representation. By reversing the conventional SF learning and temporal aggregation, we aim to learn TFs for a narrow spectral band and summarize them by using spectral aggregation. Furthermore, we parallelize SF and TF learning frameworks, and combine the two resulting features to use as a front end to a genre classification system. We expect this approach to provide a performance gain because each learned feature conveys different types of information present in a musical audio.

## 2. CONVENTIONAL FRAMEWORK FOR SPECTRAL FEATURE LEARNING

In this section, we briefly revisit how SFs are extracted using a DNN in a typical classification framework [12]. Figure 1 (a) shows the block diagram of its overall framework, which is similar to the proposed method for temporal feature learning except the input representation and feature aggregation. Let $s_i$ be a single channel waveform of $i$-th music data with a label $y_i$. Here, the label can be various high-level descriptor, including genre, mood, artist, chord, and tag. A magnitude spectrogram of $s_i$, $X_i$, is computed using short-time Fourier transform (STFT) defined by

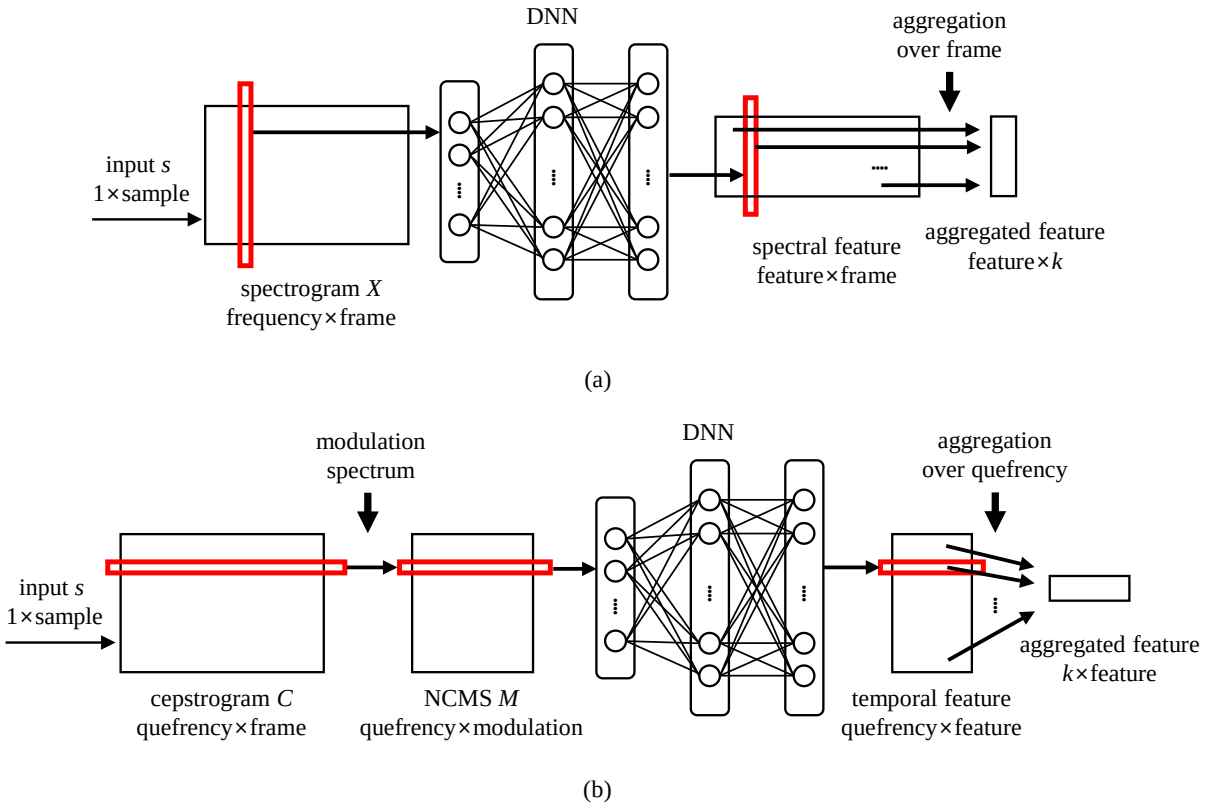



**Figure 1**. Overall frameworks for (a) conventional spectral feature learning and (b) proposed temporal feature learning. Some details (*e.g.* normalization) are omitted. $k$ denotes the number of aggregation methods. All figures in the paper are best viewed in color.

$$X_i(f,t) = \left| \sum_{n=0}^{N-1} s_i\left(\Lambda t + n\right) w\left(n\right) \exp\left(-j\frac{2\pi f n}{N}\right) \right|, \tag{1}$$

where $f$ and $t$ denote a index of frequency bin and time frame, and $N$ and $\Lambda$ indicate a size and a shift of window function $w$, respectively. $|\cdot|$ denotes the absolute operator. A different time-frequency representation such as mel-spectrogram is also widely used [1, 10].

In order to remove the bias and reduce the variance, $X$ is normalized that all frequency bins have zero-mean and unit variance across all the frames in training data as follows:

$$\bar{X}_i(f,t) = \frac{X_i(f,t) - \mu_X(f)}{\sigma_X(f)}, \tag{2}$$

where $\mu_X(f)$ and $\sigma_X(f)$ denote mean and standard deviation of the magnitude spectrogram of training data in $f$-th frequency bin, respectively. Sometimes amplitude compression or PCA whitening is added to a preprocessing step [10].

The training scheme is to learn a DNN model so that each normalized spectrum $\bar{x}_{i,t} = [\bar{X}_i(0,t), \ldots, \bar{X}_i(N/2,t)]$ ($N/2$ instead of $N$ due to its symmetry) belongs to the target class of the original input $y_i$. In other words, it can be considered as a frame-wise classification model. After training the DNN, the activations of the hidden layers are used as features.

Because many high-level musical descriptors cannot be defined within a very short segment of time, the frame-wise features usually go through a feature aggregation step before classification. The aggregation is done within the specific time range, typically 3-6s, and depending on the applications various aggregation methods exist, including mean, variance, maximum, minimum, or moments [3]. The dimension of the final feature depends on the number of aggregation methods.

To summarize, the above spectral feature extraction framework for musical applications has three steps: 1) preprocessing (STFT, normalization), 2) feature learning (DNN), and 3) temporal aggregation (*e.g.*, average and variance over frames). In the next section, we propose how each step can be modified to extract the temporal features.

## 3. PROPOSED FRAMEWORK

In this section, we present the proposed method for temporal feature learning using the normalized cepstral modulation spectrum (normalized CMS or NCMS) and DNN. Overall procedure is illustrated in Figure 1 (b).

### 3.1 Normalized cepstral modulation spectrum

We first transform a music signal to the quefrency- normalized version of CMS [8,9] because a cepstrogram is shown to be a more robust representation to capture the dynamics of the overall timbre than a spectrogram. Although there are some variations of CMS such as mel-cepstrum modulation spectrum [15], we expect that CMS is able to minimize the information loss in the procedure. To compute the NCMS, the magnitude spectrogram in Eq. (1) is first transformed into a cepstrogram domain, which is harmonic decomposition of a logarithmic magnitude spectrum using inverse discrete Fourier transform (DFT). A cepstrogram is computed from a magnitude spectrogram $X$ as follows:

$$C_i(q,t) = \frac{1}{N}\sum_{f=0}^{N-1} \ln\left(X_i(f,t)+\varepsilon\right)\exp\left(j\frac{2\pi qf}{N}\right), \quad (3)$$

where $q$ is a quefrency index, and $\varepsilon$ is a small constant to regularize a $log$ operation. In this work, we empirically set $\varepsilon$ to be $10^{-4}$.

Similar to spectrogram normalization shown in Eq. (2), cepstrogram is normalized so as to have zero-mean and unit variance across quefencies:

$$\bar{C}_i(q,t) = \frac{C_i(q,t)-\mu_C(q)}{\sigma_C(q)}, \quad (4)$$

where $\mu_C(q)$ and $\sigma_C(q)$ denote mean and standard deviation of $q$-th quefrency bin in a cepstrogram of training data, respectively.

To analyze the temporal dynamics from the data, the shift invariance has to be considered since the extracted TFs are expected to be robust against its absolute location in time or phase. Some approaches were proposed for this purpose, such as $l_2$-pooling [5], but we chose a modulation spectrum because it is simpler to compute. In addition, modulation spectral characteristics can be analyzed over a few seconds instead of a whole signal, and thus are suitable for efficiently analyzing the local characteristics. The modulation spectrum of normalized cepstrogram is obtained as follows:

$$M_i(q,v,u) = \left|\sum_{t=u\Phi}^{u\Phi+T-1} \bar{C}_i(q,t)\exp\left(-j\frac{2\pi vt}{T}\right)\right|, \quad (5)$$

where $v$ denotes the index of modulation frequency bin and $u$ is the index of the sliding window that is $T$ frames long with a $\Phi$ frames shift.

Finally, before being used as an input to a DNN, $M$ is normalized for each modulation frequency $v$ to have zero-mean and unit variance as in Eq. (2) as follows:

$$\bar{M}_i(q,v,u) = \frac{M_i(q,v,u)-\mu_M(v)}{\sigma_M(v)}, \quad (6)$$

where $\mu_M(v)$ and $\sigma_M(v)$ denote mean and standard deviation of $v$-th modulation frequency over the training data.

### 3.2 Temporal feature learning using deep neural network

The next step for temporal feature learning is the same as that of the spectral feature learning. The only difference is that an input vector of the DNN is now a normalized cepstral modulation spectrum $\bar{m}_{i,q,u} = [\bar{M}_i(q,0,u),\ldots,\bar{M}_i(q,T/2,u)], 0 \le q \le N/2$ which we expect better describes the long-term temporal properties over time for each quefrency.

### 3.3 Feature aggregation and combination

The output of a DNN in the previous section is a quefrency-wise feature, and therefore we need to aggregate it to be more appropriate as a front end to a classifier. We use the same aggregation method - *i.e.*, mean and variance - as we do in SF aggregation but only across quefrencies this time.

We believe that SFs described in Section 2 and TFs explained above represent the musical characteristics from different perspectives that can complement each other. By setting the time window size for temporal aggregation in SF to be same as that for modulation analysis in TF, say 5s, we can combine the two features and construct a complementary feature set.

In the following section, we test the effectiveness of the proposed approach and present the results obtained using a benchmark music dataset.

## 4. EXPERIMENTS

### 4.1 Data preparation

To evaluate the proposed TFs and compare it with conventional SFs, we conducted genre classification task with the GTZAN database, which consists of 1,000 30-second long music clips with the sampling rate of 22,050 Hz [16]. Each clip is annotated with one of 10 genres and for each genre there are 100 clips. Even though some drawbacks and limits were indicated [13], it is still one of the most widely used datasets for music genre classification.

We examined the two different partitioning methods. First, we randomly divided the data into three groups: 50% for training, 25% for validation, and 25% for testing, maintaining the balance among genres. We performed the experiment four times to present the averaged results. This random partitioning gaurantees that the equal number of music clips is distributed among the different genres. However, random partitioning of the GTZAN dataset may lead to the numerical evaluation results that cannot be trusted because many clips in the GTZAN dataset are from the same artists. Therefore, we also tried the 'fault-filtered' partitioning, which manually divides the dataset into 443/ 197/290 to avoid the repetition of artist across training, validation, and testing sets [6].

### 4.2 Parameter setting

Parameters in the proposed framework are basically inspired from the conventional work [12]. For STFT, we

used Hanning window of $N$=1024 samples with half overlap of $\Lambda$=512. For NCMS, the number of frames and shift to analyze the temporal dynamics were set to be $T$=214 and $\Phi$=107, respectively, which is the closest to 5s and 2.5s, respectively. The number of input units for DNN is thus 513 and 108, respectively, due to its symmetry. DNN is designed to have 3 hidden layers and each layer has 50 units for both spectral and temporal model. In other words, the network has a size of 513-50-50-50-10 for SF and 108-50-50-50-10 for TF. Rectified linear unit (ReLU) that is defined as $f(x) = max(0, x)$ was applied for the nonlinearity in every hidden layer, and the softmax function was used for the output layer. We did not use dropout or regularization terms since it did not help to improve the accuracy in our work, which is similar as previous work [12].

DNN was trained using mini-batch gradient descent with 0.01 step size and 100 batch size for both conventional and proposed algorithm. Optimization procedure was done after 200 epoches. By means of early-stopping strategy, the model which scores the lowest cross-entropy for the validation data is decided to be a final model with 10 patience.

In the aggregation stage, the outputs in the last hidden layer were aggregated using average and variance. In case of SF, the number of frames and shift for aggregation are set to be 214 and 107, respectively, which are the same as the temporal modulation window for TF. Although conventional studies also tried more complex model with various settings [2, 12], such as increasing the number of hidden units and aggregating with all the hidden layer, in this work we did not consider this kind of model settings since it is out of our scope. As shown in Figure 3, the proposed model with a simple setting already exceed the classification accuracy of the conventional approach with more complex model.

### 4.3 Genre classification

We performed genre classification using random forest (RF) with 500 trees as a classifier. Each music clip of 30s was first divided into a number of 5s-long short segments with 2.5s overlap. We then performed classification on each 5s-long segment, and used majority voting to classify the whole music clip. It is noted that both training and validation data were used to train RF since it does not require additional data for validation. The entire classification process, including training and testing, is illustrated in Figure 2.

Detailed results for each genre with the two partitioning methods are shown in Figure 3 and Figure 4. In case of random partitioning, overall accuracy of 72.6% was obtained using TFs, and 78.2% using SFs, respectively. The accuracy improved up to 85.0% when the two features are jointly used. Moreover, the combined features achieved the highest F-scores for all the genres except classical. Theses results suggest that the each type of feature contains information that helps improve genre classification.

With fault-filtered partitioning, the accuracy decreased in general, which is consistent with the results presented in [6]. Contrary to random partioning, however, the proposed TFs show much higher accuracy of 65.9% compared to 48.3% of SFs. Considering that the main difference between random and fault-filtered partitioning is artist repetition across train, validation and test sets, a possible explanation for this is that SFs are a better representation that captures similarity between the songs by the same artists. From the combined features, we obtained 59.7% accuracy which is lower than TFs alone. We believe that this unexpected performance degradation is due to the fact that the results were obtained from one trial with a fixed partition, which may have caused a bias. From an additional experiment where the classifier was trained using the training and testing sets and tested on the validation set, we obtained 50.3%, 57.4%, and 63.5% accuracies from SFs, TFs, and combined features, respectively.

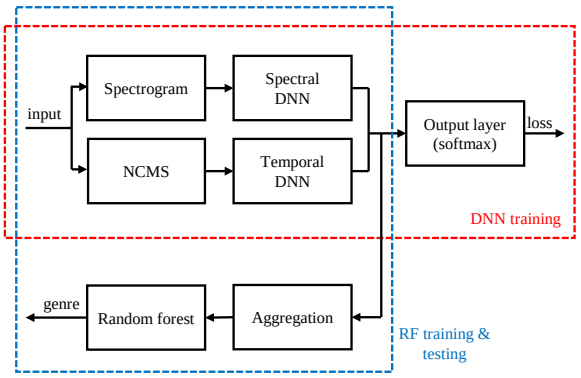



**Figure 2**. Overall framework for genre classification using conventional spectral features [12] and proposed temporal features.

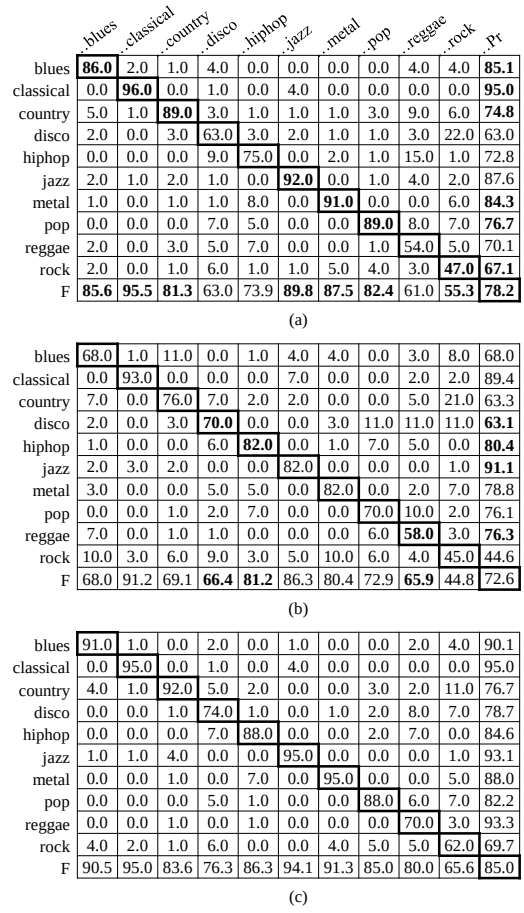
### 4.4 Feature visualization

To visually inspect the performance of different features, we visualized the features from test data using a 2-dimensional projection with t-sne [7]. Figure 5 and Figure 6 show the scatter plots of three different features, using random and fault-filtered partitioning, respectively. Although the classification accuracies are higher with random partitioning, it is not clearly represented in the figures. This may suggest that the higher performance with random partitioning is because of artist repetition, as explained in Section 4.3.

### 4.5 Discussion

Although the experimental results presented in the previous section are not sufficient to draw a firm conclusion, we can find some insights from our study worthy of further discussions. First, musical audio is an intrinsically time-varying signal, and understanding temporal dynamics is critical to better represent music. This has been done in various ways but we have demonstrated that using a more appropriate representation from the start helps achieve better performance.

The suitable domain for the analysis of temporal characteristics also leaves a room for more in-depth discus-



| | ..blues | ..classical | ..country | ..disco | ..hiphop | ..jazz | ..metal | ..pop | ..reggae | ..rock | ..Pr |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | **86.0** | 2.0 | 1.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 4.0 | **85.1** |
| classical | 0.0 | **96.0** | 0.0 | 1.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | **95.0** |
| country | 5.0 | 1.0 | **89.0** | 3.0 | 1.0 | 1.0 | 1.0 | 3.0 | 9.0 | 6.0 | **74.8** |
| disco | 2.0 | 0.0 | 3.0 | 63.0 | 3.0 | 2.0 | 1.0 | 1.0 | 3.0 | 22.0 | 63.0 |
| hiphop | 0.0 | 0.0 | 0.0 | 9.0 | 75.0 | 0.0 | 2.0 | 1.0 | 15.0 | 1.0 | 72.8 |
| jazz | 2.0 | 1.0 | 2.0 | 1.0 | 0.0 | **92.0** | 0.0 | 1.0 | 4.0 | 2.0 | 87.6 |
| metal | 1.0 | 0.0 | 1.0 | 1.0 | 8.0 | 0.0 | **91.0** | 0.0 | 0.0 | 6.0 | **84.3** |
| pop | 0.0 | 0.0 | 0.0 | 7.0 | 5.0 | 0.0 | 0.0 | **89.0** | 8.0 | 7.0 | **76.7** |
| reggae | 2.0 | 0.0 | 3.0 | 5.0 | 7.0 | 0.0 | 0.0 | 1.0 | 54.0 | 5.0 | 70.1 |
| rock | 2.0 | 0.0 | 1.0 | 6.0 | 1.0 | 1.0 | 5.0 | 4.0 | 3.0 | **47.0** | **67.1** |
| F | **85.6** | **95.5** | **81.3** | 63.0 | 73.9 | **89.8** | **87.5** | **82.4** | 61.0 | **55.3** | **78.2** |

(a)

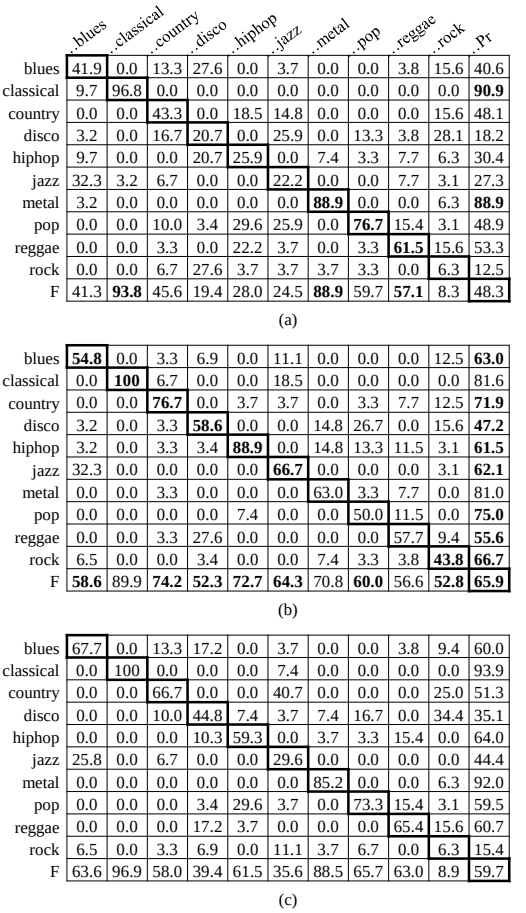
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | 68.0 | 1.0 | 11.0 | 0.0 | 1.0 | 4.0 | 4.0 | 0.0 | 3.0 | 8.0 | 68.0 |
| classical | 0.0 | 93.0 | 0.0 | 0.0 | 0.0 | 7.0 | 0.0 | 0.0 | 2.0 | 2.0 | 89.4 |
| country | 7.0 | 0.0 | 76.0 | 7.0 | 2.0 | 2.0 | 0.0 | 0.0 | 5.0 | 21.0 | 63.3 |
| disco | 2.0 | 0.0 | 3.0 | **70.0** | 0.0 | 0.0 | 3.0 | 11.0 | 11.0 | 11.0 | **63.1** |
| hiphop | 1.0 | 0.0 | 0.0 | 6.0 | **82.0** | 0.0 | 1.0 | 7.0 | 5.0 | 0.0 | **80.4** |
| jazz | 2.0 | 3.0 | 2.0 | 0.0 | 0.0 | 82.0 | 0.0 | 0.0 | 0.0 | 1.0 | **91.1** |
| metal | 3.0 | 0.0 | 0.0 | 5.0 | 5.0 | 0.0 | 82.0 | 0.0 | 2.0 | 7.0 | 78.8 |
| pop | 0.0 | 0.0 | 1.0 | 2.0 | 7.0 | 0.0 | 0.0 | 70.0 | 10.0 | 2.0 | 76.1 |
| reggae | 7.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 6.0 | **58.0** | 3.0 | **76.3** |
| rock | 10.0 | 3.0 | 6.0 | 9.0 | 3.0 | 5.0 | 10.0 | 6.0 | 4.0 | 45.0 | 44.6 |
| F | 68.0 | 91.2 | 69.1 | **66.4** | **81.2** | 86.3 | 80.4 | 72.9 | **65.9** | 44.8 | 72.6 |

(b)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | 91.0 | 1.0 | 0.0 | 2.0 | 0.0 | 1.0 | 0.0 | 0.0 | 2.0 | 4.0 | 90.1 |
| classical | 0.0 | 95.0 | 0.0 | 1.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 95.0 |
| country | 4.0 | 1.0 | 92.0 | 5.0 | 2.0 | 0.0 | 0.0 | 3.0 | 2.0 | 11.0 | 76.7 |
| disco | 0.0 | 0.0 | 1.0 | 74.0 | 1.0 | 0.0 | 1.0 | 2.0 | 8.0 | 7.0 | 78.7 |
| hiphop | 0.0 | 0.0 | 0.0 | 7.0 | 88.0 | 0.0 | 0.0 | 2.0 | 7.0 | 0.0 | 84.6 |
| jazz | 1.0 | 1.0 | 4.0 | 0.0 | 0.0 | 95.0 | 0.0 | 0.0 | 0.0 | 1.0 | 93.1 |
| metal | 0.0 | 0.0 | 1.0 | 0.0 | 7.0 | 0.0 | 95.0 | 0.0 | 0.0 | 5.0 | 88.0 |
| pop | 0.0 | 0.0 | 0.0 | 5.0 | 1.0 | 0.0 | 0.0 | 88.0 | 6.0 | 7.0 | 82.2 |
| reggae | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 70.0 | 3.0 | 93.3 |
| rock | 4.0 | 2.0 | 1.0 | 6.0 | 0.0 | 0.0 | 4.0 | 5.0 | 5.0 | 62.0 | 69.7 |
| F | 90.5 | 95.0 | 83.6 | 76.3 | 86.3 | 94.1 | 91.3 | 85.0 | 80.0 | 65.6 | 85.0 |

(c)

**Figure 3**. Figure of merit (FoM, $\times 100$) with random partitioning for (a) the conventional spectral features, (b) the proposed temporal features, and (c) the combined features. Each row and column represents the predicted and true genres respectively. The elements in the matrix denote the recall (diagonal), precision (last column), F-score (last row), confusions (off-diagonal), and overall accuracy (the last element of diagonal). The higher values of recall, precision, and F-score between (a) and (b) are emphasized in bold.



| | ..blues | ..classical | ..country | ..disco | ..hiphop | ..jazz | ..metal | ..pop | ..reggae | ..rock | ..Pr |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | 41.9 | 0.0 | 13.3 | 27.6 | 0.0 | 3.7 | 0.0 | 0.0 | 3.8 | 15.6 | 40.6 |
| classical | 9.7 | 96.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **90.9** |
| country | 0.0 | 0.0 | 43.3 | 0.0 | 18.5 | 14.8 | 0.0 | 0.0 | 0.0 | 15.6 | 48.1 |
| disco | 3.2 | 0.0 | 16.7 | 20.7 | 0.0 | 25.9 | 0.0 | 13.3 | 3.8 | 28.1 | 18.2 |
| hiphop | 9.7 | 0.0 | 0.0 | 20.7 | 25.9 | 0.0 | 7.4 | 3.3 | 7.7 | 6.3 | 30.4 |
| jazz | 32.3 | 3.2 | 6.7 | 0.0 | 0.0 | 22.2 | 0.0 | 0.0 | 7.7 | 3.1 | 27.3 |
| metal | 3.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **88.9** | 0.0 | 0.0 | 6.3 | **88.9** |
| pop | 0.0 | 0.0 | 10.0 | 3.4 | 29.6 | 25.9 | 0.0 | **76.7** | 15.4 | 3.1 | 48.9 |
| reggae | 0.0 | 0.0 | 3.3 | 0.0 | 22.2 | 3.7 | 0.0 | 3.3 | **61.5** | 15.6 | 53.3 |
| rock | 0.0 | 0.0 | 6.7 | 27.6 | 3.7 | 3.7 | 3.7 | 3.3 | 0.0 | 6.3 | 12.5 |
| F | 41.3 | **93.8** | 45.6 | 19.4 | 28.0 | 24.5 | **88.9** | 59.7 | **57.1** | 8.3 | 48.3 |

(a)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | **54.8** | 0.0 | 3.3 | 6.9 | 0.0 | 11.1 | 0.0 | 0.0 | 0.0 | 12.5 | **63.0** |
| classical | 0.0 | **100** | 6.7 | 0.0 | 0.0 | 18.5 | 0.0 | 0.0 | 0.0 | 0.0 | 81.6 |
| country | 0.0 | 0.0 | **76.7** | 0.0 | 3.7 | 3.7 | 0.0 | 3.3 | 7.7 | 12.5 | **71.9** |
| disco | 3.2 | 0.0 | 3.3 | **58.6** | 0.0 | 0.0 | 14.8 | 26.7 | 0.0 | 15.6 | **47.2** |
| hiphop | 3.2 | 0.0 | 3.3 | 3.4 | **88.9** | 0.0 | 14.8 | 13.3 | 11.5 | 3.1 | **61.5** |
| jazz | 32.3 | 0.0 | 0.0 | 0.0 | 0.0 | **66.7** | 0.0 | 0.0 | 0.0 | 3.1 | **62.1** |
| metal | 0.0 | 0.0 | 3.3 | 0.0 | 0.0 | 0.0 | 63.0 | 3.3 | 7.7 | 0.0 | 81.0 |
| pop | 0.0 | 0.0 | 0.0 | 0.0 | 7.4 | 0.0 | 0.0 | 50.0 | 11.5 | 0.0 | **75.0** |
| reggae | 0.0 | 0.0 | 3.3 | 27.6 | 0.0 | 0.0 | 0.0 | 0.0 | 57.7 | 9.4 | **55.6** |
| rock | 6.5 | 0.0 | 0.0 | 3.4 | 0.0 | 0.0 | 7.4 | 3.3 | 3.8 | **43.8** | **66.7** |
| F | **58.6** | 89.9 | **74.2** | **52.3** | **72.7** | **64.3** | 70.8 | **60.0** | 56.6 | **52.8** | **65.9** |

(b)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| blues | 67.7 | 0.0 | 13.3 | 17.2 | 0.0 | 3.7 | 0.0 | 0.0 | 3.8 | 9.4 | 60.0 |
| classical | 0.0 | 100 | 0.0 | 0.0 | 0.0 | 7.4 | 0.0 | 0.0 | 0.0 | 0.0 | 93.9 |
| country | 0.0 | 0.0 | 66.7 | 0.0 | 0.0 | 40.7 | 0.0 | 0.0 | 0.0 | 25.0 | 51.3 |
| disco | 0.0 | 0.0 | 10.0 | 44.8 | 7.4 | 3.7 | 7.4 | 16.7 | 0.0 | 34.4 | 35.1 |
| hiphop | 0.0 | 0.0 | 0.0 | 10.3 | 59.3 | 0.0 | 3.7 | 3.3 | 15.4 | 0.0 | 64.0 |
| jazz | 25.8 | 0.0 | 6.7 | 0.0 | 0.0 | 29.6 | 0.0 | 0.0 | 0.0 | 0.0 | 44.4 |
| metal | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 85.2 | 0.0 | 0.0 | 6.3 | 92.0 |
| pop | 0.0 | 0.0 | 0.0 | 3.4 | 29.6 | 3.7 | 0.0 | 73.3 | 15.4 | 3.1 | 59.5 |
| reggae | 0.0 | 0.0 | 0.0 | 17.2 | 3.7 | 0.0 | 0.0 | 0.0 | 65.4 | 15.6 | 60.7 |
| rock | 6.5 | 0.0 | 3.3 | 6.9 | 0.0 | 11.1 | 3.7 | 6.7 | 0.0 | 6.3 | 15.4 |
| F | 63.6 | 96.9 | 58.0 | 39.4 | 61.5 | 35.6 | 88.5 | 65.7 | 63.0 | 8.9 | 59.7 |

(c)

**Figure 4**. Figure of merit (FoM, $\times 100$) with fault-filtered partitioning. Details are the same as Figure 3.

sion. While NCMS shows good performance in our experiments, it is probable that there exists a representation that can better describe temporal properties in music. One possible way would be analyzing temporal dynamics of SFs learned from DNN. It might be able to minimize the feature extraction step, and the process should be simpler by concatenating the spectral/temporal DNNs in series.

## 5. CONCLUSION

In this paper, we presented a novel feature learning framework using a deep neural network. In particular, while most studies have been trying to learn the spectral features from a short music segment, we focused on learning the features that represent the long-term temporal characteristics, which are expected to convey different information from that in the conventional spectral features. To this end, we used a normalized cepstral modulation spectrum as an input to DNN, and introduced a feature aggregation method over quefrencies. Experiments with genre classification show that the proposed temporal features yielded performance comparable to or better than that of the spectral features, depending on the partitioning methods of the dataset. We plan to apply the proposed method to various MIR-related tasks, including mood classification or instrument identification where spectral features are predominantly used. We also intend to develop a single framework in which both spectral and temporal features are jointly learned.

## 6. ACKNOWLEDGEMENTS

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-H8501-16-1016) supervised by the IITP (Institute for Information & communications Technology Promotion).

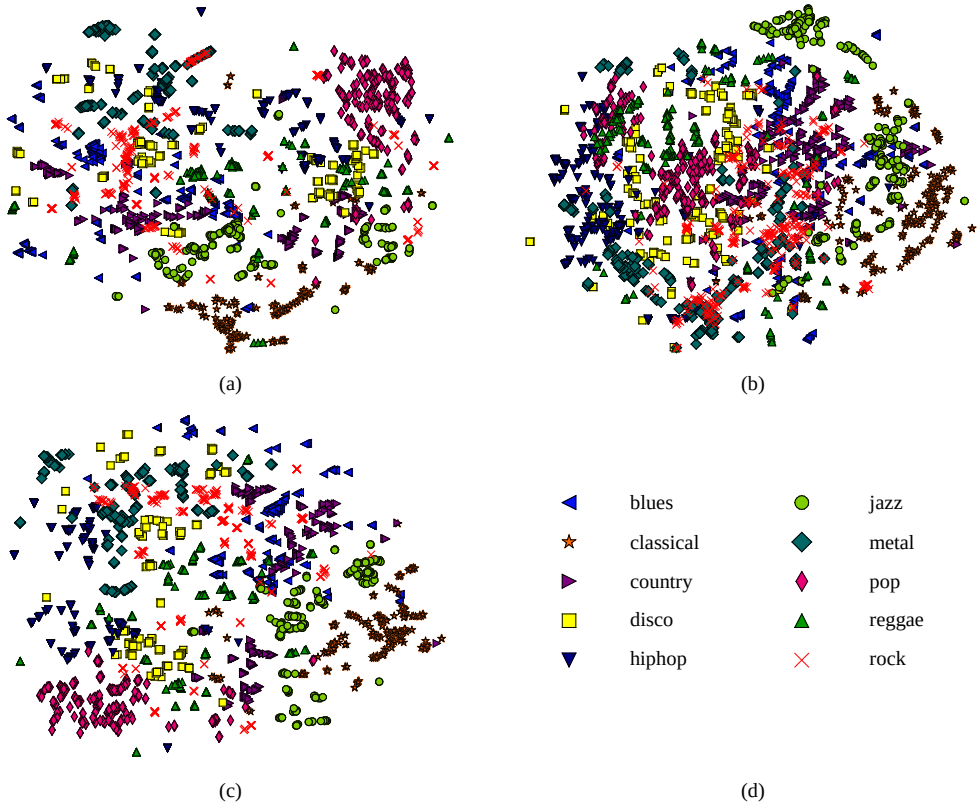



**Figure 5**. 2-dimensional scatter plots using t-sne [7] with random partitioning for (a) the conventional spectral features, (b) the proposed temporal features, and (c) the combined features. Each marker represents a 5s excerpt of a music signal whose genre is labeled as in (d).

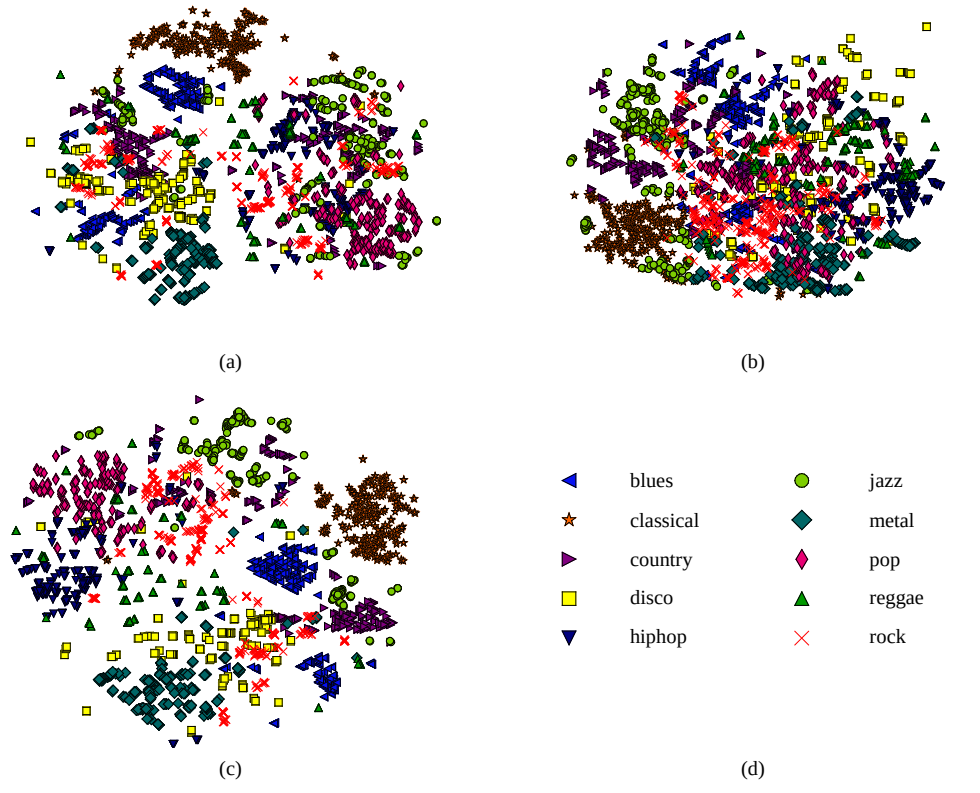



**Figure 6**. 2-dimensional scatter plots using t-sne [7] with fault-filtered partitioning. Details are the same as Figure 5.

## 7. REFERENCES


[1] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. Florence, Italy, 2014.

[2] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, pages 339–344. Utrecht, Netherlands, 2010.

[3] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *ISMIR*, pages 729–734, 2011.

[4] Eric J. Humphrey, Juan P. Bello, and Yann LeCun. Feature learning and deep architectures: New directions for music informatics. *Journal of Intelligent Information Systems*, 41(3):461–481, 2013.

[5] Aapo Hyvärinen and Patrik Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

[6] Corey Kereliuk, Bob L. Sturm, and Jan Larsen. Deep learning and music adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, 2015.

[7] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[8] Rainer Martin and Anil Nagathil. Cepstral modulation ratio regression (CMRARE) parameters for audio signal analysis and classification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[9] Anil Nagathil, Timo Gerkmann, and Rainer Martin. Musical genre classification based on a highly-resolved cepstral modulation spectrum. In *Proceedings of the European Signal Processing Conference*, 2010.

[10] Juhan Nam, Jorge Herrera, Malcolm Slaney, and Julius O Smith. Learning sparse feature representations for music annotation and retrieval. In *ISMIR*, pages 565–570, 2012.

[11] Aggelos Pikrakis. A deep learning approach to rhythm modeling with applications. In *Proceedings of the International Workshop on Machine Learning and Music*, 2013.

[12] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. Florence, Italy, 2014.

[13] Bob L. Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, 2014.

[14] Bob L. Sturm, Corey Kereliuk, and Jan Larsen. ¿El caballo viejo? latin genre recognition with deep learning and spectral periodicity. *Mathematics and Computation in Music*, pages 335–346, 2013.

[15] Vivek Tyagi, Iain McCowan, Hemant Misra, and Hervé Bourlard. Mel-cepstrum modulation spectrum (MCMS) features for robust asr. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 399–404, 2003.

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