*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# Music Genre Classification Using Duplicated Convolutional Layers in Neural Networks

*Hansi Yang, Wei-Qiang Zhang*[*]

Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China

yhs17@mails.tsinghua.edu.cn,wqzhang@tsinghua.edu.cn

## Abstract

Music genres are conventional categories that identify some pieces of music as belonging to a shared tradition or set of conventions. In this paper, we proposed an approach to improve music genre classification with convolutional neural networks (CNN). Using mel-scale spectrogram as the input, we used duplicate convolutional layers whose output will be applied to different pooling layers to provide more statistical information for classification. Also, we made some modifications on residual learning by taking more outputs from convolutional layers. By comparing two different network topologies, our experimental results on the GTZAN dataset show that the proposed method can effectively improve the classification accuracy.

**Index Terms**: Music genre classification, convolutional neural network, mel-scale spectrogram, residue learning

## 1. Introduction

In the past few years, with the prevalence of personal multimedia devices, structuring and organizing music is becoming increasingly necessary for various applications such as music recommendation and music auto-tagging. However, due to the large amount of music available on different platforms, it is undoubtedly impossible for humans to organize such a large amount of music. Genre classification is currently one of the ways used to structure the music content. An effective and precise music genre classification system is therefore urgently needed to enable automatic structuring and organization of large archives of music. As a classification problem, the typical process of an automatic genre classification system mainly consists of two steps: 1) features extraction and selection; 2) classification. Undoubtedly, finding suitable features of the audio signal is crucial to the success of the system. The common approach in the past is to extract some hand-crafted features from the original songs. Baniya et al. used timbral texture (MFCC and other spectral features) and rhythmic content features based on wavelet decomposition to improve the performance [1]. Hand-crafted features have some disadvantages. Firstly, it's difficult to design the features for a specific task, which requires expertise in specific field and engineering ingenuity. In addition, the method is hard to generalize. Different features for different tasks or even different environments need to be calculated separately.

With the development of deep learning, neural networks have shown its efficiency in different fields, including music information retrieval (MIR) [2, 3]. In this paper, we apply convolutional neural network on genre classification task and propose an approach to improve the classification accuracy with it. Our approach is using duplicate convolutional layers whose output will be applied to different pooling layers. This is a method inspired by residual learning [4] but different in principle. Instead of adding the output from convolutional layers together and feeding it into pooling layers, we feed different outputs into different pooling layers and concatenate the outputs of pooling layers as the total output. Then the total output will be fed into the fully-connected layer. This will lead to increased dimension of input for dense layers, thus may lead to better performances. We also made some modifications on previous works in residual learning [4, 5] to investigate whether taking more outputs from the convolutional layers can improve classification accuracy. This modification will be compared with the proposed method to test the effect of our approach as well. In addition, we use the mel-scale spectrogram as the input of our network. The reason is that it is a scale for the measurement of the psychological magnitude pitch [6] therefore it may be easier for neural network to extract features from it.

The rest of this paper is organized as follows. In Section 2, the related work and latest advance of deep learning in MIR is introduced. We then describe the details of our methodologies in Section 3, followed by the experimental setup and results. Finally, we draw a conclusion and describe potential future work in Section 5.

## 2. Related Work

Deep neural networks, especially convolutional neural networks (CNNs) have been applied to fields such as computer vision and speech recognition successfully. There has been a lot of interest in investigating the effect of using deep neural networks in music information retrieval (MIR). Dong [7] used CNNs to extract musical pattern features from the mel-scale spectrogram of audio signals. This work proved that CNNs had potential capacity to capture informative features from the variations of musical patterns with minimal prior knowledge needed. However, the proposed models cannot fit training set very well while training and the reported accuracy is not very satisfying. Choi et al. [8] applied Convolutional Recurrent Neural Network (CRNN), which combines convolutional neural network and recurrent neural network together, on music genre classification. Using Million Song Dataset (MSD), they achieved a satisfying accuracy. But due to the sequential feature of the recurrent neural network, it takes a long time to train the model. Also, they used a long segment of music signal (about 29 seconds) as the input, which may be not realistic in some applications. Zhang et al. [9] employed CNNs with k-max pooling layers for semantic modeling of music. The proposed method could produce more robust music representations by adding more layers. Dieleman et al. [10] investigated the performance of the features learned from raw audio signals by using CNNs. They found that the networks were able to automatically discover frequency decompositions. However, the CNN-based method did not outperform spectrogram-based approaches.

[*] corresponding author

http://dx.doi.org/10.21437/Interspeech.2019-1298

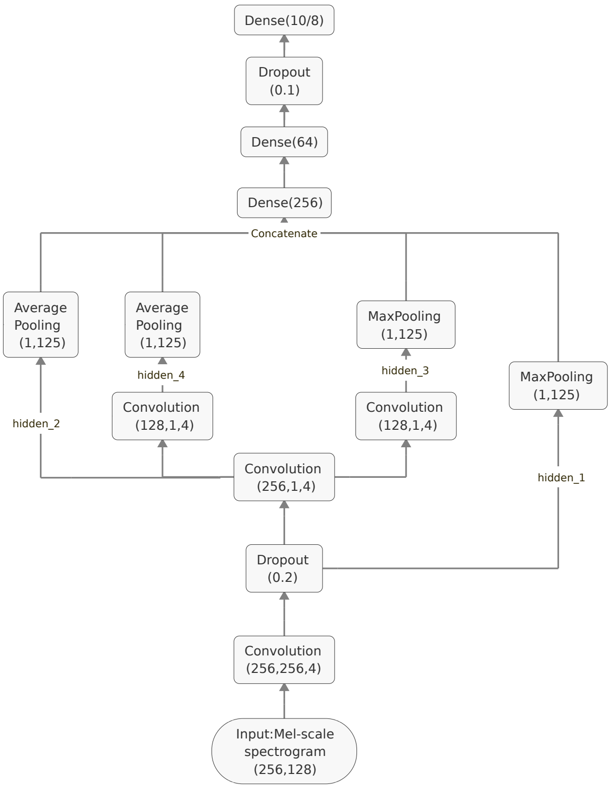

Figure 1: *The network architecture of net1.*

Motivated and inspired by the recent success of using CNNs in other fields [7, 9, 8], in this paper we propose an approach of using duplicated convolutional layers to improve music genre classification accuracy with CNNs.

## 3. Methodology

Deep neural networks alleviate the need of task-depend prior knowledge since the features are automatically tailored to the task at hand. However, the net architecture greatly affects the system performance. Thus it should be designed carefully. In this section, we describe two different network architectures (Figure 1 and Figure 2) that will be investigated in our experiments. The detailed architectures of the two nets will be described below.

### 3.1. Duplicated Convolutional Layers

Figure 1 shows the architecture of the first neural network (referred to as net1 below) used in our experiments. Both approaches we proposed are applied in this network architecture.

Our approach of using duplicated convolutional layers is partly inspired by residual learning proposed by He et al [4]. We will summarize their work briefly below. Suppose the function to be learnt by the neural networks is $H(x)$, then instead of approximating $H(x)$, they explicitly let their network approximate a residual function $F(x) := H(x) - x$ (assuming that the input and output are of the same dimensions). Thus they take some outputs from the convolutional layers in the middle and add them to the output from the last convolutional layer.

From their work, residual learning makes it easier to optimize a deeper net and the network can gain accuracy in image recognition from increased depth. However, adding the outputs together can lead to the loss of information, so instead of adding them together, we will first feed them into different pooling layers and concatenate them. The process of concatenating the output together is partly inspired by [11], but using duplicated layers and different pooling layers is not mentioned in their work. Our proposed methods of using different pooling layers will preserve distinct statistical features of the input after the pooling operation, and this can make the previous convolutional layers extract different features from its input while optimizing for better classification accuracy. Compared with the popular practice of using only one type of pooling layer after each convolutional layer, our method of using duplicated convolutional layers with different types of pooling layers can learn more features from the input without a huge growth in the number of parameters.

The input of this network is the mel-scale spectrogram of the input audio signal, which contains 128 frames and each frame has 256 mel-scale frequency bins. The first convolutional layer has 256 different kernels with a size of $256\times4$ and does not use padding. The second convolutional layer has the same number of kernels with a size of $1\times4$ and uses padding to make the output the same dimensions as the input. The third and fourth convolutional layers both takes the output of the second convolutional layer ad the input. Both of them have 128 kernels with a size of $1\times4$ , which is half of the kernels in previous convolutional layers. They also use padding to keep the output from shrinking in time axis.

After the convolutional layers, we feed all outputs from convolutional layers into pooling layers directly rather than add them together. Because of the same size and input of the third and fourth convolutional layers, we feed their outputs into different pooling layers (max-pooling and average-pooling) to provide more statistical information for the following layers. The first and second output are also fed into different kind of pooling layers for the same consideration. Then the outputs of four pooling layers is concatenated and fed into following layers. The last three layers are dense layers with 256, 64 and 10 hidden units respectively, which are used as a classifier.

Rectified linear units (ReLUs) [12] are used as the activation function in all convolutional and dense layers except for the top layer where the softmax function is applied instead. Regularizers and dropout layers is applied to this neural network to avoid overfitting. The first convolutional layer uses $\ell_2$-regularizer with a penalty-weight 0.02 and other convolutional layers use the same kind of regularizer with a different penalty-weight 0.01. Dropout layers are added after the first convolutional layer and before the last dense layer with a rate of 0.2 and 0.1 respectively to prevent overfitting.

### 3.2. Contrastive Network: Residual Network

Figure 2 shows the architecture of the second neural network (referred to as net2 below) used in our experiments. It mainly works as a contrast to test the effect of the proposed method. It is similar to net1 but with only three convolutional layers and the last convolutional layer has the same number of parameters as two duplicated convolutional layers used in net1. After the last convolutional layer, the outputs of 3 layers are added together. Then both max- and average-pooling operations are applied across the entire time axis. The last three layers are also fully-connected layers but with 128, 32 and 10 hidden units, respectively.

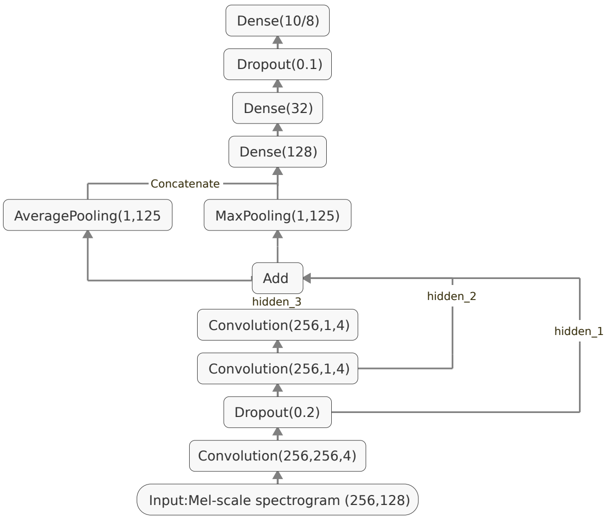

Figure 2: *The network architecture of net2.*

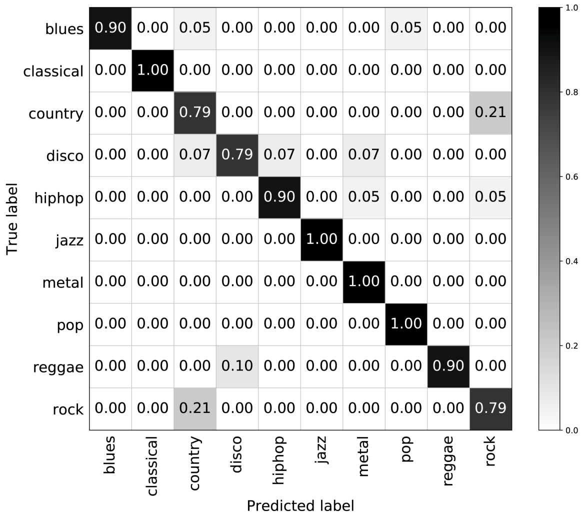

Figure 3: *Confusion matrix of net1.*

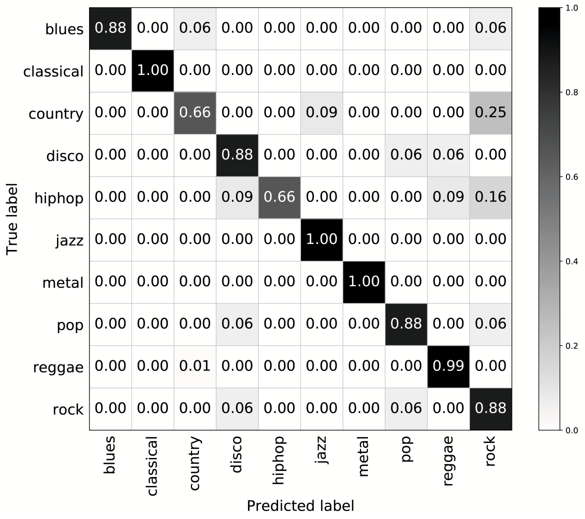

Figure 4: *Confusion matrix of net2.*

# 4. Experiments and Results

In this section, we present the results of our experiments used to evaluate the methodologies described in Section 3.

## 4.1. Dataset

We used GTZAN dataset collected by Tzanetakis and Cook [13] in our experiments. It has been widely used as a benchmark for music genre classification [14] . It contains 1000 song excerpts that are almost evenly distributed into ten different genres: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae and Rock. Each song excerpt lasts about 30 seconds and is sampled at 22050Hz, 16 bits.

In our experiments, the dataset is split into 8/1/1 training, validation and test splits. The number of songs for different genres in the training, validation and test sets is balanced. The classification accuracy was used as the measure of the performance and all the results reported below were averaged over ten runs.

## 4.2. Experimental Setup

We first calculate the mel-scale spectrogram on frames of length 1024 with an overlap of 50%. The output for each frame is a 256-dimensional vector and we used 128 frames as our input to the neural network, which is equal to approximately 3 seconds of raw audio inputs.

Both models are built with Keras and TensorFlow. When training the networks in all experiments, we used Adadelta [15] as the optimizer with the default learning rate 1.0. The loss function we chose in both networks was categorical cross-entropy. Each mini batch we used to train the network contains 50 samples. The output of the networks are the probabilities of different genres for each music clip. We added up the probabilities of the clips from the same song, and chose the genre with the maximum value as the label of the song.

## 4.3. Results

The confusion matrix of our two neural networks is plotted on Figure 3 and Figure 4, respectively.

It can be seen from the confusion matrix that both of our networks perform well on 3 genres: classical, jazz and metal, where they all achieved 100% accuracy. And net1 outperforms net2 on 4 genres: blues, country, hiphop, pop (100%) while net2 outperforms net1 on other 3 genres: disco, reggae (99%) and rock. Overall, net1 achieves an accuracy of 90.7% while net2 achieves 88.3%, which shows that the proposed approach can effectively improve the accuracy as expected, especially on pop genre. However, net2 also have some advantages on genres such as reggae, thus the proposed method can still be improved. It also should be noted that both of our networks perform worst on country genre and they all tend to classify some of them into rock genre. Previous work [7] also shows that country genre is hard to classify and suggests country music may have characteristic features (e.g., beat) that require longer time (i.e. more than 3 seconds) to capture.

Figure 5 shows the spectrograms of a 3-second pop music segment which net1 classifies as pop while net2 classifies as disco. It can be seen from the spectrograms that Mel-scale is a

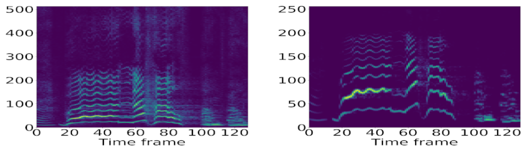

Figure 5: *Comparison of different spectrogram. (Left: Hertz-scale, Right: Mel-scale)*

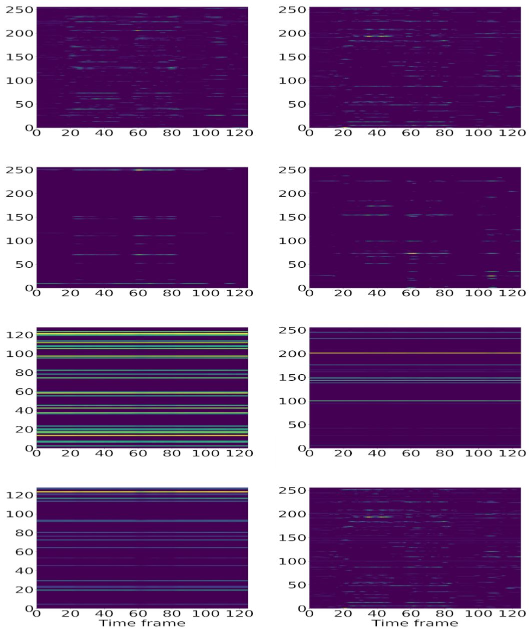

Figure 6: *Features extracted by two networks. (Left: net1, Right: net2)*

better representation of the energy distribution in the frequency domain as it more clearly shows the frequencies where much of total energy concentrates. Due to the fact that music genre is often related to the features in the frequency domain rather than time domain, Mel-scale spectrogram can be a more suitable feature for the task of music genre classification.

Figure 6 visualizes the features extracted by convolutional layers of our two networks, which are shown from top to bottom respectively. The bottom two figures of the left column are features extracted by the duplicated convolutional layers of net1. It is shown from those figures that our proposed method of using duplicated convolutional layers with different types of subsequent pooling layers can extract distinct features from the same input, thus provide more statistical information for the following dense layers and contribute to the improving classification accuracy. In contrast, the last figure on the right, which is the final output of convolutional layers and will be fed into the dense layers for classification, shows little statistical information compared with those figures on the left. Although the third figure, which is the output of the third convolutional layer, is very similar to the last two figures on the right and provides very distinct features, the adding operation disappears those features and lead to the result that there is almost no difference between the final output and the output of the first layer.

In Table 1, we compare our work with previous results on the GTZAN data set. Comparing our net2 with nnet2 from Zhang et al. [5], we can find that the use of mel-scale spectrogram and more output from convolutional layers can contribute to the improvement of classification accuracy. Our net1 result outperforms all listed previous results, which proves that the proposed approach can make a huge improvement to the classification accuracy.

Table 1: *Results of different networks on GTZAN dataset*

| Methods | Features | Accuracy |
|---|---|---|
| net1 | mel-spectrogram | 90.7% |
| net2 | mel-spectrogram | 88.3% |
| nnet1 [5] | STFT | 84.8% |
| nnet2 [5] | STFT | 87.4% |
| KCNN(k=5)+SVM [9] | mel-spectrum, SFM, SCF | 83.9% |
| DNN(ReLU+SGD +Dropout) [16] | FFT(aggregation) | 83.0% |
| Multilayer invariant representation [17] | STFT with log representation | 82.0% |

Table 2 compares the number of parameters in our two networks and two networks from Zhang et al. [5] and their accuracy. It can be seen from the table that net1 achieves better result with less parameters when compared with nnet2 [5], which prevent the issue of overfitting.

Table 2: *Number of parameters of different networks*

| Methods | Number of Parameters | Accuracy |
|---|---|---|
| net1 | 1,001,162 | 90.7% |
| net2 | 857,322 | 88.3% |
| nnet1 [5] | 464,906 | 84.8% |
| nnet2 [5] | 1,250,928 | 87.4% |

## 5. Conclusions

In this paper, we investigated the effectiveness of using CNNs for music genre classification. Our experimental results show that using duplicate convolutional layers whose output will be applied to different pooling layers are effective to improve music genre classification with CNNs. Also we tested the efficiency of a modification of taking more convolutional layers' outputs as the total output on residual learning. We found that this method can also improve classification accuracy but are less effective than the proposed method. The result also shows that they have advantages on different genres, which indicates that different genres may be easiest to classify for different network architectures.

In the future, we'll try to fuse new methods such as using an assisting Recurrent Neural Network (RNN) with the proposed approach. Also, since the spectrogram is still hand-crafted features, we'll also study end-to-end learning to extract salient musical representations from the raw audio signals directly. The reason why different genres are easiest to classify for different network architectures is also a topic that need to be investigated further.

## 6. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219.

# 7. References

[1] 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 *Proc. 16th International Conference on Advanced Communication Technology (ICACT2014)*, 2014.

[2] E. J. Humphrey, J. P. Bello, and Y. LeCun, “Feature learning and deep architectures: new directions for music informatics,” *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 461–481, Dec 2013.

[3] A. Alexandridis, E. Chondrodima, G. Paivana, M. Stogiannos, E. Zois, and H. Sarimveis, “Music genre classification using radial basis function networks and particle swarm optimization,” in *Proc. 6th Computer Science and Electronic Engineering Conference (CEEC’14)*, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] W. Zhang, W. Lei, X. Xu, and X. Xing, “Improved music genre classification with convolutional neural networks.” in *Proc. Interspeech 2016*, 2016, pp. 3304–3308.

[6] S. S. Stevens, J. Volkmann, and E. B. Newman, “A scale for the measurement of the psychological magnitude pitch,” *Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[7] M. Dong, “Convolutional neural network achieves human-level accuracy in music genre classification,” *arXiv preprint arXiv:1803.05337*, 2018.

[8] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in *Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.

[9] 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 *Proc. 5th ACM on International Conference on Multimedia Retrieval (ICMR 2015)*, 2015, pp. 379–386.

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

[11] G. Huang, Z. Liu, and K. Weinberger, “Densely connected convolutional networks,” in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 12.

[12] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proc. 27th International Conference on Machine Learning (ICML 2010)*, 2010, pp. 807–814.

[13] G. Tzanetakis, “Automatic musical genre classification of audio signals.” in *Proc. 2nd International Symposium/Conference on Music Information Retrieval (ISMIR 2001)*, 2001.

[14] B. L. Sturm, “An analysis of the gtzan music genre dataset,” in *Proc. 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM’12)*, 2012, pp. 7–12.

[15] M. D. Zeiler, “Adadelta: An adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.

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

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