IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2020, accepted August 12, 2020, date of publication August 18, 2020, date of current version August 31, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3017661*

# Multi-Level Local Feature Coding Fusion for Music Genre Recognition

**WING W. Y. NG, (Senior Member, IEEE), WEIJIE ZENG, AND TING WANG, (Student Member, IEEE)**
School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

Corresponding author: Ting Wang (tingwang@ieee.org)

This work was supported in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the National Natural Science Foundation of China under Grant 61876066, and in part by the Guangdong Province Science and Technology Plan Project (Collaborative Innovation and Platform Environment Construction) under Grant 2019A050510006.

**ABSTRACT** Music genre recognition (MGR) plays a fundamental role in the context of music indexing and retrieval. Unlike images, music genres consist of immediate characteristics that are highly diversified with abstractions in different levels. However, most representation learning methods for MGR focus on global features and make decisions from features in the same level. To remedy such defects, we intergrate a convolutional neural network (CNN) with NetVLAD and self-attention to capture the local information across levels and learn their long-term dependencies. A meta classifier is used to make the final MGR classification by learning from aggregated high-level features from different local feature coding networks. Experimental results show that the proposed approach yields higher accuracies than other state-of-the-art models on GTZAN, ISMIR2004, and Extended Ballroom dataset.

**INDEX TERMS** Music genre recognition, NetVLAD, self-attention, convolutional neural network, representation learning.

## I. INTRODUCTION

With the increase of online music databases and user-interactive applications, developing effective automatic tools for music classification and retrieval has become an essential issue. Music information retrieval (MIR) aims to retrieve useful information from music and classify it into different categories. For MIR problems, music genre is significantly important because it facilitates both the search for music and the organization of music collections. Furthermore, music genre also reveals the interplay of cultures. Extracting influential features contributes to the automatic classification of music genres. There are a lot of approaches to extract descriptive features for music genre recognition (MGR) [1], [2]. The scatter transform and the transfer learning have been widely used for image and audio classification [3]–[6] but rarely used for MGR in combination. Various methods for building music genre classifiers have been studied, including support vector machines (SVM) [7]–[9], Gaussian processes [10], convolutional neural network (CNN) [11]–[13], recurrent neural network (RNN) [14], and long short-term memory (LSTM) [15]. It is proven that CNNs learning representation features yield better performance in comparison to LSTMs and traditional methods which extract handcrafted features [16]. In contrast, many handcrafted features have strong complementarity, but MGR classification tasks usually use only one of them. In the field of MGR, most current representation learning methods for MGR focus on global features and make decisions from features in the same level. But music genre consists of immediate characteristics which are highly diversified and have different levels of abstractions [1], [11]. Moreover, the fusion in the final ensemble is in decision-level, which may ignore the internal relationship among features at early stages [8], [9].

As a local representation method, the NetVLAD has been widely studied in recent years [17], [18]. Music streams with highly diversified characteristics have abstractions in different levels, which have different influences on understanding of genres and are often distributed locally in repeated audio clips. The self-attention mechanism determines the importance of different features by focusing on dependencies of all positions in the signal [19]. In this work, we design a feature encoding network for music stream by combining the NetVLAD and the self-attention mechanism to fully explore local features of different levels of abstractions and learn their long-term dependencies. The code for the proposed approach is available on GitHub.[1] Major contributions of this research are summarized as follows:

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

[1] https://github.com/bieqingcheng/music-genre-recognition

(i) A multi-level feature coding network using a CNN network with NetVLAD and the self-attention is proposed to capture the local information across different levels and learn their dependencies. Genres of music are locally positioned in different levels and time scales in a music stream. The NetVLAD captures the local information from each layer and aggregates them together to provide an integrated description. Then, the self-attention learns long-term dependencies across these aggregating features from different levels.
(ii) The complementary nature of the scatter transform feature and the transfer feature for the MGR are explored, which enriches the diversity of features to make representations learned by the proposed model more useful.
(iii) A meta classifier is used to learn the implicit relationship among high-level features to obtain more useful information. Then, it is retrained using aggregated heterogeneous high-level features learned by feature coding networks to improve the classification performance of the proposed ensemble model for the MGR. Experimental results also show that the proposed model yields better accuracies in comparison to other methods.

The rest of this paper is organized as follows: Section II presents related works on feature extraction and classification methods for MGR tasks. Section III describes the proposed model. Section IV shows experimental results and discussions on datasets for MGR tasks. Lastly, Section V concludes this work.

## II. RELATED WORK

Feature extraction and classification methods are major focuses of MGR tasks. This Section reviews feature extraction and classification methods for MGR tasks in Sections II-A and II-B, respectively.

### A. FEATURE EXTRACTION METHODS

Most MGR models extract audio features to achieve satisfactory performance. In general, audio features can be divided into handcrafted and non-handcrafted features. In some cases, handcrafted features capture statistical characteristics with bag-of-frames analysis to observe amplitude characteristics over time frames (e.g. timbre [20], pitch, and rhythm [21]). In other cases, handcrafted features extract spectrogram textures and their temporal variations with time-frequency analysis to describe the temporal change of energy distribution over frequency bins (e.g. Mel-spectrogram [22], [23], harmonic and percussive spectrogram [24], constant-Q transform spectrogram [25], [26], and scatter transform spectrogram [27]). The local binary patterns (LBP), local phase quantization (LPQ), Gabor filter feature extraction (GF), binarized statistical image features (BSIF), locally encoded transform feature histogram (LEN), and the codebookless model (CLM) are also used to analyze the similarity of spectrograms [8], [9], [28].

Representation learning techniques such as CNN, LSTM, and transfer learning are widely applied to obtain non-handcrafted features for MGR tasks [16], [29]. The transfer feature learning utilizes the existing knowledge to process the available data from diversified feature spaces. In [6],a transfer feature is expressed as a concatenated feature vector using activations of multi-layer feature maps in a convolutional network that is pretrained on a very large dataset (i.e. Million Song Dataset) [30].

### B. CLASSIFICATION METHODS

Classification methods of MGR tasks can be broadly divided into supervised classification and unsupervised clustering methods [1], [2]. As an instance of supervised methods for MGR, the local feature selection strategy uses a self-adaptive harmony search algorithm [7]. In recent years, many MGR solutions have shifted towards the use of deep learning, which outperforms traditional machine learning approaches on MGR tasks. The CNN representation learning outperforms hand-crafted features, being complementary to the latter [29]. Based on the fact that most energy of a spectrogram distributes over only a few temporal steps, an attention mechanism is incorporated into a bidirectional recurrent neural network (BRNN) so that all temporal steps are taken into account by assigning different weights to pay more attention to important temporal steps [15]. The BRNN, the GRU, and the BRNN-based model with parallelized and serial attentions are compared to show the effectiveness of attention mechanisms. With different levels of abstractions in music genres, a CNN-based architecture with multi-level and multi-scale features is exploited to better leverage the distributed properties of genres [11]. Furthermore, input signals are downsampled and transfer learning is utilized with the previous multi-level and multi-scale techniques [12].

The non-negative matrix factorization (NMF) [31], the non-negative tensor factorization (NTF) [32], and the sparse coding [33] are instances of unsupervised learning methods for MGR. In addition to the sparse coding, there are many feature coding methods used in image classification, including the hard coding [34], the soft coding [35], the low-rank sparse coding [36], [37], the vector of locally aggregated descriptor (VLAD) coding [18], and the Fisher vector (FV) coding [38]. However, traditional feature coding methods are unsupervised clustering-based approaches, which may not be suitable for classification tasks. The traditional VLAD coding model is extended to an end-to-end model NetVLAD by using a learned softmax for local feature assignments [17]. Parameters of the network are trained by the back-propagation algorithm. Some studies [39], [40] also explore neural networks involving clustering techniques for MGR. A bootstrapped k-means is proposed to responses [39]. A multi-layer perceptron combined with the spherical k-means algorithm is introduced to enhance the performance of transfer learning [40].

## III. PROPOSED METHOD

The overall scheme for MGR is presented in Section III-A. Descriptions of feature extraction are given in Section III-B.

**Algorithm 1** The Procedure of the Proposed Method for Music Genre Recognition

**Input**: Music dataset
**Output**: The best classification result $R_{best}$

1. Divide the original audio signal into multiple segments and extract 8 types of features for each segment.
2. For each feature build an independent feature coding network:
3. Use a network like VGG as the backbone network.
4. Extract the local features from the outputs of feature maps from the 5 blocks by Algorithm 2.
5. Aggregate these five local features as $F_{agg}$.
6. Learn the long-range dependencies of $F_{agg}$ by Algorithm 3.
7. Generate the classification result $R_F$ and the learned high-level feature $F_c$ from two fully connected layers.
8. End for
9. Initially use the sum rule to select the best 6 feature combinations as $F_6$ based on $R_F$.
10. Assign $\phi$ to the classification result set $R_{set}$.
11. For each feature combination $F_{alone}$ in $F_6$:
12. Aggregate the learned high-level features $F_c$ of $F_{alone}$ as a new high-level feature $F_{high}$.
13. Output the classification result $R_M$ by feeding $F_{high}$ to the meta classifier.
14. $R_{set} = R_{set} \cup R_M$.
15. End for
16. Select the best classification result $R_{best}$ from $R_{set}$.

The feature coding network focusing on local features across different levels is illustrated in Section III-C. The design of loss function and the fusion strategy from decision-level to feature-level for learning internal relations among high-level features at an early stage are given in Sections III-D and III-E, respectively.

### A. OVERALL SCHEME

The procedure of the whole approach is presented in Algorithm 1. In Algorithm 1, $F_{agg}$, $R_F$, and $F_c$ denote the aggregation of local features produced by 5 Netvlad layers, high-level features extracted by feature coding network, and classification result of feature coding network, respectively. $F_{agg}$, $R_F$, and $F_c$ are also shown in Figure 2. $F_6$ are the best 6 feature combinations generated by the sum rule based on $R_F$ of eight types of low-level features. $F_{high}$ and $R_M$ are an aggregation of the learned high-level features input to the meta classifier and the classification result of meta classifier, respectively. $R_{best}$ is the best classification result of the meta classifier selected from $F_6$.

Firstly, the original audio signal is divided into multiple segments. The duration of each segment for ISMIR2004 is 30 seconds while the duration of each segment for both GTZAN and Extended Ballroom is 10 seconds. Then, eight types of low-level features are extracted from these segments. These features include timbre [20], rhythm histogram features (RH) [21], statistical spectrum descriptors (SSD) and transfer features [6], Mel-spectrogram [22], harmonic spectrogram [24], percussive spectrogram, constant-$Q$ spectrogram [25] and scatter transform spectrogram [27]. Timbre, RH, and SSD capture statistical characteristics from time-frequency representation while Mel-spectrogram, harmonic spectrogram, percussive spectrogram, constant-$Q$ spectrogram, and scatter transform spectrogram are time-frequency representation using different filter technologies. Transfer feature is a learned feature extracted by a CNN with transfer learning. Figure 1 shows five types of time-frequency representations. Same with other works, we sample at 22050 samples per second for feature extraction.

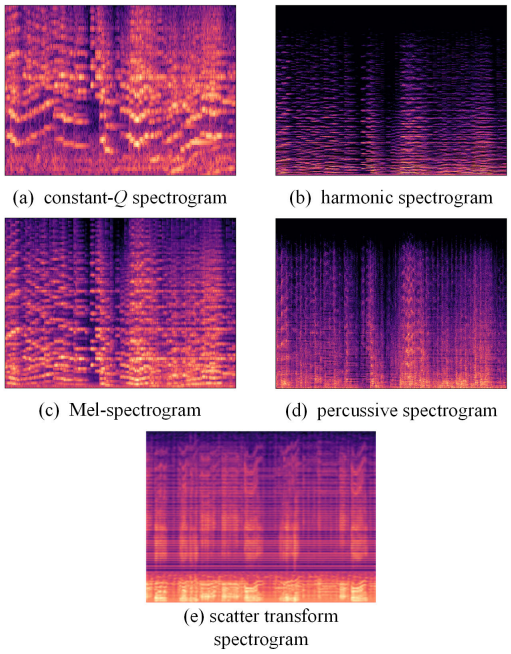


**FIGURE 1.** Five types of visual representation features.

After extraction of eight types of low-level features, high-level features are obtained by representation learning using feature coding network. In the representation learning phase, each feature coding network uses a feature as inputs. Specifically, the feature coding network uses a VGG as the backbone network which consists of 5 cascading blocks.

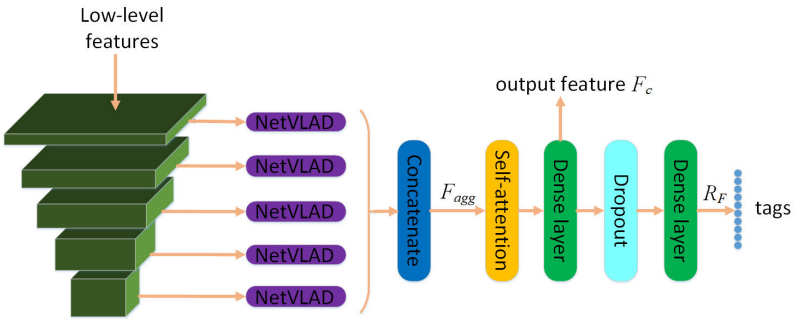


**FIGURE 2.** Block diagram of the feature coding network with NetVLAD and self-attention.

Each block consists of a convolutional layer, a batch normalization layer, an ELU activation layer, and a pooling layer. The feature map produced by each block represents features at a particular level of abstraction. Then, these 5 feature maps at different levels of abstractions are fed into 5 NetVLAD layers to extract their local features. Self-attention is applied to learn long-term dependencies among these local features across different levels. In our method, an ensemble of feature coding networks is trained where each network uses a selected low-level feature as an input. Then, a meta-CNN is trained using the learned high-level features extracted by feature coding networks. There are 247 combinations (combinations of 2 to 8 features) over 8 low-level features. A simple sum rule is applied to select the best 6 feature combinations (i.e. $F_6$) yielding the highest testing accuracies. For each feature combination in $F_6$, the learned high-level features are aggregated and then fed to a meta-CNN. Finally, the best classification result $R_{best}$ is obtained by a meta-CNN using the learned high-level features aggregation of the best feature combination as inputs.

### B. FEATURE EXTRACTION

In our work, the eight features consist of timbre, rhythm, Mel-spectrogram, harmonic spectrogram, percussive spectrogram, constant-$Q$ transform spectrogram, scattering transform spectrogram, and transfer feature, respectively.

Components of timbre feature include mean and variance of the spectral centroid, spectral flux, time-domain zero crossings, low-energy component, and 13 Mel-Frequency cepstral coefficients [20]. In this work, we used the Marsyas toolbox to generate the timbre feature. Rhythm consists of rhythm histogram feature (RH) and statistical Spectrum Descriptors (SSD) [21]. RH extracts a set of sixty acoustic characteristics, each of which corresponds to the aggregation of amplitude modulations of twenty-four critical zones' individually computed by a rhythm pattern. SSD is a set of statistical measures that describe fluctuations in zones criticisms and captures some timbre information of the twenty-four critical bands defined according to the Bark scale. Codes for RH and SSD are provided on Github.[2]

Inspired by measured responses from the human auditory system, Mel-spectrogram is computed by mapping the spectral magnitudes of short-time Fourier transform onto the perceptually motivated Mel-scale using the filterbank technique [22], [23]. The concept of anisotropic spectrogram diffusion is used to split the music signal into two portions. One is a harmonic spectrogram and the other is a percussion spectrogram [24]. Constant-$Q$ transform provides a logarithmically spaced frequency basis and a frequency resolution that depends on geometrically spaced center frequencies of the analysis windows [25], [26]. In this work, we use the Librosa python package to generate the Mel-spectrogram, harmonic spectrogram, percussive spectrogram, and constant-$Q$ transform spectrogram.

A scattering transform defines a locally translation invariant representation which is stable with respect to time-warping deformation [27]. It is defined as a convolutional network whose filters are fixed to be wavelet and low-pass averaging filters coupled with modulus nonlinearities. We extract the scatter transform spectrogram by the Kymatio python package.

Transfer learning is designed to leverage an already trained model in a relevant field. The transfer feature used in this paper comes from the transfer learning method proposed in [6]. A convolutional neural network is designed and trained for the source task and the trained network is used as a feature extractor for the target MGR task on GTZAN, ISMIR2004, and Extended Ballroom datasets. The code for transfer feature is available on GitHub.[3]

### C. MULTI-LEVEL FEATURE CODING NETWORK

The structure of the multi-level feature coding network is shown in Figure 2. The feature coding network is mainly divided into four parts: backbone network, NetVLAD layer, self-attention layer, and fully connected layer. The backbone network consists of 5 cascaded blocks with each block containing four basic units: a convolutional layer, a batch normalization (BN) layer, an ELU activation layer, and a pooling layer, respectively. Different spectral bands have different distributions, so features should be learned from different bands. Therefore, a 2D convolution is used to learn both

[2]https://github.com/tuwien-musicir/rp_extract

[3]https://github.com/keunwoochoi/transfer_learning_music

temporal and spectral structures [41]. In our method, 2D convolution networks with a kernel size of $3 \times 3$, stride size $= 1$, and padding size $= 1$ are used to build the multi-level feature coding network. The ELU is used as an activation function in every convolutional layer while the BN is applied after convolution and before activation. The number of feature maps is set as 32 in the first three blocks and 64 in the last two blocks. The number of feature maps is increased in the final two blocks so as to provide sufficient resolution of the learned features.

Extracted features from convolutional layers in 5 blocks can be expressed as $F \in \Re^{H\times W\times D}$, where $D$, $H$, and $W$ denote the number of the convolutional filters, height and width of the feature map of convolutional layers, respectively. $F$ can be considered as a descriptor set containing $H \times W$ deep descriptors with the dimension of $D$. The NetVLAD takes $F$ as inputs. $K$ cluster centers (visual words) in NetVLAD are used to construct a dictionary of VLAD. The descriptor set $F \in \{f_j\}_{j=1}^{H\times W}$ represents extracted features of the convolutional layer, where $f_j \in \Re^{D\times 1}$ is the $j^{th}$ descriptor of $F$. The final VLAD representation is a $K \times D$ vector computed by Eq. (1), where $V(F_j) \in \Re^{K\times D\times 1}$ is the VLAD vector of $f_j \in \Re^{D\times 1}$ and $V(F_j)$ can be written as Eq. (2). In Eq. (3), $a_j(k)$ is the weight coefficient of $f_j$ and $c_k$. $c_k$ is the $k^{th}$ cluster center. By softly assigning $f_j$ to the nearest cluster $c_k$, the expression of the soft assignment coding is shown in Eq. (4), where $w_k$ and $b_k$ are trainable parameters for each cluster.

$$V(F) = \sum_{j=1}^{H\times W} V(f_j) \tag{1}$$

$$V(F_j) = \left[\delta\left(f_j^1\right)^T, \delta\left(f_j^2\right)^T, \cdots, \delta\left(f_j^K\right)^T\right]^T \tag{2}$$

$$\delta\left(f_j^k\right) = a_j(k)(f_j - c_k) \tag{3}$$

$$a_j(k) = \frac{e^{w_k^T f_j + b_k}}{\sum_{i=1}^{K} e^{w_i^T f_j + b_i}} \tag{4}$$

$$V(F)(k,d) = \sum_{j=1}^{H\times W} \frac{e^{w_k^T f_j + b_k}}{\sum_{i=1}^{K} e^{w_i^T f_j + b_i}} (f_j(d) - c_k(d)) \tag{5}$$

With Eqs. (1)–(4) combined, the final form of NetVLAD layer is expressed as Eq. (5), where $V(F)(k,d)$ is the $((K-1)\times D)^{th}$ of $V(F)$ while $f_j(d)$ and $c_k(d)$ are the $d^{th}$ element of $f_j$ and $c_k$, respectively. Thus, NetVLAD coding with a size of $K \times D$ is generated by summation over the residuals between features and their corresponding center. The dictionary size $K$ has an important influence on the discriminative power, and it determines the size of coding. Because the subsequent self-attention layer consumes huge GPU memories [19], the dictionary size $K$ is set as 20 rather than a larger number. The learning procedure of NetVLAD is shown in Algorithm 2.

The music stream has more complex intrinsic patterns that are highly diversified and have different levels of abstractions. Combining multi-layer audio features helps to find the genre positioned in different time scales. Therefore, all 5 NetVLAD codings of the 5 blocks are aggregated and fed to the self-attention layer to capture long-term dependencies across different levels. The self-attention maps are derived from the aggregation $X \in \Re^{M\times K}$ of the 5 NetVLAD codings, where $K$ and $M$ denote the dictionary size and the row amount of the matrix $X$, respectively. Let $X = \{X_1, X_2, \cdots, X_M\}$ be the aggregation of NetVLAD codings, where $X_1, X_2, \cdots$, and $X_M$ are column vectors. The attention value at position $i$ is obtained by Eq. (6).

$$Y_i = \frac{1}{C(X_i)} \sum_{j=1}^{M} \varphi(X_i, X_j) g(X_j) \tag{6}$$

$$C(X_i) = \sum_{j=1}^{M} \varphi(X_i, X_j) \tag{7}$$

$$g(X_j) = W_g X_j \tag{8}$$

$$\varphi(X_i, X_j) = e^{(W_k X_i)^T (W_q X_j)} \tag{9}$$

$$Z_i = X_i + \alpha Y_i \tag{10}$$

where $W_g$, $W_k$ and $W_q$ denote the weight matrices implemented by 2D convolution with a kernel size of $1 \times 1$. $g$ is the linear function as shown in Eq. (8), and $\varphi$ denote the embedded Gaussian function to calculate the dependency between $X_i$ and $X_j$ as shown in Eq. (9). In Eq. (6), the function $\varphi$

**Algorithm 2** The Procedure of NetVLAD

**Input**: The feature map $F$ for output of each block
**Output**: NetVLAD coding $V(F)$

1. Let $V(F) = \phi$, and initialize the cluster center $c_k$, $w_i$ and $b_i$ in Eq. (4).
2. For $f_i$ in $F$ do:
3. Calculate soft assignment coding $a_j^k$ for $f_i$ relative to each cluster $c_k$ by Eq. (4).
4. Calculate residuals $f_i - c_k$ relative to each cluster.
5. For each cluster center, calculate the local feature coding $V(F_j)$ by Eq. (2) and Eq. (3).
6. $V(F) = V(F) \cup V(F_j)$.
7. Update cluster $c_k$, $w_i$ and $b_i$ in Eq. (4) with the gradient provided by back propagation.
8. End for
9. NetVLAD coding is obtained by applying intra-normalization and $L_2$ normalization to $V(F)$.

**Algorithm 3** The Procedure of Self-Attention Learning

**Input**: The aggregation $X$ of NetVLAD codings $V(F)$
**Output**: Self-attention features $Z$

1. Let $Z = \phi$, and initialize the weights of $W_k$, $W_g$ and $W_q$ in Eq. (8) and Eq. (9).
2. For $n = 1, 2, 3, \cdots, N$ in do:
3. For $X_i$ in $X$ do:
4. Calculate the dependency between $X$ at any position $j$ and $X$ at current position $i$ by Eq. (9).
5. Calculate the intensities of $X$ at any position $j$ by Eq. (8).
6. Calculate the attention value of $X$ at current position $i$ by Eq. (6) and Eq. (7).
7. Employ residual learning to the attention layer by calculating the contributions of local and non-local sources by Eq. (10).
8. Let $Z = Z \cup Z_i$.
9. End for
10. End for

calculates a scalar to reveal the relevance between signal intensities of $X_i$ and $X_j$ while the function $g$ computes the feature embedding of $X$ at position $j$. The contribution is determined by both the relevance and the signal intensity. The response is subsequently normalized by a factor $C(X_i)$, which is defined in Eq. (7). Then the output of the self-attention layer with residual learning can be expressed by Eq. (10), where the scale parameter $\alpha$ balances the contributions between local and non-local sources in the training process. Furthermore, we add the multi-head to the self-attention by concatenating outputs of self-attention for $N$ times. Its advantage is to enable the model to learn relevant information in different representation subspaces. The learning procedure of self-attention is shown in Algorithm 3.

Finally, output features of the self-attention layer are flattened and fed into two fully connected layers for inference. A dropout layer and a batch normalization layer are applied between two fully connected layers. Overall, the feature coding network has 23.03 million parameters and 0.35 billion floating point operations (FLOPs). Among them, the backbone network, the 5 NetVLAD layers, the self-attention layer, and the last two dense layers have 0.3169 billion FLOPs, 0.0093 billion FLOPs, 0.0007 billion FLOPs, and 0.0231 billion FLOPs, respectively. When using the NVIDIA TitanX GPU, average computation time for each batch during the training phase and the test phase is 34.20 ms and 11.41 ms, respectively.

### D. THE DESIGN OF LOSS FUNCTION

Softmax loss and center loss are selected to form a combined loss function in this work. Softmax loss and center loss are shown in Eq. (11) and Eq. (12), respectively, where $\eta$, $o$ and $c_{y_i}$ denote the training sample amount, the number of classes and the center of all samples in classes $y_i$ corresponding to $i^{th}$ sample, respectively. $\zeta$ in Eq. (13) is the overall loss function of our model which balances the softmax and center losses with a parameter $\lambda$. With $\lambda$ growing larger, the distance within the class becomes smaller.

$$\zeta_s = -\sum_{i=1}^{\eta} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{o} e^{W_{y_j}^T x_i + b_j}} \tag{11}$$

$$\zeta_c = \frac{1}{2}\sum_{i=1}^{\eta} \|x_i - c_{y_i}\|_2^2 \tag{12}$$

$$\zeta = \zeta_s + \lambda\zeta_c \tag{13}$$

Softmax loss function makes features spread in narrow strips. Although classes are separated from each other, feature distribution within a class is not compact. Owing to the large variance within a class, robustness of the model may be deteriorated if merely softmax loss is used. Center loss learns a center for features of each class and punishes on distances between features [42], thus making the distribution of features within a class more compact. As shown in Figure 3, features extracted by feature coding network with both softmax loss and center loss is more compact and yields better distinguishability than features extracted by that with softmax loss alone.

### E. FUSION STRATEGY

Before building the ensemble, the best combination of features is selected using a simple sum rule on those feature coding networks. Then, high-level features of the selected combination are aggregated and fed to a meta-CNN which consists of three blocks, a self-attention layer, and two fully connected layers. Each block in the meta-CNN consists of a convolution layer with 1D convolutions, a batch normalization layer, an ELU activation layer, and a pooling layer. The number of convolutional kernels of the convolution layer in the three blocks are 32, 64, and 32, respectively. Sequentially, a self-attention is used to learn the dependency among heterogeneous high-level features. Finally, output features of the self-attention layer are flattened and fed into two fully connected layers for final inference. A dropout layer and a batch normalization layer are applied between the two fully connected layers. Meta-CNN has 20.19 million parameters and 0.02 billion FLOPs. Average computation time of each

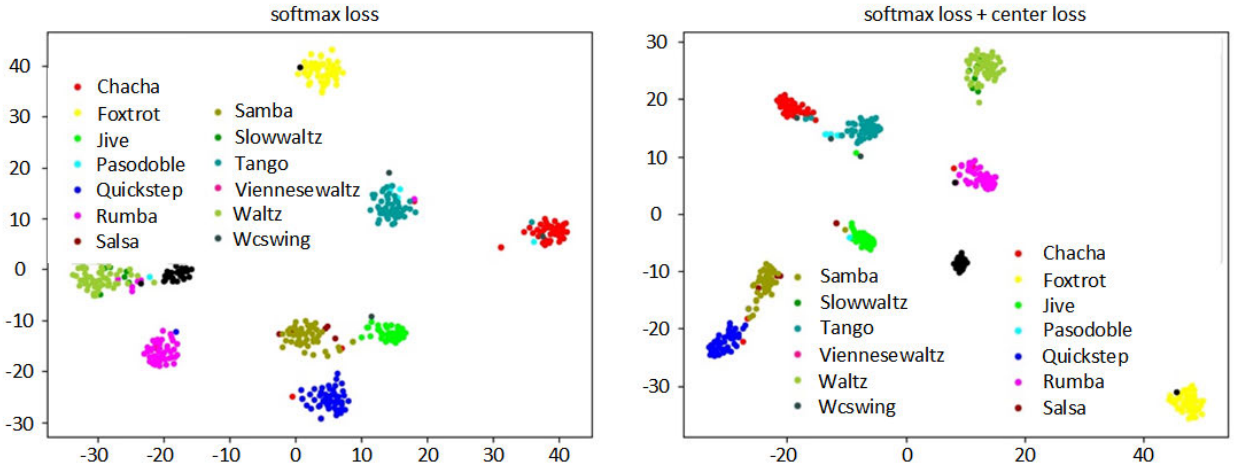


**FIGURE 3.** 2-D distribution visualization of the feature extracted by the feature coding network on the Extended Ballroom dataset.

batch during the training phase and test phase is 44.65ms and 14.33ms, respectively, using an NVIDIA TitanX GPU.

## IV. EXPERIMENTS

Details of three benchmark datasets and experimental settings are given in Sections IV-A and IV-B, respectively. Then, Section IV-C provides experimental results and discussions on these three datasets.

### A. DESCRIPTION OF DATASETS

Datasets used for evaluation of models are GTZAN, ISMIR2004, and Extended Ballroom. The GTZAN dataset consists of 10 genre classes: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock [20]. Each genre class consists of 100 audio recordings around 30 seconds with 1000 music excerpts in total. These excerpts are taken from radio, compact disks, and MP3 files. Each item is stored as a mono audio file of 22.050 kHz and 16-bit. ISMIR2004 is a genre classification dataset proposed for the music information retrieval contest organized by the Music Technology Group and Pompeu Fabra University. It consists of 1458 samples of six different genres: classical (640), electronic (229), jazz/blues (52), metal/punk (90), rock/pop (203), and world (245). The 1458 music pieces are divided into training set (50%) and test set (50%) in the contest. The Extended Ballroom dataset is an extension of the Ballroom dataset. The number of music excerpts in the Extended Ballroom dataset is 4180, which is six times larger than the Ballroom dataset [43]. The 4180 music excerpts include Chacha (455), Foxtrot (507), Jive (350), Pasodoble (53), Quickstep (497), Rumba(470), Salsa (47), Samba (468), Slowwaltz (65), Tango (464), Viennese Waltz (252), Waltz (529), and Wcswing (23).

For a fair comparison, we do not use any artist filter [2] before conducting MGR, following the baseline of several studies. In addition to *Accuracy* [1], a widely used evaluation criterion, *Precision*, *Recall*, and $F_1$ are also used. $CCP(\mathrm{Id})$, $TCP(\mathrm{Id})$, and $TNE(\mathrm{Id})$ denote the number of correctly classified positives for class Id, the total number of objects classified as class Id, and the total number of elements in class Id, respectively. $TP(\mathrm{Id})$, $TN(\mathrm{Id})$, $FP(\mathrm{Id})$, and $FN(\mathrm{Id})$ denote true positives, true negatives, false positives, and false negatives of class Id, respectively. Moreover, GTZAN contains many replications, distortions, and mislabeled samples [44]. Hence, we also conduct experiments with the artist filter to mitigate Sturm's flaws on the GTZAN dataset, which is shown in Section IV-C5.

$$Precision(Id) = \frac{CCP(Id)}{TCP(Id)} \times 100\% \tag{14}$$

$$Recall(Id) = \frac{CCP(Id)}{TNE(Id)} \times 100\% \tag{15}$$

$$Accuracy(Id) = 100\% \times \frac{TP(Id)+TN(Id)}{TP(Id)+TN(Id)+FP(Id)+FN(Id)} \tag{16}$$

$$F_1(Id) = 2 \times \frac{Precision(Id) \times Recall(Id)}{Precision(Id) + Recall(Id)} \times 100\% \tag{17}$$

### B. EXPERIMENTAL SETTINGS

For the GTZAN dataset, both 10-fold cross validation and a manual split by [45] are applied to evaluate the proposed method. Training set and test set are given for the ISMIR2004 dataset, so they are used directly. For the Extended Ballroom dataset, a 10-fold cross validation is also applied for evaluation. The RMSProp optimizer with a smoothing constant of 0.9 and the $L_2$ penalty of $4e$-5 is used for feature coding network training. The learning rate is set to be 0.001 which decays every two epochs with an exponential rate of 0.94. Before training the meta-CNN for final classification, extracted high-level features are normalized to have zero mean and unit standard deviation. The RMSProp optimizer with a fixed learning rate of $1e$-5 is used to optimize

the meta-CNN. Experiments are carried out on the deep learning platform PyTorch with an NVIDIA TitanX GPU.

### C. RESULTS AND DISCUSSIONS

Firstly, we present results of hyperparameters selection and ablation experiments of NetVLAD and self-attention. Then, tests on fusion strategies and visualization for both the training process and extracted features are reported. The artist filter is used to evaluate the GTZAN dataset to mitigate its flaws. Finally, the results of the proposed model are compared with other state-of-the-art models.

#### 1) HYPERPARAMETERS SELECTION

In the proposed model, there are four key hyperparameters: the dictionary size $K$, the number of heads $N$ in the self-attention mechanism, the $\lambda$ in the loss function $\zeta$ as shown in Eq. (13), and the choice of activation function. We test the cases where $K = 5, 10, 15, 20, 25, 30, 35, 40$ and $N = 5, 10, 15, 20, 25, 30,$ and $35$. We take the Mel-spectrogram of ISMIR2004 dataset as an example to select the appropriate $K$ and $N$ in the feature coding network.

The accuracies obtained for different $K$ and $N$ are shown in Figure 4 and Figure 5, respectively. There is an increment of the accuracy with $K$ increasing in Figure 4. However, accuracies do not change much when $K > 20$. A smaller $K$ implies faster training time, so $K = 20$ is selected. In Figure 5, $N = 10$ provides the highest accuracy. Hence, it is selected as the optimal number of $N$ for further analysis.

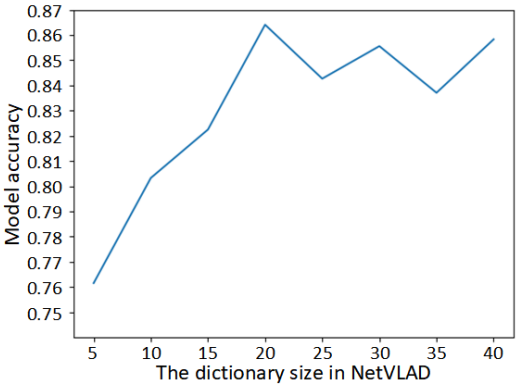


**FIGURE 4.** Variation of the accuracies for the feature coding network with the dictionary size in NetVLAD.

Accuracies obtained by the feature coding network based on different loss functions are shown in Table 1. Table 1 shows that the proposed network with a combination of loss functions (Eq. (13)) yields an improvement of 0.69% test accuracy compared with that merely employing softmax loss (i.e. $\zeta_s$). In addition, $\zeta$ with $\lambda = 0.001$ yields the best test accuracy. Center loss may be beneficial for enlarging distances among classes and enhancing the robustness of the model. Therefore, $\lambda = 0.001$ is used in our experiments.

Figure 6 shows the convergence of feature encoding networks using different activation functions. When using the ELU function, the feature coding network converges faster

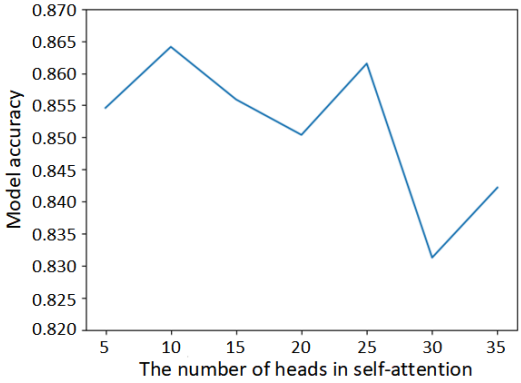


**FIGURE 5.** Variation of the accuracies for the feature coding network with the number of heads in self-attention.

**TABLE 1.** Accuracies of feature coding networks with different loss functions.

| Different loss functions | Accuracy |
|---|---|
| Feature coding network + $\zeta_s$ | 0.8573 |
| Feature coding network + $\zeta$ with $\lambda = 0.1$ | 0.8340 |
| Feature coding network + $\zeta$ with $\lambda = 0.01$ | 0.8587 |
| Feature coding network + $\zeta$ with $\lambda = 0.001$ | 0.8642 |
| Feature coding network + $\zeta$ with $\lambda = 0.0001$ | 0.8628 |

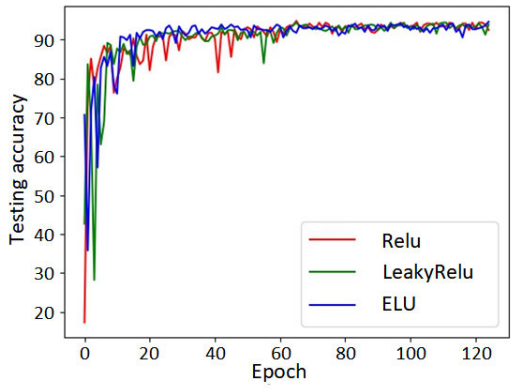


**FIGURE 6.** Comparison of test accuracies among different loss functions for feature coding networks.

with fewer fluctuations compared to those using Relu and LeakyRelu activation function.

#### 2) ABLATION TESTS

Figure 7 shows the results of ablation experiments on self-attention and NetVLAD using the Mel-spectrogram of ISMIR 2004 dataset as an example. It is shown that the model with both self-attention and NetVLAD yields the best performance. The model with neither self-attention nor NetVLAD yields the worst result. This result suggests that global features are not sufficient enough to improve accuracy compared with local features obtained by NetVLAD. Besides, the self-attention mechanism improves global features by learning their dependencies. However, Figure 7 shows that the attention mechanism does not improve the model with NetVLAD a lot. This can be explained by the fact that

**TABLE 2.** Comparison of accuracies for feature coding networks with different low-level features.

| Features | GTZAN | ISMIR2004 | Extended Ballroom |
|---|---|---|---|
| **timbre** | $0.7900 \pm 0.0293$ | 0.7846 | $0.8013 \pm 0.0062$ |
| **rhythm** | $0.7610 \pm 0.0262$ | 0.7201 | $0.7741 \pm 0.0083$ |
| **Mel-spectrogram** | $0.8910 \pm 0.0277$ | 0.8642 | $0.9254 \pm 0.0072$ |
| **constant-$Q$ spectrogram** | $0.8330 \pm 0.0287$ | 0.8422 | $0.9267 \pm 0.0085$ |
| **harmonic spectrogram** | $0.8340 \pm 0.0317$ | 0.8573 | $0.9129 \pm 0.0095$ |
| **percussive spectrogram** | $0.8370 \pm 0.0290$ | 0.7942 | $0.8555 \pm 0.0112$ |
| **scatter transform spectrogram** | $0.8950 \pm 0.0254$ | 0.8601 | $0.9380 \pm 0.0070$ |
| **transfer feature** | $0.8410 \pm 0.0192$ | 0.8738 | $0.9217 \pm 0.0088$ |

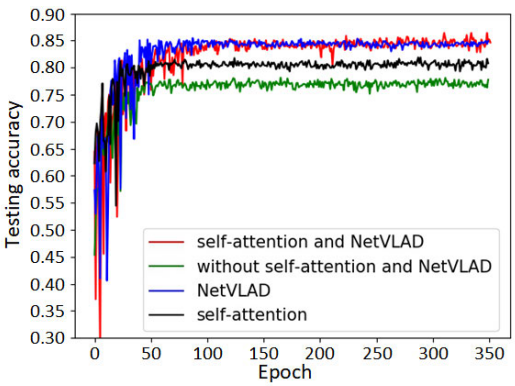


**FIGURE 7.** Effects of self-attention and NetVLAD with Mel-spectorgam as inputs.

local features across different levels provide discriminative information for MGR, which may fairly overlap with further information obtained from the self-attention mechanism. Nonetheless, self-attention learns dependencies of these local features across different levels, which helps to improve the performance of the model.

### 3) FEATURE COMBINATIONS

To study the performance of feature coding network for an individual feature, accuracies obtained by feature coding networks with different low-level features are shown in Table 2. No single low-level feature yields the best performance on these datasets. However, the rhythm feature yields the worst performance for all three datasets. Table 2 shows that the feature coding network using scatter transform spectrogram yields the best test accuracies of 89.50% for GTZAN and 93.80% for Extended Ballroom, respectively. For the ISMIR2004 dataset, the feature coding network using transfer feature yields the best test accuracy of 87.38%. These results can be expected because the scatter transform spectrogram is a local translation invariant representation being stable to time-warping deformation while the transfer feature was trained on a very large Million Song Dataset with rich label sets for various aspects of music including mood, genre, and instrumentation.

Then we evaluate performances yielded with different combinations of low-level features. In our method, an ensemble of feature coding networks is trained and each base network uses a type of low-level features as inputs. Given 247 combinations (combinations of 2 to 8 features) over 8 low-level features, the simple sum rule is applied to select the top 6 feature combinations yielding the highest test accuracies for each dataset. Tables 3, 4, and 5 show test accuracies on chosen datasets yielded by the networks with both a simple sum rule and a meta-CNN using the top 6 feature combinations. In these tables, $c$, $h$, $m$, $p$, $s$, $t$, and $T$ denote constant-$Q$ transform feature, harmonic spectrogram, Mel-spectrogram, percussive spectrogram, scatter spectrogram, timbre feature, and the transfer feature, respectively.

**TABLE 3.** Accuracies for different feature combination on ISMIR2004 dataset.

| Feature combination | Meta-CNN | Sum rule |
|---|---|---|
| $\{c, h, m, s, t, T\}$ | 0.9246 | 0.9218 |
| $\{c, h, m, p, s, T\}$ | 0.9149 | 0.9190 |
| $\{c, m, p, s, t, T\}$ | 0.9135 | 0.9190 |
| $\{c, h, m, p, s, t, T\}$ | 0.9135 | 0.9122 |
| $\{c, m, p, t, T\}$ | 0.9108 | 0.9122 |
| $\{c, h, t, T\}$ | 0.9090 | 0.9122 |

**TABLE 4.** 10-fold cross validation accuracies for different feature combinations on GTZAN dataset.

| Feature combination | Meta-CNN | Sum rule |
|---|---|---|
| $\{m, p, s, T\}$ | $0.9650 \pm 0.0081$ | $0.9420 \pm 0.0098$ |
| $\{c, h, s, t\}$ | $0.9360 \pm 0.0162$ | $0.9380 \pm 0.0181$ |
| $\{c, p, s, t\}$ | $0.9380 \pm 0.0125$ | $0.9310 \pm 0.0117$ |
| $\{m, s\}$ | $0.9580 \pm 0.0154$ | $0.9300 \pm 0.0184$ |
| $\{c, h, m, s\}$ | $0.9430 \pm 0.0168$ | $0.9300 \pm 0.0204$ |
| $\{h, m, s, t\}$ | $0.9420 \pm 0.0160$ | $0.9300 \pm 0.019$ |

**TABLE 5.** 10-fold cross validation accuracies for different feature combinations on Extended Ballroom dataset.

| Feature combination | Meta-CNN | Sum rule |
|---|---|---|
| $\{s, T\}$ | $0.9550 \pm 0.0123$ | $0.9454 \pm 0.0084$ |
| $\{m, s, T\}$ | $0.9486 \pm 0.0082$ | $0.9430 \pm 0.0054$ |
| $\{c, s, T\}$ | $0.9481 \pm 0.0125$ | $0.9428 \pm 0.0070$ |
| $\{m, c, s, T\}$ | $0.9414 \pm 0.0094$ | $0.9409 \pm 0.0078$ |
| $\{h, s, t, T\}$ | $0.9323 \pm 0.0068$ | $0.9397 \pm 0.0077$ |
| $\{c, p, s, T\}$ | $0.9342 \pm 0.0160$ | $0.9390 \pm 0.0080$ |

Obviously, the 6 best feature combinations for the three datasets are different. This shows that neither a single combination nor single feature provides the best results for all datasets. Experimental results also show that fusion by a

**TABLE 6.** Confusion matrix based on $\{c, h, m, s, t, T\}$ with effectiveness measures on ISMIR2004 dataset.

| | classical | electronic | jazz | metal/punk | rock/Pop | world | *Recall* |
|---|---|---|---|---|---|---|---|
| **classical** | 314 | 0 | 0 | 0 | 1 | 4 | 98.43% |
| **electronic** | 0 | 105 | 1 | 0 | 6 | 3 | 91.30% |
| **jazz** | 0 | 0 | 26 | 0 | 0 | 0 | 100% |
| **metal/punk** | 0 | 1 | 0 | 42 | 2 | 0 | 93.33% |
| **rock/pop** | 1 | 5 | 2 | 5 | 85 | 3 | 84.16% |
| **world** | 8 | 10 | 0 | 0 | 3 | 102 | 82.93% |
| *Precision* | 97.21% | 86.78% | 89.66% | 89.36% | 87.63% | 91.07% | |
| $F_1$ | 97.82% | 88.98% | 94.55% | 91.30% | 85.86% | 86.81% | |
| *Accuracy* | 91.95% | 91.70% | 92.08% | 92.20% | 92.91% | 93.87% | |

**TABLE 7.** Confusion matrix based on $\{m, p, s, T\}$ with effectiveness measures on GTZAN dataset.

| | blues | classical | country | disco | hiphop | jazz | metal | pop | reggae | rock | *Recall* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **blues** | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| **classical** | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00% |
| **country** | 0 | 0 | 98 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 98.00% |
| **disco** | 0 | 1 | 1 | 95 | 2 | 0 | 0 | 1 | 0 | 0 | 95.00% |
| **hiphop** | 0 | 0 | 0 | 1 | 95 | 0 | 1 | 1 | 1 | 1 | 95.00% |
| **jazz** | 0 | 1 | 0 | 0 | 0 | 99 | 0 | 0 | 0 | 0 | 99.00% |
| **metal** | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 | 0 | 1 | 99.00% |
| **pop** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 98 | 0 | 1 | 98.00% |
| **reggae** | 0 | 1 | 2 | 0 | 3 | 2 | 0 | 1 | 90 | 1 | 90.00% |
| **rock** | 2 | 0 | 2 | 0 | 0 | 1 | 2 | 0 | 2 | 91 | 91.00% |
| *Precision* | 98.04% | 97.09% | 94.23% | 98.96% | 95.00% | 97.06% | 97.06% | 96.08% | 96.78% | 94.79% | |
| $F_1$ | 99.01% | 98.52% | 96.08% | 96.94% | 95.00% | 98.02% | 98.02% | 97.03% | 93.26% | 92.86% | |
| *Accuracy* | 96.31% | 96.21% | 96.12% | 96.89% | 96.50% | 96.31% | 96.31% | 96.31% | 97.18% | 96.89% | |

**TABLE 8.** Confusion matrix based on $\{s, T\}$ with effectiveness measures on Extended Ballroom dataset.

| | Cha | Fox | Jiv | Pas | Qui | Rum | Sal | Sam | Slo | Tan | Vie | Wal | Wcs | *Recall* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cha** | 443 | 0 | 0 | 0 | 1 | 10 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0.9736 |
| **Fox** | 0 | 502 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0.9901 |
| **Jiv** | 0 | 5 | 342 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0.9771 |
| **Pas** | 7 | 1 | 1 | 37 | 1 | 2 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0.6981 |
| **Qui** | 0 | 3 | 1 | 0 | 487 | 0 | 0 | 3 | 1 | 0 | 1 | 1 | 0 | 0.9799 |
| **Rum** | 0 | 0 | 1 | 0 | 0 | 460 | 0 | 3 | 3 | 1 | 0 | 2 | 0 | 0.9787 |
| **Sal** | 0 | 1 | 2 | 0 | 0 | 3 | 21 | 18 | 0 | 1 | 1 | 0 | 0 | 0.4468 |
| **Sam** | 1 | 0 | 2 | 0 | 1 | 1 | 4 | 459 | 0 | 0 | 0 | 0 | 0 | 0.9808 |
| **Slo** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 1 | 44 | 0 | 0.3077 |
| **Tan** | 3 | 0 | 0 | 0 | 2 | 3 | 0 | 0 | 2 | 454 | 0 | 0 | 0 | 0.9784 |
| **Vie** | 0 | 3 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 239 | 8 | 0 | 0.9484 |
| **Wal** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 3 | 524 | 0 | 0.9905 |
| **Wcs** | 3 | 3 | 4 | 0 | 3 | 1 | 0 | 1 | 0 | 4 | 0 | 0 | 4 | 0.1739 |
| *Precision* | 0.9694 | 0.9691 | 0.968 | 1 | 0.9819 | 0.9465 | 0.8400 | 0.9464 | 0.7407 | 0.9763 | 0.9755 | 0.9003 | 1 | |
| $F_1$ | 0.9715 | 0.9795 | 0.9730 | 0.8222 | 0.9809 | 0.9623 | 0.5833 | 0.9633 | 0.4348 | 0.9774 | 0.9618 | 0.9433 | 0.2963 | |
| *Accuracy* | 0.9546 | 0.9525 | 0.9543 | 0.9587 | 0.9553 | 0.9514 | 0.9601 | 0.9512 | 0.9638 | 0.9548 | 0.9566 | 0.9431 | 0.9594 | |

meta-CNN is more competitive than the fusion by a sum rule in some scenarios. The best results on all three datasets are yielded by fusion using a meta-CNN, which verifies that learning the internal relationship among different features at early stages improves the results. $m$, $s$, and $T$ are useful low-level features for the best performing meta-CNNs. The low-level feature combination yielding the best test accuracy is used for each dataset in our model. Therefore, the proposed method uses $\{c, h, m, s, t, T\}$ for ISMIR2004, $\{m, p, s, T\}$ for GTZAN and $\{s, T\}$ for Extended Ballroom for experiments in the next sub-section, respectively.

To more accurately interpret results obtained for MGR, Confusion Matrix, *Precision*, *Recall*, $F_1$, and *Accuracy* for each class on chosen datasets are given in Tables 6, 7, and 8, respectively. The first row and the first column of a confusion matrix correspond to the predicted genre and target genre, respectively. For the ISMIR2004 dataset, ''electronic'' is the most confused while ''world'' is the most misclassified class. ''Classical'' achieves the highest $F_1$ score of 97.82% while the differences in classification accuracy for each genre are insignificant. In Table 7, samples of ''blues'' and ''classical'' are all predicted correctly on the GTZAN dataset. ''Blues'' achieves the highest F1 score of 99.01% while ''rock'' achieves the worst $F_1$ score of 92.86%. Samples of ''country'' and ''rock'' are the most confused. Especially, ''rock'', ''reggae'', and ''disco'' have the largest numbers of misclassified samples while ''blues'', ''classical'', and ''metal'' have the largest numbers of correctly classified samples. When the

artist filter is not used, the distribution of classification results of our method is consistent with experimental results in [44]. In Table 8, Cha, Fox, Jiv, Pas, Qui, Rum, Sal, Sam, Slo, Tan, Vie, Wal, and Wcs denote "Chacha", "Foxtrot", "Jive", "Pasodoble", "Quickstep", "Rumba", "Salsa", "Samba", "Slowwaltz", "Tango", "Viennesewaltz", "Waltz", and "Wcswing", respectively. "Pasodoble" and "Wcswing" are the least confused. The "Waltz" has the largest number of correctly classified samples with a moderate degree of confusion. "Wcswing" has the largest number of misclassified samples, but is also the least confused. This is because "Wcswing" has only 23 samples, which is ten times fewer than the other music genres.

#### 4) VISUALIZATION

Visualizations on the training process and extracted features are provided to prove that the proposed model is not overfitting on small datasets. T-SNE (t-distributed stochastic neighbor embedding) is used to visualize the best feature combination before feeding it into the meta-CNN. Figures 8, 9, and 10 show the 2-D visualizations of the best feature combination extracted by feature coding network for ISMIR2004, GTZAN, and Extended Ballroom datasets, respectively.

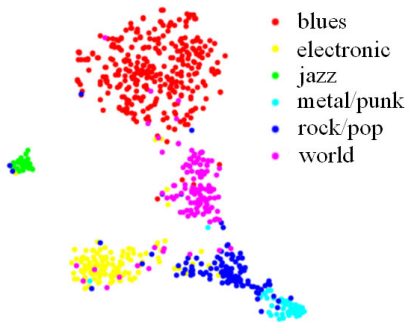


**FIGURE 8.** 2-D distribution visualization of the best feature combination extracted by the feature coding network on ISMIR2004 dataset.

In Figure 8, the distribution of "jazz" is the most compact on the ISMIR2004 dataset, followed by "metal/punk"..The genre of "world" is the least compact. As shown in Figure 9, distributions of "blues" and "classical" are the most compact for the GTZAN dataset, followed by "jazz" and "metal". "Rock" is the most loosely distributed. As shown in Figure 10, distributions of "Wcswing", "Pasodoble", and "Quickstep" are the most compact for the Extended Ballroom dataset while distributions of "Salsa" and "Slowwaltz" are the loosest. Overall, the overlapping between classes is insignificant for all three datasets. As shown in Figures 8, 9, and 10, the distribution of each genre is concentrated at a center instead of multiple centers. Boundaries of most genres are clear and can be divided using a smooth hypersurface. These show that our deep model does not overfit on these small datasets. Furthermore, we feed the best feature combination to the meta classifier and visualize the training process of the model on these three datasets in Figure 11. Figure 11 shows that the proposed model does not overfit and yields good test accuracies (i.e. good generalization capability).

#### 5) GTZAN WITH ARTIST FILTER

Following the evaluation of the GTZAN dataset in [2], we conduct experiments with 3-fold cross-validation and the artist-filtered split as in [45]. In these experiments, all duplicate songs and unrecognizably distorted songs are removed. An artist filter (AF) is applied to make sure that no song from the same artist appears in both training and the test set of a fold. As shown in Table 4, Mel-spectrogram, percussive spectrogram, scatter spectrogram, and transfer feature form the best feature combination for GTZAN without AF, so they are used to evaluate the performance of GTZAN with AF. Accuracies of these features and their combination through the meta-CNN fusion are shown in Table 9. Table 10 shows the confusion matrix for the GTZAN with AF using the feature combination of $\{m, p, s, T\}$. The proposed method yields a worse test accuracy for GTZAN with AF (shown in Table 9) in comparison to that of without AF (shown in Table 2). Although the test accuracies drop for all the features when AF is used, the test accuracy of the transfer feature yields the smallest drop because it transfers knowledge learned from a large dataset. Our results with AF in Table 10 are highly similar to that in [44], where AF is proposed to evaluate GTZAN. For example, "rock" and "disco" are the most confused while "classical" and "metal" have the most correctly classified samples. The fusion accuracy with AF in Table 10 drops to 86.92% but it is still 31.10% higher than the accuracy with AF in [44].

**TABLE 9.** Accuracies for different feature and their fusion on GTZAN with AF.

| Features | Accuracy |
|---|---|
| **Mel-spectrogram** | $0.7004 \pm 0.0030$ |
| **scatter transform spectrogram** | $0.7046 \pm 0.0146$ |
| **percussive spectrogram** | $0.6708 \pm 0.0423$ |
| **transfer feature** | $0.7703 \pm 0.0171$ |
| **fusion** | $0.8692 \pm 0.0030$ |

#### 6) COMPARISONS WITH STATE-OF-THE-ART METHODS

Table 11 shows that test accuracies yielded by the proposed model outperform other state-of-the-art models [28], [45] on ISMIR2004, GTZAN, and Extended Ballroom datasets. For a fair comparison, all methods in Table 11 do not use AF. Our proposed model is called the FusionNet. For the ISMIR2004 dataset, FusionNet yields the test accuracy of 92.46%, which is 0.36% and 1.56% higher than the second-best and the third-best model in comparison, respectively. For the GTZAN dataset, FusionNet yields the test accuracy of 96.50%, which is 0.8% and 5.9% higher than the second-best and the third-best model in comparison, respectively. For the Extended Ballroom dataset, the test accuracy of FusionNet is 95.50%, which is 0.60% and 2.80%

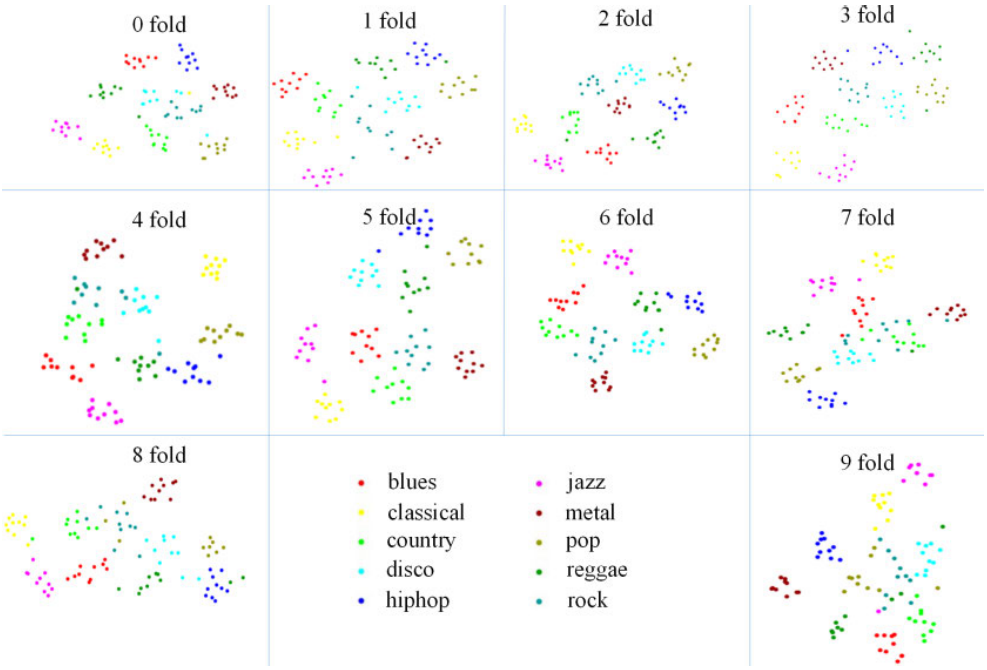


**FIGURE 9.** 2-D distribution visualization of the best feature combination extracted by the feature coding network on GTZAN dataset.

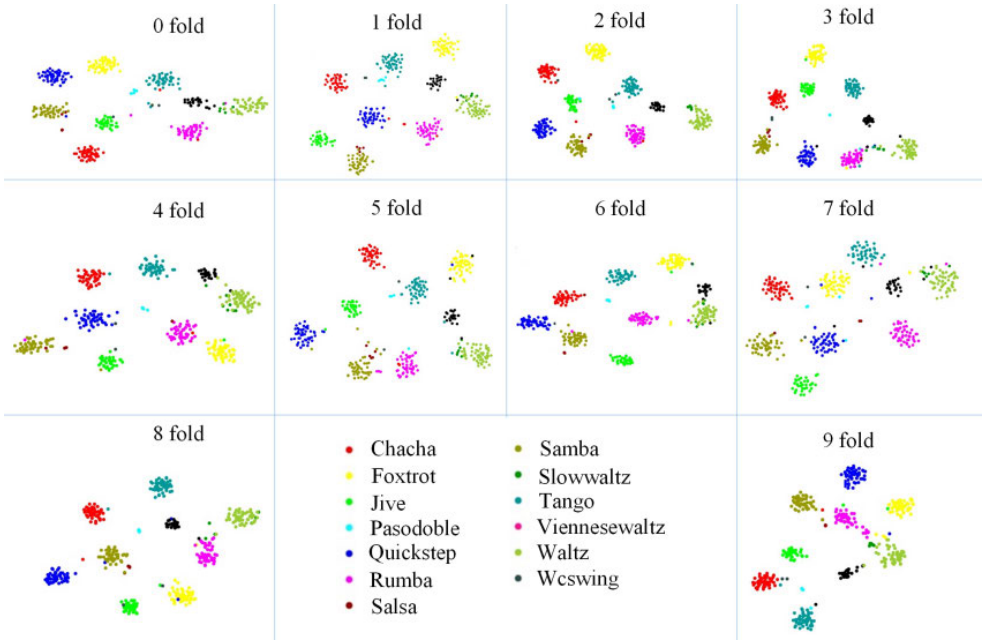


**FIGURE 10.** 2-D distribution visualization of the best feature combination extracted by the feature coding network on Extended Ballroom dataset.

higher than the second-best and the third-best model, respectively. Although the improvement in test accuracy is minor in comparison to the second-best method for the ISMIR2004 dataset, the proposed method yields satisfactory or significant improvements in other cases.

For methods based on improved handcrafted features for MGR tasks, a novel feature set derived from long-term modulation spectral analysis of spectral and cepstral features is proposed to characterize the temporal evolution of an audio signal [46]. Then, an information fusion approach that integrates both the feature-level fusion and decision-level fusion is employed to improve the classification accuracy. A classifier combined with the joint sparse low-rank representation is proposed to identify subspaces where the samples are projected [48]. A music genre classification model is proposed to capture the temporal evolution of the spectral characteristics of the music signal and reduce the computational complexity [49]. It contains spectro-temporal features

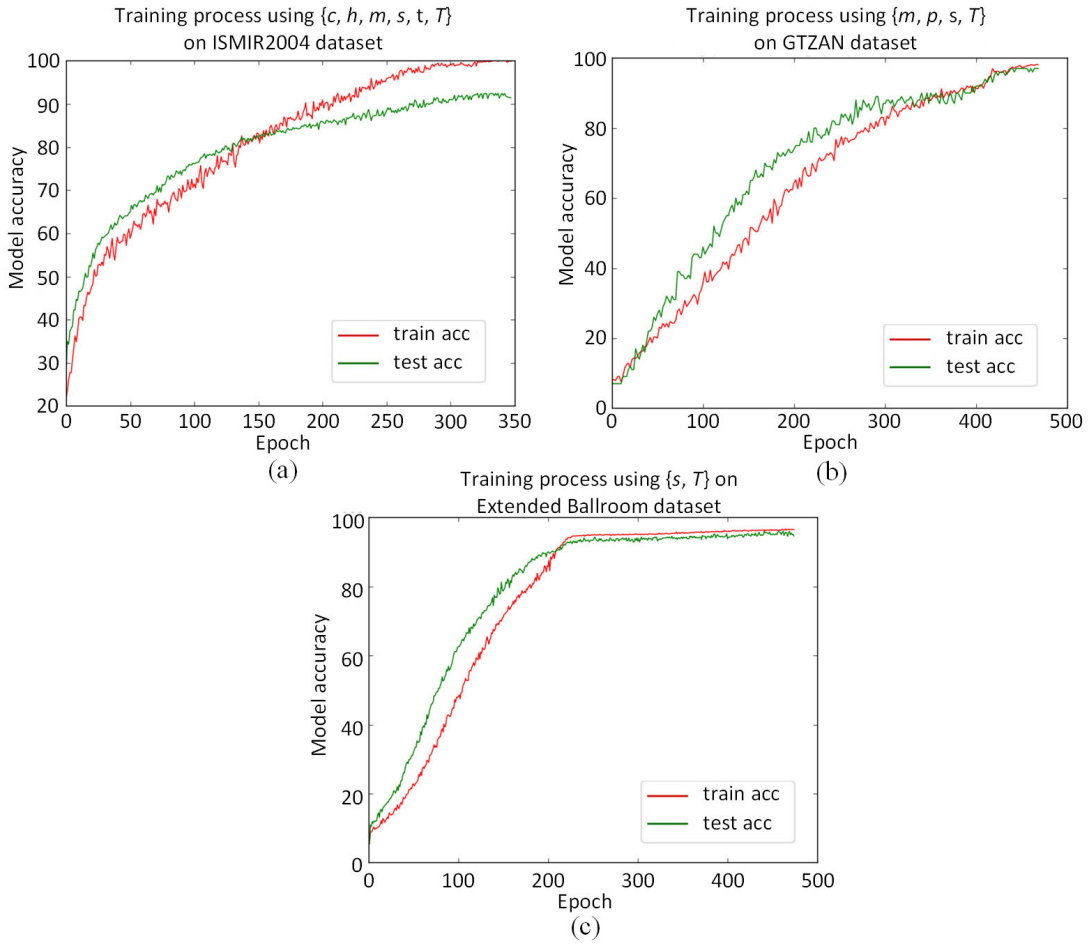


**FIGURE 11.** Training process using the best feature combination.

**TABLE 10.** Confusion matrix based on $\{m, p, s, T\}$ along with the effectiveness measures on GTZAN dataset with AF.

| | **blues** | **classical** | **country** | **disco** | **hiphop** | **jazz** | **metal** | **pop** | **reggae** | **rock** | *Recall* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **blues** | 82 | 0 | 4 | 5 | 0 | 5 | 0 | 0 | 1 | 3 | 82.00% |
| **classical** | 0 | 98 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 98.00% |
| **country** | 1 | 2 | 78 | 2 | 0 | 3 | 0 | 2 | 0 | 12 | 78.00% |
| **disco** | 0 | 1 | 1 | 83 | 2 | 0 | 1 | 4 | 2 | 0 | 84.96% |
| **hiphop** | 0 | 0 | 0 | 1 | 90 | 0 | 2 | 2 | 3 | 0 | 91.84% |
| **jazz** | 2 | 3 | 0 | 0 | 0 | 82 | 0 | 0 | 0 | 0 | 94.25% |
| **metal** | 0 | 0 | 1 | 0 | 1 | 0 | 88 | 0 | 0 | 1 | 96.70% |
| **pop** | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 82 | 0 | 3 | 91.11% |
| **reggae** | 0 | 0 | 3 | 4 | 4 | 3 | 0 | 0 | 73 | 1 | 82.95% |
| **rock** | 2 | 0 | 9 | 5 | 1 | 2 | 6 | 5 | 2 | 68 | 68.00% |
| *Precision* | 94.25% | 94.23% | 78.79% | 80.58% | 91.84% | 85.42% | 90.72% | 86.32% | 90.12% | 77.27% | |
| $F_1$ | 87.70% | 96.08% | 78.39% | 82.58% | 91.84% | 89.62% | 93.61% | 88.65% | 86.39% | 72.34% | |
| *Accuracy* | 88.09% | 86.51% | 86.97% | 86.06% | 86.88% | 86.06% | 86.33% | 86.42% | 87.53% | 88.00% | |

based on timbre features, a SVM ranker for feature selection, and a RBF kernel estimation for SVM classification. Two novel scale and shift-invariant time-frequency representations of the audio content are proposed in [47] to model the inter-relationship between the various frequency bands.

For methods based on representation learning, a bidirectional recurrent neural network with serial attention and parallelized attention is proposed to focus on details of the target area [15]. A CNN-RNN-cascaded deep learning model that uses almost no handcrafted features is proposed in [14]. In [12], a CNN-based architecture with multi-level and multi-scale features [11] is extended by transfer learning. A transfer feature is proposed in [6], and the deep network trained on the Million Song Dataset is used as a feature extractor for small datasets.

For methods based on ensemble learning, a combination of weighted classifiers are used to enhance the performance obtained by the fusion of sum rule [8]. In its follow-up work,

**TABLE 11. Comparative results of the proposed method and other state-of-the-art models on GTZAN, ISMIR2004, and Extended Ballroom dataset.**

| Method | ISMIR2004 | GTZAN | Extended Ballroom |
|---|---|---|---|
| **FusionNet** | **92.46%** | **96.50%** | **95.50%** |
| Nanni et al., 2018 [28] | 92.10% | 95.70% | - |
| Nanni et al., 2017 [9] | 90.90% | 90.60% | - |
| Nanni et al., 2016 [8] | 90.20% | 89.90% | - |
| Lee et al., 2009 [46] | 86.83% | 90.60% | - |
| Marchand et al., 2016 [47] | - | - | 94.90% |
| Yu et al., 2019 [15] | - | 90.00% | 92.70% |
| Choi et al., 2017 [6] | - | 89.80% | 86.70% |
| Panagakis et al., 2014 [48] | 85.45% | 89.40% | - |
| Lim et al., 2012 [49] | 89.90% | 87.40% | - |
| Costa et al., 2017 [29] | 87.10% | - | - |
| Bisharad et al., 2019 [14] | - | 85.36% | - |
| Lee et al., 2018 [12] | - | 82.10% | - |
| Lee et al., 2017 [11] | - | 72.00% | - |

the authors employ and evaluate more novel representations and texture descriptors [9]. The authors conduct tests on different texture descriptors and a model based on CNN in the extended version of this work [28]. The complementarity between handcrafted features and CNN features is investigated for the first time on music classification tasks [29].

Before our work, the best performances achieved on the ISMIR2004 and the GTZAN datasets are reported in [28], which fuses the results from the ensemble of handcrafted texture descriptors and a CNN-based model. The approach proposed in [28] achieves the test accuracies of 92.10% and 95.70% on the ISMIR2004 and GTZAN datasets, respectively. The highest testing accuracy of 94.90% on the Extended Ballroom dataset is reported in [47]. This method is based on the scale and shift-invariant time-frequency representations. Our proposed approach FusionNet yields test accuracies of 92.46%, 96.50%, and 95.50% on the ISMIR2004, GTZAN, and Extended Ballroom datasets, respectively. There are several reasons for worse performances yielded by other state-of-the-art methods. Methods based on improved handcrafted features merely use a single feature and thus fail to provide enough discriminative information. Although methods based on representation learning extract salient features directly from the audio signals, they are designed based on global features, thus failing to capture more valuable local features across different levels.

## V. CONCLUSION

In this work, we propose an ensemble approach for music genre recognition based on the fusion of high-level feature sets learned from different types of low-level features. A multi-level feature coding network uses a CNN with self-attention and NetVLAD to learn high-level features for each low-level feature. The NetVLAD extracts more dominant features by capturing local information from different feature levels while the self-attention learns long-term dependencies across levels. The proposed model is effective in capturing discriminative features, thus yielding the best test accuracy on GTZAN, ISMIR2004, and Extended Ballroom dataset.

In future, we intend to train the network in a multi-task learning manner by optimizing the local CNNs and global aggregated networks simultaneously to provide better performance. Furthermore, we will investigate different filter visualization techniques to interpret the filters and apply the proposed method on other tasks such as audio event classification, emotion prediction and music tagging.

## REFERENCES

[1] D. C. Corràa and F. A. Rodrigues, "A survey on symbolic data-based music genre classification," *Expert Syst. Appl.*, vol. 60, pp. 190–210, Oct. 2016.
[2] B. L. Sturm, "Classification accuracy is not enough: On the evaluation of music genre recognition systems," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 371–406, Dec. 2013.
[3] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat, "Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3d electronic densities," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Los Angeles, NV, USA, 2017, pp. 6540–6549.
[4] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky, "Scattering networks for hybrid representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2208–2221, Sep. 2019.
[5] N. Zeghidour, G. Synnaeve, M. Versteegh, and E. Dupoux, "A deep scattering spectrum—Deep siamese network pipeline for unsupervised acoustic modeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Oct. 2016, pp. 4965–4969.
[6] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, SuZhou, China, 2017, pp. 141–149.
[7] Y.-F. Huang, S.-M. Lin, H.-Y. Wu, and Y.-S. Li, "Music genre classification based on local feature selection using a self-adaptive harmony search algorithm," *Data Knowl. Eng.*, vol. 92, pp. 60–76, Jul. 2014.
[8] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, "Combining visual and acoustic features for music genre classification," *Expert Syst. Appl.*, vol. 45, pp. 108–117, Mar. 2016.
[9] L. Nanni, Y. M. G. Costa, D. R. Lucio, C. N. Silla, and S. Brahnam, "Combining visual and acoustic features for audio classification tasks," *Pattern Recognit. Lett.*, vol. 88, pp. 49–56, Mar. 2017.
[10] K. Markov and T. Matsui, "Music genre and emotion recognition using Gaussian processes," *IEEE Access*, vol. 2, pp. 688–697, 2014.
[11] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1208–1212, Aug. 2017.
[12] J. Lee, J. Park, K. Kim, and J. Nam, "SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification," *Appl. Sci.*, vol. 8, no. 1, p. 150, Jan. 2018.
[13] S. Vishnupriya and K. Meenakshi, "Automatic music genre classification using convolution neural network," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Coimbatore, India, Jan. 2018, pp. 1–4.
[14] D. Bisharad and R. H. Laskar, "Music genre recognition using convolutional recurrent neural network architecture," *Expert Syst.*, vol. 36, no. 4, p. e12429, Aug. 2019.
[15] Y. Yu, S. Luo, S. Liu, H. Qiao, Y. Liu, and L. Feng, "Deep attention based music genre classification," *Neurocomputing*, vol. 372, pp. 84–91, Jan. 2020.
[16] R. M. Pereira, Y. M. G. Costa, R. L. Aguiar, A. S. Britto, L. E. S. Oliveira, and C. N. Silla, "Representation learning vs. Handcrafted features for music genre classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary Jul. 2019, pp. 1–8.
[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.
[18] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA Jun. 2010, pp. 3304–3311.
[19] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Los Angeles, NV, USA, 2017, pp. 5998–6008.
[20] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.

[21] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, London, U.K., 2005, pp. 34–41.
[22] M.-J. Wu and J.-S.-R. Jang, "Combining acoustic and multilevel visual features for music genre classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 1, pp. 1–17, Aug. 2015.
[23] H. Meng, T. Yan, F. Yuan, and H. Wei, "Speech emotion recognition from 3D log-mel spectrograms with deep learning network," *IEEE Access*, vol. 7, pp. 125868–125881, 2019.
[24] J. Driedger, M. Müller, and S. Disch, "Extending harmonic-percussive separation of audio," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, TaiBei, Taiwan, 2014, pp. 611–616.
[25] N. Holighaus, M. Dorfler, G. A. Velasco, and T. Grill, "A framework for invertible, real-time Constant-Q transforms," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 4, pp. 775–785, Apr. 2013.
[26] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time–frequency representations for audio scene classification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 142–153, Jan. 2015.
[27] J. Anden and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, Aug. 2014.
[28] L. Nanni, Y. M. G. Costa, R. L. Aguiar, C. N. Silla, and S. Brahnam, "Ensemble of deep learning, visual and acoustic features for music genre classification," *J. New Music Res.*, vol. 47, no. 4, pp. 383–397, Aug. 2018.
[29] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Appl. Soft Comput.*, vol. 52, pp. 28–38, Mar. 2017.
[30] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Miami, FL, USA, 2011, pp. 591–596.
[31] J. Ludeña-Choez, R. Quispe-Soncco, and A. Gallardo-Antolín, "Bird sound spectrogram decomposition through non-negative matrix factorization for the acoustic classification of bird species," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0179403.
[32] E. Benetos and C. Kotropoulos, "Non-negative tensor factorization applied to music genre classification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 8, pp. 1955–1967, Nov. 2010.
[33] K. Markov and T. Matsui, "High level feature extraction for the self-taught learning algorithm," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, pp. 1–11, Dec. 2013.
[34] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, USA, 2006, pp. 2169–2178.
[35] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2486–2493.
[36] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1794–1801.
[37] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-rank sparse coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 281–288.
[38] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, Dec. 2013.
[39] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Porto, Portugal, May 2019, pp. 139–144.
[40] V. D. Oord, A. Aäron, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pretraining for audio-based music classification," in *Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Taipei, Taiwan, 2014, pp. 29–34.
[41] H. Kazemi, M. Iranmanesh, and N. M. Nasrabadi, "Automatic target recognition using deep convolutional neural networks," in *Proc. Autom. Target Recognit.*, New York, NY, USA, Apr. 2018, pp. 805–811.
[42] Y. Wen, K. Zhang, and Z. Li, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 499–515.
[43] U. Marchand and G. Peeters, "The extended ballroom dataset," in *Proc. 17th Int. Soc. for Music Inf. Retr. Conf.*, New York, NY, USA, USA, 2016, pp. 1–8.
[44] B. L. Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," 2013, *arXiv:1306.1461*. [Online]. Available: http://arxiv.org/abs/1306.1461
[45] J. H. Foleiss and T. F. Tavares, "Texture selection for automatic music genre classification," *Appl. Soft Comput.*, vol. 89, pp. 106–127, Oct. 2020.
[46] C.-H. Lee, J.-L. Shih, K.-M. Yu, and H.-S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, Jun. 2009.
[47] U. Marchand and G. Peeters, "Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Vietri sul Mare, Italy Sep. 2016, pp. 1–6.
[48] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce, "Music genre classification via joint sparse low-rank representation of audio features," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1905–1917, Dec. 2014.
[49] S.-C. Lim, J.-S. Lee, S.-J. Jang, S.-P. Lee, and M. 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.

**WING W. Y. NG** (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Hong Kong Polytechnic University, Hong Kong, in 2001 and 2006, respectively.

He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is currently the Deputy Director of the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information. His current research interests include neural networks, deep learning, smart grid, smart healthcare, smart manufacturing, and nonstationary information retrieval.

Dr. Ng is currently an Associate Editor of *International Journal of Machine Learning and Cybernetics*. He is the Principle Investigator of four China National Natural Science Foundation projects and a Program for New Century Excellent Talents in University from China Ministry of Education. He served as the Board of Governor of the IEEE Systems, Man and Cybernetics Society, in 2011 and 2013.

**WEIJIE ZENG** received the B.Sc. degree in electronics information science and technology from the South China University of Technology, Guangzhou, China, in 2017, where he is currently pursuing the M.Sc. degree with the School of Computer Science and Engineering. His current research interests include deep learning and its applications in video and audio signals.

**TING WANG** (Student Member, IEEE) received the M.Sc. degree in computer science from Northeast Normal University, Changchun, China, in 2017. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

Her current research interests include learning methods and generalization capabilities for deep neural networks, and their applications in real-world problems, such as smart healthcare and smart grid.