IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Parallel Recurrent Convolutional Neural Networks Based Music Genre Classification Method for Mobile Devices

RUI YANG[1,2], LIN FENG[1,3], HUIBING WANG[4], JIANING YAO[3], and SEN LUO[3].

[1]Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning 116023 China
[2]Software College, Shenyang Normal University, Shenyang, Liaoning 110034 China
[3]School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian, Liaoning 116023 China
[4]College of Information Science and Technology, Dalian Maritime University, Dalian, Liaoning 116026 China

Corresponding author: Lin Feng (e-mail: Fenglin@dlut.edu.cn).

This study was funded by LiaoNing Revitalization Talents Program No. XLYC1806006, and National Natural Science Foundation of People's Republic of China No. 61672130, No. 61972064. All the authors declare that they have no conflict of interest.

ABSTRACT With the rapid development of the mobile internet of things (IoTs) and mobile sensing devices, a large amount of mobile computing-oriented applications have attracted attention both from industry and academia. Deep learning based methods have achieved great success in artificial intelligence (AI) oriented applications. To advance the development of AI-based IoT systems, effective and efficient algorithms are in urgent need for IoT Edge Computing. Time-series data classification is an ongoing problem in applications for mobile devices (e.g. music genre classification on mobile phones). However, the traditional methods require field expertise to extract handcrafted features from the time-series data. Deep learning has been demonstrated to be effective and efficient in this kind of data. Nevertheless, the existing works neglect some of the sequential relationships found in the time-series data, which are significant for time-series data classification. Considering the aforementioned limitations, we propose a hybrid architecture, named the parallel recurrent convolutional neural network (PRCNN). The PRCNN is an end-to-end training network that combines feature extraction and time-series data classification in one stage. The parallel CNN and Bi-RNN blocks focus on extracting the spatial features and temporal frame orders, respectively, and the outputs of two blocks are fused into one powerful representation of the time-series data. Then, the syncretic vector is fed into the softmax function for classification. The parallel network structure guarantees that the extracted features are robust enough to represent the time-series data. Moreover, the experimental results demonstrate that our proposed architecture outperforms the previous approaches applied to the same datasets. We also take the music data as an example to conduct contrastive experiments to verify that our additional parallel Bi-RNN block can improve the performance of time-series classification compared with utilizing CNNs alone.

INDEX TERMS Convolutional Neural Networks, Parallel, Time-series Data Classification.

I. INTRODUCTION

WITH the extensive utilization of various mobile devices, mobile computing has attracted attention both from industry and academia [1], [2]. An increasing amount of music is spreading widely on mobile devices, which is difficult for users and platforms to organize [3], [4]. Furthermore, it is impossible to organize and distinguish such a large amount of music manually. Therefore, constructing a convenient way to address this problem is challenging but of vital importance. Most of the state-of-the-art methods aim to classify the music genre, which is a top-level music label, to help users categorize and describe various types of music [5]. Meanwhile, the exact classification of the music genre is crucial to enable the music platforms to organize music into different groups. For this reason, the classification of the music genre has attracted wide attention in the field of music information retrieval (MIR) [6], [7].

Two crucial components for music genre classification, feature extraction and classifier learning, may greatly influence the performance of most classification systems [8]–

[12]. Feature extraction concentrates on exploring the suitable representations of the samples that will be classified using feature vectors or pairwise similarity measures [13]–[20]. After feature extraction, the features and representations of the music are fed into a classifier, which aims to map the feature vectors into different music genres. Baniya et al. [21] adopt the timbral texture features (i.e., the mel-frequency cepstral coefficient) and rhythm content features such as the beat histogram (BH) [5] to represent the music samples. Then, they combine an extreme learning machine (ELM) [22]– [26] and bagging classification [27]. Arabi et al. [28] use the statistical chord features and chord progression information in conjunction with the low-level features [13]. In addition, by utilizing a support vector machine (SVM), they prove that the chord features in conjunction with the low-level features can produce a higher classification accuracy. The state-of-the-art method is reported by Sarkar et al. [29], which employs empirical mode decomposition (EMD) for signal component extraction and depends only on the pitch-based features. Even though all the methods above achieve a good performance in certain situations, the handcrafted features still have some fatal disadvantages. Extracting handcrafted features from music samples is a complex process, so it requires researchers with expertise in the musical domain. Furthermore, the features that are extracted for a certain task lack universality, and they may perform poorly in other tasks. In recent years, deep learning networks, especially convolutional neural networks (CNNs), has been successfully utilized in various image classification tasks [30]– [35]. Meanwhile, Sander et al. [36] proved that music spectrograms, which are similar to normal images, can also achieve good performance with CNNs. Under this circumstance, there is a growing tendency to learn robust feature representations from the music spectrograms by using CNNs [37], [38]. In contrast, in the traditional methods, the CNNs provide an end-to-end training architecture that combines feature extraction with music classification in one stage. Multiple works based on CNNs have shown their superiority in music genre classification.

Notably, unlike ordinary images, music spectrograms have heavily sequential relationships. However, the existing music genre classification methods using CNNs are not able to model the long-term temporal information in the music spectrograms. Moreover, the model structure should consider the available hardware computing capability [39], [40]. Recurrent neural networks [41] (RNNs) can model long-term dependencies, such as in the music structure or recurrent harmonies [42], which are significant for music classification. To address the limitations mentioned above, we propose a hybrid learning architecture named the parallel recurrent and convolutional neural network (PRCNN), which consists of a CNN block and a parallel bidirectional recurrent neural network (Bi-RNN) block [43].

The main contributions of this paper are as follows:

- Utilize the short-term Fourier transform (STFT) to transform the time-domain information in the music samples to frequency-domain information, which facilitates a visual analysis of the music.
- Propose a hybrid model structure to combine the spatial features and temporal frame orders of the music samples, which consists of a CNN block and a parallel bidirectional recurrent neural network (Bi-RNN) block. Based on the structure, the frequency-domain information is considered as images through the CNN-based deep neural networks and is more suitable for music genre classification than simple CNNs.

The remainder of this paper is organized as follows. In Section 2, we analyze the previous related works of music genre classification and carefully investigate their contributions and limitations. Section 3 describes in detail the construction of our proposed hybrid model, the PRCNN, for music genre classification. In Section 4, we conduct various experiments based on two datasets and demonstrate the validity of our proposed architecture. Finally, we draw conclusions in Section 5.

II. RELATED WORK

Music genre classification, which is used to categorize and describe enormous amounts of music, is a widely studied area in music information retrieval [5]. Various studies indicate that feature extraction, a crucial component of music genre classification, can greatly affect the performance of music genre classification. Thus, most existing works focus on extracting robust features from the music samples using various approaches to improve the classification performance. Motivated by the successes in computer vision [44], CNNs have also attracted much attention in the field of music genre classification. By constructing deep networks, CNNs have a powerful capacity to learn the more representative features of the music samples. In addition, CNNs require less engineering effort and little prior knowledge of the particular field.

Li et al. [37] demonstrate that the variations of the musical patterns obtained with certain transformations, such as the fast Fourier transform (FFT) and mel-frequency cepstral coefficient (MFCC), are similar to that of images, which work well with CNNs in image classification [31]. Moreover, they prove that CNNs are feasible for the automatic extraction of music pattern features. They feed the extracted features into a classifier, such as an SVM or decision tree, and implement genre classification by using majority voting. Although their work shows an opportunity to replace handcrafted features, the performance of their proposed structure on the testing data is not as good as on the training data. Zhang et al. [38] proposed two networks to improve the performance of CNNs in music genre classification. To offer more statistical information to the following layers,the max-pooling layer is operated in conjunction with the average-pooling layers in the networks. Furthermore, to learn more representative music features from deeper networks, they utilize the shortcut connections inspired by residual learning [45] in another network. In addition, their proposed CNNs show an improved music genre classification performance compared with that of the previous approaches applied to the GTZAN [5] dataset. However, as mentioned in the previous section, some of the temporal information in the music patterns that is crucial for music genre classification may be lost by CNNs. Considering the lost temporal information, Choi et al. [46] design a hybrid model named the convolutional recurrent neural network (CRNN). They use a 2-layer RNN with gated recurrent units (GRU) [47] as the temporal summarizer following the top of the CNN structure. Compared with the three existing CNNs described in [47], the CRNN shows an improved performance in music classification by learning more temporal information. However, the hybrid model also has limitations that impair the performance of music classification. Even though the CRNN structure incorporates an RNN as the temporal summarizer, its performance heavily depends on the results of the previous convolutional layers. Moreover, the temporal relationships of the original music samples are partly lost during convolution.

To preserve the spatial features and temporal frame orders of the original music samples as much as possible, we carefully design a hybrid model that consists of parallel CNN and Bi-RNN blocks. In the next section, we will describe our proposed hybrid architecture for music genre classification in detail.

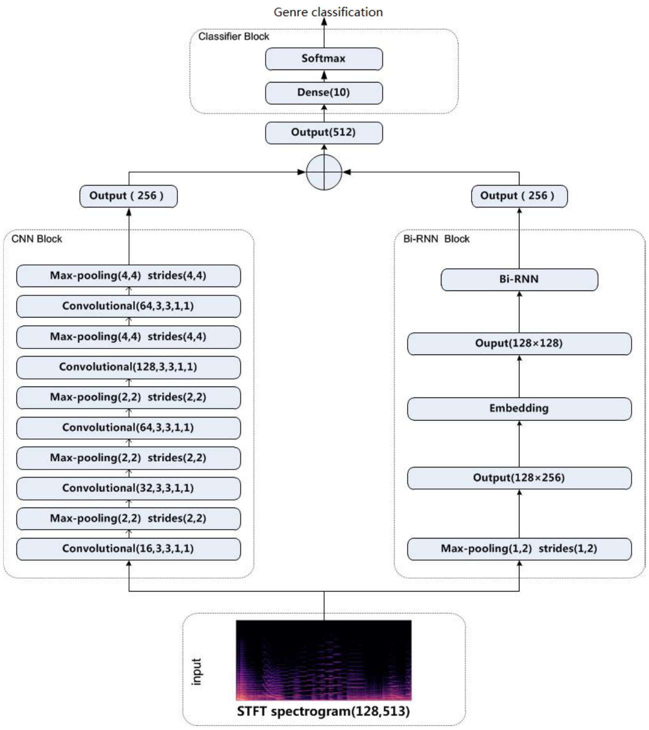
Genre classification
Classifier Block
Softmax
Dense(10)
Output(512)
Output (256)
Output (256)
CNN Block
Max-pooling(4,4) strides(4,4)
Convolutional(64,3,3,1,1)
Max-pooling(4,4) strides(4,4)
Convolutional(128,3,3,1,1)
Max-pooling(2,2) strides(2,2)
Convolutional(64,3,3,1,1)
Max-pooling(2,2) strides(2,2)
Convolutional(32,3,3,1,1)
Max-pooling(2,2) strides(2,2)
Convolutional(16,3,3,1,1)
Bi-RNN Block
Bi-RNN
Ouput(128×128)
Embedding
Output(128×256)
Max-pooling(1,2) strides(1,2)
input
STFT spectrogram(128,513)

FIGURE 1. The network architecture of the PRCNN.

III. METHODOLOGY

As illustrated in Figure 1, our proposed hybrid architecture is divided into four blocks: the input, CNN, Bi-RNN and classifier blocks. We use the short-term Fourier transform (STFT) spectrogram of the music samples as the input for our network. The input is actually a 128×513 matrix, which is fed simultaneously into the parallel CNN and Bi-RNN blocks to extract features. As mentioned above, CNNs have excellent capability in extracting the spatial features of music, such as the timbre features. However, there is some important sequential information in the STFT music spectrograms that may be lost during the CNN training. Thus, the parallel Bi-RNN block is employed as a supplement to extract the temporal frame orders from the spectrogram. Then, the outputs of the two parallel blocks are fused into one feature vector, which will then be classified. After a fully connected layer, we obtain a 10-dimensional vector and feed it into a softmax function, which produces the probabilities of 10 genres. In our architecture, the maximum probability is chosen as the predicted genre of the testing sample.

As mentioned in Section 1, feature extraction is a crucial component that substantially affects the performance of music genre classification. Therefore, in the rest of this section, we describe in detail the parallel CNN and Bi-RNN blocks that are utilized for feature extraction.

A. CONVOLUTIONAL NEURAL NETWORK BLOCK

As depicted in Figure 1, except for the input layer, the CNN block of our proposed hybrid architecture has 10 layers with weights, including five convolutional layers and five max-pooling layers. In the CNN block, the first convolutional layer filters the input spectrogram with 16 3×3 kernels. Meanwhile, 1×1 padding is added to reduce the marginal information loss during convolution. Similar to the first convolutional layer, each kernel in our CNN block has a size of 3×3, and 1×1 padding is added in each convolutional layer. The remaining convolutional layers have 32, 64, 128 and 64 filters, respectively.

After each convolutional layer, a max-pooling operation is followed to further process the output of the previous convolutional layer. In the CNN block, the first three max-pooling layers output the maximum value within a 2×2 rectangular neighborhood with a 2×2 stride. The upper two max-pooling layers report the maximum value of a 4×4 region with a 4×4 stride to learn more robust representations. The max-pooling operation can extract the most prominent music features, such as the amplitude, and reduce the computational load for the upper layers. Moreover, the max-pooling operation can also provide translation invariance and reduce overfitting by subsampling.

In all the convolutional layers, we use rectified linear units (ReLUs) [48] as the activation functions. In contrast to the sigmoid function, the ReLU activation function is define as $f(x) = max\ (0, x)$, and thus it does not become saturated when the neuron is active. Compared with the traditional sigmoid and tanh activation functions, the ReLU function provides faster convergence and it has a notable ability to mitigate the vanishing gradient problem. After each convolution, we also use batch normalization (BN) [49] to speed up the training

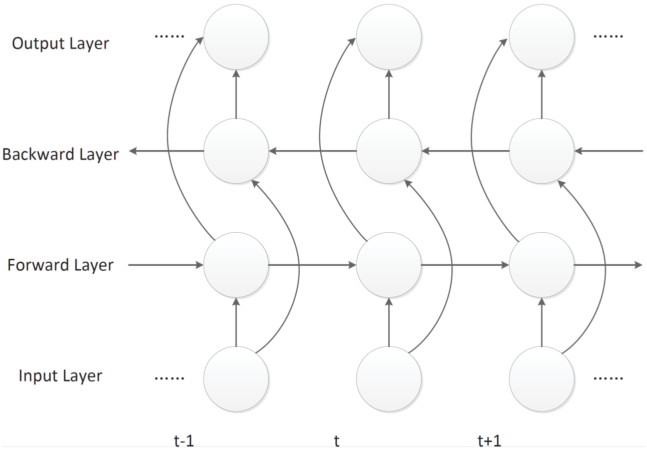

FIGURE 2. The network architecture of the Bi-RNN block.

process.

The output of our CNN block is flattened into a 256-dimensional vector, X_{cnn}:

$$X_{cnn} = (X_1, X_2, ..., X_{256})^T. \tag{1}$$

This output will be fed later into the classifier block of our architecture in conjunction with the output of the Bi-RNN block.

B. BIDIRECTIONAL RECURRENT NEURAL NETWORK BLOCK

As illustrated in Figure 1, excluding the input layer, the Bi-RNN block consists of 5 layers. In this block, the input spectrogram is processed by a max-pooling layer to reduce its dimension first. After this step, the dimension of the spectrogram is reduced to 128×256. Since the upper Bi-RNN layer has a complex structure, we employ an embedding layer for further dimension reduction to decrease the number of parameters. After embedding, a 128×128 feature map is fed into a 1-layer bidirectional RNN with GRU activation units for feature extraction, as illustrated in Figure 2. Similar to the output of the CNN block, our Bi-RNN block produces a 256-dimensional vector, which is defined as:

$$X_{rnn} = (X_1, X_2, ..., X_{256})^T. \tag{2}$$

The outputs, X_{cnn} and X_{rnn}, will be fused into a 512-dimensional feature representation that is used for classification.

Traditional recurrent neural networks (RNNs) take advantage of only previous contexts and ignore the backward dependencies, which are also important for feature extraction. However, many applications demonstrate that the prediction obtained from testing a sample heavily depends on the whole input sequence, including the past and future information. Another limitation of traditional RNNs is that they suffer from the well-known problems of vanishing and exploding gradients when dealing with long-term dependencies. Thus, in our hybrid architecture, we exploit a bidirectional RNN layer with GRUs instead of a traditional RNN layer to improve the feature extraction performance. The structure of the Bi-RNN block is shown in Figure 2, and we will describe it in detail later in the paper.

The design of the Bi-RNN block is motivated by two main considerations: 1) an RNN with GRUs is used to extract more of the temporal features that are lost in CNNs, and 2) the past and future information in a whole sequence is fully exploited to extract more representative features.

1) GATED RECURRENT UNITS

The gated recurrent unit (GRU) is proposed in [47] to adaptively capture the information from the variable-length sequences in the recurrent blocks. The traditional RNN has the well-known problems of vanishing and exploding gradients, and the GRU can capture the temporal correlations from the music samples and overcome these problems. The GRU integrates the input and forget gates into one "update gate" and appends a "reset gate". For the forward layer in the i^{th} unit, the activation, $h_i^{(t)}$, at time t is calculated by the previous activation, $h_i^{(t-1)}$, and the current candidate activation:

$$h_i^{(t)} = u_i^{(t)}\tilde{h}_i^{(t)} + (1 - u_i^{(t)})h_i^{(t-1)}, \tag{3}$$

where u stands for "update gate" and $\tilde{h}_i^{(t)}$ denotes the candidate activation. The "update gate" is used to determine how much the unit updates from its activation:

$$u_i^{(t)} = \sigma(b_u + [W_u x^{(t)}]_i + [U_u h^{(t-1)}]_i), \tag{4}$$

where b, W and U denote the biases, input weights and recurrent weights of the i^{th} GRU, respectively. The i^{th} element of a vector is denoted by $[.]_i$. The input vector at time t is defined as $x^{(t)}$. The candidate activation, $\tilde{h}_i^{(t)}$, is computed similarly to the "update gate":

$$\tilde{h}_i^{(t)} = tanh(b + [Wx^{(t)}]_i + [U(r^{(t)} \otimes h^{(t-1)})]_i), \tag{5}$$

where r represents the "reset gate" and denotes an elementwise multiplication operation. If $r(t)$ is close to 0, the "reset gate" is off and the unit will forget the past information. The "reset gate" is defined with the following formula:

$$r_i^{(t)} = \sigma(b_r + [W_r x^{(t)}]_i + [U_r h^{(t-1)}]_i). \tag{6}$$

The update and reset gates can separately "neglect" parts of a vector. The "update gates" decide how much the past states should impact the current states. The "reset gates" provide a nonlinear effect on the correlation between the past and future states. They decide which parts should be considered in future states.

2) BIDIRECTIONAL RECURRENT NEURAL NETWORK

In the parallel RNN block, we utilize an RNN with GRUs in both the forward and backward directions. As illustrated in Figure 2, the input layer is fed into both the forward and backward layers. Meanwhile, the output layer is produced by the two bidirectional layers. However, the two reverse layers have no direct connections. In our bidirectional architecture, the forward GRUs are calculated by the past states along the positive time axis, while the back forward GRUs are computed by the future states along the reverse time axis. For instance, the activation at time t of the backward GRUs is calculated by the future activation, $h_i^{(t+1)}$, and the current candidate activation, $\tilde{h}_i^{(t)}$.

$$h_i^{(t)} = u_i^{(t)}\tilde{h}_i^{(t)} + (1 - u_i^{(t)})h_i^{(t+1)}. \tag{7}$$

Similarly, the other formulas are computed along the reverse time axis. In contrast to the unidirectional architecture, the Bi-RNN with GRUs can learn more powerful representations by taking advantage of the whole sequence.

C. FEATURE FUSION AND THE CLASSIFIER BLOCK

As mentioned above, the Bi-RNN block utilized in our architecture is a supplement for CNN block to extract features. To take full advantage of the two learned feature maps, we decide to fuse them into one more powerful representation. The representation is actually a 512-dimensional vector and defined as:

$$F = x_{cnn} \oplus x_{rnn}, \tag{8}$$

where F refers to the fused feature vector and $\oplus$ indicates a simple concatenation of the two outputs of the CNN and Bi-RNN blocks. After feature fusion, the syncretic vector is fed into a fully connected layer and softmax layer successively. Then, a 10-dimensional vector, $\hat{y}$, is acquired, which is computed by:

$$\hat{\boldsymbol{y}} = softmax(W^T F + b), \tag{9}$$

where $(W^T F + b)$ represents an affine transformation with an additional bias, which maps the fused 512-dimensional vector into a 10-dimensional feature vector, v. Then, each value in vector v is calculated by the softmax function into $P(i)$, which is displayed as follows:

$$P(i) = \frac{exp(v^{(i)})}{\sum_{i=1}^{k} exp(v^{(i)})}, \tag{10}$$

where $P(i)$ represents the probability of belonging to a particular music genre and $v^{(i)}$ denotes the i^{th} value of vector v. After the softmax function, the result, $\hat{y}$, which is a vector with values in the range [0, 1], denotes a categorical probability distribution over the different genres.

The loss function we adopt for training is the crossentropy loss, whose formula is defined as follows:

$$L = crossentropy(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log \hat{y_{ij}} \tag{11}$$

where n denotes the number of samples and m represents the number of categories.

The gradients of X_{cnn} and X_{rnn} are used in the backward process and are computed as follows:

$$\begin{aligned} \left[\frac{\partial L}{\partial X_{cnn}}, \frac{\partial L}{\partial X_{rnn}}\right] &= W\frac{\partial L}{\partial F} \\ &= W\frac{\partial L}{\partial(X_{cnn} \oplus X_{rnn})} \\ &= W(\hat{\boldsymbol{y}} - \boldsymbol{y}) \end{aligned} \tag{12}$$

IV. EXPERIMENTS

To demonstrate the outstanding performance of our proposed hybrid architecture in time-series data classification, experiments are conducted on music data as an example. We conduct various contrastive experiments on both the GTZAN and Extended Ballroom datasets. The experimental results verify that our proposed PRCNN outperforms the previous approaches. Moreover, we also verify the effectiveness of the parallel RNN in supplementing the CNN for feature extraction.

A. DATASET DESCRIPTION

Two classic datasets are utilized in our experiments. One is the GTZAN dataset [5], and the other191 is the Extended Ballroom [50] dataset.

1) GTZAN

The GTZAN dataset has been used as a benchmark for various music genre classification systems. It consists of 1000 song excerpts that are evenly distributed into ten different genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock. Each song is approximately 30 seconds in duration and is sampled at a rate of 22050 Hz at 16 bits.

2) EXTENDED BALLROOM

The Extended Ballroom dataset is a modified version of the Ballroom dataset [51], and consists of specific Ballroom dance subgenres. The Extended Ballroom dataset is approximately 4 times larger than the GTZAN dataset. It contains 4180 tracks that are 30 seconds in duration, which are divided into 13 rhythm classes: chacha(455), jive(350), quickstep(497), rumba(470), samba(468), tango(464), Viennese waltz(252), waltz(529), foxtrot(507), pasodoble(53), salsa(47), slow waltz(65), and wcswing(23).

B. EXPERIMENTAL SETUP

Deep neural networks require a large amount of input data to learn representative features. However, the two datasets used in our experiments contain only 1000 song excerpts and 4180 music tracks, which is insufficient for deep neural networks.

To increase the number of training songs, we divide each song excerpt into shorter 3 second music clips with a 50% overlap. Thus, the augmented datasets ensure that overfitting is avoided to some extent and better performance is achieved in feature learning. Similar to the process in [52], [38], we apply the fast Fourier transform (FFT) to frames of length 1024 at a 22050 kHz sampling rate with a 50% overlap and use the absolute value of each FFT frame. We finally construct a STFT spectrogram with 128 frames, and each frame is represented as a 513-dimensional vector.

For each dataset, we divide the song excerpts into training, validation and testing datasets with proportions of 8/1/1, respectively. The songs are proportionally distributed by genre in the training, validation and testing sets. Our experiments are executed with 10-fold cross validation, and the performance of music genre classification is measured by classification accuracy. To avoid reporting all the results, our experimental results reported below are averaged over ten runs.

In our experiments, we choose the Adam algorithm [53] as the optimization algorithm to train our parallel networks. The number of training epochs is set at 100, and the learning rate linearly decreases within the range of [0.002, 0]. To prevent our networks from overfitting, we employ the dropout technique in the CNN and Bi-RNN blocks with dropout rates of 0.2 and 0.5, respectively. Moreover, after fusing the outputs of the CNN and Bi-RNN blocks, we also apply dropout to the syncretic feature vector with a rate of 0.5. We use batches of 64 samples and shuffle all the samples after each epoch.

C. RESULTS

1) PERFORMANCE ON THE GTZAN DATASET

To validate the effectiveness of our proposed approach in music genre classification, we conduct our experiments on the GTZAN dataset, which is a benchmark in this field. The music genre classification accuracies of our proposed PRCNN model are reported in Table 1. For comparison, we also summarize the results of the other models that were applied to the same dataset presented in [38]. To validate the effectiveness of the proposed PRCNN network, two types of CNN-based networks and BRNN networks with different layers are employed as comparison algorithms. As summarized in Table 1, the hybrid architecture, which contains parallel CNN and Bi-RNN blocks, outperforms all the previous results listed above. However, since the GTZAN dataset has only 1000 song excerpts, the problem of overfitting cannot be averted. Thus, we utilize a 1-layer RNN network instead of a 2-layer RNN network for feature extraction. As shown in Table 1, the improved accuracy shows that exploiting the CNN in conjunction with a 1-layer RNN has better performance than using the 2-layer RNN. The results listed in Table 1 demonstrate that our parallel network can significantly improve the performance of music genre classification on the GTZAN dataset compared to using only the CNN network or BRNN network alone.

TABLE 1. GENRE CLASSIFICATION RESULTS ON THE GTZAN DATASET

Methods	Features	Accuracy
CNN+2-layer RNN	**STFT**	**88.8**%
CNN+1-layer RNN	**STFT**	**90.2**%
2-layer BRNN [43]	STFT	76.2%
1-layer BRNN [43]	STFT	72.9%
CNN1 [38]	STFT	84.8%
CNN2 [38]	STFT	87.4%
KCNN(k=5) +SVM [54]	Mel-spectrum, SFM, SCF	83.9%
DNN (ReLU + SGD + Dropout) [52]	FFT (aggregation)	83.0%
Multilayer invariant representation [55]	STFT with log representation	82.0%

TABLE 2. GENRE CLASSIFICATION RESULTS ON THE EXTENDED BALLROOM DATASET

Methods	Features	Accuracy
CNN+2-layer RNN	**STFT**	**92.5**%
CNN+1-layer RNN	**STFT**	**92.3**%
2-layer BRNN [43]	STFT	90.3%
1-layer BRNN [43]	STFT	88.7%
Transform learning [56]	convnet feature	86.7%
Transform learning [56]	MFCCs + convnet feature '12345'	81.9%

2) PERFORMANCE ON THE EXTENDED BALLROOM DATASET

Since the GTZAN dataset has only 1000 song excerpts, the problem of overfitting occurs easily in most situations. Thus, we also conduct experiments on the Extended Ballroom dataset, which has approximately 4 times more music clips, to further verify the effectiveness of our proposed PRCNN. Hence, the overfitting problem of the PRCNN network with a 2-layer RNN network is reduced. As shown in Table 2, the classification accuracy of the PRCNN with a 2-layer RNN network outperforms that with a 1-layer RNN network. The PRCNN can utilize the short-term and long-term temporal music spectrogram information simultaneously, which ensures that the discriminative music information is extracted by the PRCNN. However, the RNN cannot fully extract the short-term information, and the CNN cannot fully model the long-term temporal information. As shown in Table 2, the PRCNN outperforms the 1-layer RNN network and the 2-layer RNN network. Compared to the other music gene classification methods, the PRCNN can also achieves the best result.

3) IMPACT OF THE PARALLEL RNN

As discussed previously, we have carefully analyzed the existing achievements of deep learning in the music genre classification field. According to previous works, we conclude that employing CNNs as feature extractors is not sufficient to capture the representative features of the music samples. In this situation, we suggest that RNNs should be used simultaneously in feature extraction to acquire the temporal frame orders for a better classification performance.

TABLE 3. IMPROVED PERFORMANCE WITH THE RNN FOR DIFFERENT CNNS ON THE GTZAN DATASET

CNNs	Without RNN	With RNN
Our CNN	88.0%	**92.0%**
AlexNet	81.4%	**88.8%**
VGG-11	86.8%	**88.7%**
ResNet-18	86.8%	**87.6%**

TABLE 4. IMPROVED PERFORMANCE WITH THE RNN FOR DIFFERENT CNNS ON THE EXTENDED BALLROOM DATASET

CNNs	Without RNN	With RNN
Our CNN	92.2%	**92.5%**
AlexNet	83.5%	**92.0%**
VGG-11	92.3%	**93.4%**
ResNet-18	93.1%	**93.38%**

To validate the effectiveness of the additional parallel Bi-RNN block, we designed contrastive experiments on both the GTZAN and Extended Ballroom datasets. We first discard the parallel Bi-RNN in our architecture and utilize our CNN alone. Meanwhile, we also select three typical CNNs that have excellent classification performance. Then, we conducted experiments by combining these CNNs with a parallel Bi-RNN for feature extraction. All the results depicted in the following tables reveal that the models with an RNN have better performances than those using CNNs alone. The improved performance demonstrates the effectiveness of our additional parallel RNN.

The three typical CNNs that are utilized in our experiments are introduced briefly:

• AlexNet: The Alexnet [57]model was proposed for image classification in the ImageNet LSVRC-2010 contest. It consists of five convolutional and three fully connected layers and attains a low error rate in training millions of images.

• VGG-11: The VGGNet models, a series of models proposed in [58], achieved the state-of-the-art accuracy in the ILSVRC-2013 competition. VGG-11 is a VGGNet with 11 weight layers that has better performance on small-scale datasets, such as the GTZAN and Extended Ballroom datasets utilized in our experiments.

• ResNet-18: The ResNet [45] model can obtain a higher accuracy from deep neural networks by greatly increasing the depth with residual blocks. The ResNet-18 model, as its name suggests, is a deep convolutional neural network that consists of 18-layer residual nets.

It is worth noting that the number of training songs utilized in our experiments is limited for the typical CNNs mentioned above. Thus, to avert the problem of overfitting, the CNNs used in our experiments are not very deep. In addition, we halved the number of channels in the three CNNs. Moreover, in the VGG-11 and AlexNet models, we implemented the fully connected layers with a size of 512 instead of 4096.

As illustrated in Tables 3 and 4, all the CNNs, including the CNNs in our proposed architecture, have a better performance when a parallel RNN is added rather than when using the CNNs alone. Therefore, adding a parallel RNN for feature extraction can improve the performance of music genre classification, and moreover, our Bi-RNN is extensible to different CNNs to improve music genre classification performance.

V. CONCLUSION

In this paper, we propose a hybrid architecture, the PRCNN, to improve the performance of music genre classification. This end-to-end learning architecture consists of parallel CNN and Bi-RNN blocks for feature extraction. The CNN block focuses on extracting the spatial features from the spectrograms of the music samples. In contrast, the Bi-RNN block is designed to model the temporal frame orders that are lost in the CNN. Furthermore, the bidirectional architecture can make current states depend on not only the previous information but also the future contexts. The outputs of the two parallel blocks are fused into a more powerful feature vector for music classification. We conducted experiments on both the GTZAN and Extended Ballroom datasets to verify the effectiveness of the PRCNN. The experimental results presented in our paper adequately demonstrate that our proposed PRCNN outperforms the previous works in music genre classification. Moreover, to verify the extensibility of the additional parallel RNN, we employ three typical CNNs for evaluation. The results show that all the CNNs with a parallel RNN block achieve better performance than CNNs alone.

ACKNOWLEDGMENT

This study was funded by LiaoNing Revitalization Talents Program No. XLYC1806006, and National Natural Science Foundation of People's Republic of China No. 61672130, No. 61972064. All the authors declare that they have no conflict of interest.

REFERENCES

[1] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," IEEE/ACM Transactions on Networking, vol. 27, no. 3, pp. 1028-1042, 2019.

[2] T. Qiu, B. Li, W. Qu, E. Ahmed, and X. Wang, "TOSG: A topology optimization scheme with global-small-world for industrial heterogeneous Internet of Things," IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3174-3184, 2019.

[3] L. Zhang , S. Wang , G.B. Huang , W. Zuo , J. Yang , D. Zhang, "Manifold Criterion Guided Transfer Learning via Intermediate Domain Generation," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 12, pp. 3759-3773, 2019.

[4] L. Zhang , W. Zuo ; J. Yang ; D. Zhang, "LSDT: Latent Sparse Domain Transfer Learning for Visual Adaptation," IEEE Transactions on Image Processing, vol. 25, no. 3, pp. 1177-1191, 2016.

[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," IEEE Transactions on speech and audio processing, vol. 10, pp. 293–302, 2002.

[6] J. Shawe-Taylor and A. Meng, "An investigation of feature models for music genre classification using the support vector classifier," in 6th

International Conference on Music Information Retrieval (ISMIR 2005), pp. 604-609, 2005.

[7] K. West and S. Cox, "Finding An Optimal Segmentation for Audio Genre Classification," in IEEE Conference on ISMIR, 2005, pp. 680–685.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification (2nd Edition)," En Broeck the Statistical Mechanics of Learning Rsity, 2000.

[9] H. Wang, L. Feng, J. Zhang, and Y. Liu, "Semantic discriminative metric learning for image similarity measurement," IEEE Transactions on Multimedia, vol. 18, pp. 1579–1589, 2016.

[10] Z. Huang, H. Zhu, J. T. Zhou, and X. Peng, "Multiple Marginal Fisher Analysis," IEEE Transactions on Industrial Electronics, vol. 1, pp. 1–1, 2018.

[11] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-Graph for Robust Subspace Learning and Subspace Clustering," IEEE Transactions on Cybernetics, vol. 47, pp. 1053–1066, 2017.

[12] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce, "Music genre classification via joint sparse low-rank representation of audio features," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, pp. 1905–1917, 2014.

[13] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," IEEE transactions on multimedia, vol. 13, pp. 303–319, 2011.

[14] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA Domain Name Classification Method Based on Long Short-Term Memory with Attention Mechanism," Applied Sciences, vol. 9(20), pp. 4205, 2019.

[15] X. Peng, J. Lu, Z. Yi, and R. Yan, "Automatic Subspace Learning via Principal Coefficients Embedding," IEEE Transactions on Cybernetics, vol. 47, pp. 3583–3596, 2017.

[16] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A Unified Framework for Representation-Based Subspace Clustering of Out-of-Sample and Large-Scale Data," IEEE Transactions on Neural Networks and Learning Systems, vol. 27, pp. 2499–2512, 2016.

[17] S. Xu, S. Liu, and L. Feng, "Neighborhood Graph Embedding for Nodes Clustering of Social Network," in IEEE 21st International Conference on High Performance Computing and Communications, 2019, pp. 718–725.

[18] S. Xu, S. Liu, J. Zhou, and L. Feng, "Fuzzy rough clustering for categorical data," International Journal of Machine Learning and Cybernetics, vol. 10, pp.3213–3223, 2019.

[19] S. Xu, L. Feng, S. Liu, J. Zhou, and H. Qiao, "Multi-feature weighting neighborhood density clustering," Neural Computing and Applications, Sep. 2019. DOI:10.1007/s00521-019-04467-4.

[20] J. Wang, S. Xu, B. Duan, C. Liu, and J. Liang, "An Ensemble Classification Algorithm Based on Information Entropy for Data Streams," Neural Processing Letters, vol.50, pp. 2101–2117, 2019.

[21] B. K. Baniya, D. Ghimire, and J. Lee, "A novel approach of automatic music genre classification based on timbrai texture and rhythmic content features," in Advanced Communication Technology (ICACT), 2014 16th International Conference on, 2014, pp. 96–102.

[22] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," Neurocomputing, vol. 70, pp. 489–501, 2006.

[23] M. Kim, S. W. Kim, and Y. Han, "EPSim-C: A Parallel Epoch-Based Cycle-Accurate Microarchitecture Simulator Using Cloud Computing," Electronics 2019, vol. 8, 716 DOI:10.3390/electronics 8060716.

[24] J. Su, L. Gao, W. Li, Y. Xia, N. Cao, and R. Wang, "Fast Face Tracking-by-Detection Algorithm for Secure Monitoring," Applied Sciences 2019, 9(18),3774. DOI: 10.3390/app9183774

[25] L. Feng, S. Xu, F. Wang, S. Liu, and H. Qiao, "Rough extreme learning machine: A new classification method based on uncertainty measure," Neurocomputing, vol. 325, pp. 269–282, 2019.

[26] R. Yang, S. Xu, and L. Feng, "An ensemble extreme learning machine for data stream classification," Algorithms 2018, vol. 11, 107. DOI:10.3390/a11070107.

[27] L. Breiman, "Bagging predictors," Machine learning, vol. 24, pp. 123–140, 1996.

[28] A. F. Arabi and G. Lu, "Enhanced polyphonic music genre classification using high level features," in Signal and Image Processing Applications (ICSIPA), IEEE International Conference on, 2009, pp. 101–106.

[29] R. Sarkar and S. K. Saha, "Music genre classification using EMD and pitch based feature," in Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on, 2015, pp. 1–6.

[30] Q. Hu, H. Wang, T. Li, and C. Shen, "Deep CNNs With Spatially Weighted Pooling for Fine-Grained Car Recognition," IEEE Transactions on Intelligent Transportation Systems, vol. 18(11), pp.3147-3156, 2017.

[31] D.C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 2011, Vol. 22, pp. 1237.

[32] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer Hashing: From Shallow to Deep," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, pp.6191–6201, 2018.

[33] X. Peng, J. Feng, S. Xiao, W. Yau, J. T. Zhou, and S. Yang, "Structured Auto Encoders for Subspace Clustering," IEEE Transactions on Image Processing, vol. 27, pp. 5076–5086, 2018.

[34] L. Zhang, J. Liu, B. Zhang, D. Zhang, C. Zhu, "Deep Cascade Model-Based Face Recognition: When Deep-Layered Learning Meets Small Data," IEEE Transactions on Image Processing, vol. 29, pp. 1016-1029, 2019.

[35] L. Zhang, Q. Duan, D. Zhang, W. Jia, X. Wang, "AdvKin: Adversarial Convolutional Network for Kinship Verification," IEEE Transactions on Cybernetics, vol. 50, pp. 1–14, 2020.

[36] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 6964–6968.

[37] T. L. Li, A. B. Chan, and A. Chun, "Automatic musical pattern feature extraction using convolutional neural network," in Proc. Int. Conf. Data Mining and Applications, 2010.

[38] W. Zhang, W. Lei, X. Xu, and X. Xing, "Improved Music Genre Classification with Convolutional Neural Networks," INTERSPEECH, pp. 3304–3308, 2016.

[39] T. Qiu, H. Wang, K. Li, H. Ning, A. K. Sangaiah, and B. Chen, "SIGMM: A novel machine learning algorithm for spammer identification in industrial mobile cloud computing," IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2349-2359, 2019.

[40] C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, "CVCG: Cooperative V2V-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks," IEEE Transactions on Mobile Computing, vol. 18, no. 12, pp. 2811-2828, 2018.

[41] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, pp. 179–211, 1990.

[42] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, pp. 179–211, 1990.J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on, IEEE, 2016, pp. 1–6.

[43] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, pp. 2673–2681, 1997.

[44] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," IEEE transactions on neural networks, vol. 8, pp. 98–113, 1997.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[46] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017. DOI:10.1109/ICASSP.2017.7952585

[47] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," Computation and Language, arXiv preprint arXiv:1409.1259 2014.

[48] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807–814.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International Conference on Machine Learning, 2015, pp. 448–456.

[50] U. Marchand and G. Peeters, "The extended ballroom dataset," 2016.

[51] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer, "Evaluating rhythmic descriptors for musical genre classification," 2004.

[52] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 6959–6963.

[53] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Computer Science. arXiv preprint arXiv:1412.6980 2014.

[54] P. Zhang, X. Zheng, W. Zhang, S. Li, S. Qian, W. He, S. Zhang, and Z. Wang, "A Deep Neural Network for Modeling Music," in the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 379–386.

[55] C. Zhang, G. Evangelopoulos, S. Voinea, L. Rosasco, and T. Poggio, "A deep representation for invariance and music classification," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 2014, pp. 6984–6988.

[56] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," Computer Vision and Pattern Recognition. arXiv preprint arXiv:1703.09179 2017.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol.1, pp. 1097–1105, 2012.

[58] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Computer Science. arXiv preprint arXiv:1409.1556 2014.

RUI YANG received BS degree from Dalian University of Technology, china, in 2004, the MS degree from Harbin Institute of Technology, China, in 2009. He is a associate professor in Shenyang Normal University. Currently, he is working toward the PhD degree in the School of Computer Science and Technology, Dalian University of Technology, China. His research interests include attribute reduction, image processing, machine learning.

LIN FENG received the BS degree in electronic technology from Dalian University of Technology, China, in 1992, the MS degree in power engineering from Dalian University of Technology, China, in 1995, and the PhD degree in mechanical design and theory from Dalian University of Technology, China, in 2004. He is currently a professor and doctoral supervisor in the School of Innovation Experiment, Dalian University of Technology, China. His research interests include intelligent image processing, robotics, data mining, and embedded systems.

HUIBING WANG received the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. During 2016 and 2017, he is a visiting scholar at the University of Adelaide, Adelaide, Australia. He has authored and co-authored more than 20 papers in some famous journals or conferences, including TMM, TITS, TSMCS, ECCV, etc. Furthermore, he serves as reviewers for TNNls, Nurocomputing, PR Letters, etc. Now, he is a postdoctor in Dalian Maritime University, Dalian, Liaoning, China. His research interests include computing vision and machine learning.

JIANING YAO received the BS degree from Dalian Maritime University, China, in 2015, and the MS degree in School of Computer Sciences from Dalian University of Technology, China, in 2018. Her research interests include video analysis and machine learning.

SEN LUO received the BS degree from Dalian Maritime University, China, in 2015, and the MS degree in School of Computer Sciences from Dalian University of Technology, China, in 2018. His research interests include pattern recognition, music analysis and machine learning.

• • •