*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Improved Music Genre Classification with Convolutional Neural Networks

*Weibin Zhang *,Wenkang Lei *, Xiangmin Xu, Xiaofeng Xing*

School of Electronic and Information Engineering
South China University of Technology, GuangZhou, China

eeweibin@scut.edu.cn, lei.wenkang@mail.scut.edu.cn,
xmxu@scut.edu.cn, xfxing@scut.edu.cn

## Abstract

In recent years, deep neural networks have been shown to be effective in many classification tasks, including music genre classification. In this paper, we proposed two ways to improve music genre classification with convolutional neural networks: 1) combining max- and average-pooling to provide more statistical information to higher level neural networks; 2) using shortcut connections to skip one or more layers, a method inspired by residual learning method. The input of the CNN is simply the short time Fourier transforms of the audio signal. The output of the CNN is fed into another deep neural network to do classification. By comparing two different network topologies, our preliminary experimental results on the GTZAN data set show that the above two methods can effectively improve the classification accuracy, especially the second one.

**Index Terms**: music genre classification, convolutional neural network, residual learning

## 1. Introduction

In the past few years, with the prevalence of personal multimedia devices, a large amount of music is increasingly available on various application platforms. Structuring and organising such a large amount of music is becoming impossible for humans. 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 organisation of large archives of music.

The genre of music is a kind of a high level label. As a classification problem, the typical process of an automatic genre classification system consists of three steps: 1) features such as timbre, spectro-temporal and statistical features are extracted from original audio signal; 2) some techniques are applied to select the meaningful subset of the features [1] or aggregate features [2, 3] to improve the classification accuracy; 3) a classifier based on machine learning methods is trained over the selected features to automatically classify the input music into different genres. As a crucial part of the system, finding suitable representations or features is a key factor to the success of the system. A common way to do is to extract some hand-crafted features from the original songs. This process requires expertise in specific field and engineering ingenuity. Sarkar et al. used empirical mode decomposition (EMD) to capture the local characteristics of different genres and then computed the pitch based features from the decomposed songs [4]. Baniya et al. used timbral texture (MFCC and other spectral features) and rhythmic content features based on wavelet decomposition to improve the performance [5]. These hand-crafted features have some disadvantages: firstly, its difficult to design the features for a specific task; secondly, the method is lack of universality (i.e. different features for different tasks need to be calculated separately); thirdly, the model lacks extensibility since the performance improvement of the system dose not rely on an unified framework, for example, a different feature set or classifier are usually required to achieve better classification accuracy on a different task.

With the development of deep learning, neural networks are very effective in different fields, including music information retrieval (MIR) [6, 7, 8, 9]. In this paper, we propose two ways to improve music genre classification accuracy with convolutional neural networks: 1) combining max- and average-pooling to provide more statistical information to higher level neural networks; 2) using shortcut connections to skip one or more layers, a method inspired by the residual learning [10].

The rest of this paper is organised 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 learning, especially convolutional networks (CNNs) have recently been applied to computer vision

*Both authors contributed equally to this paper.



http://dx.doi.org/10.21437/Interspeech.2016-1236

and speech recognition successfully. There has been a lot of interest in investigating unsupervised feature learning by using deep neural networks in MIR. A Convolutional Deep Belief Network (CDBN) was proposed to improved music genre and artist classification performance by using audio spectrogram and MFCC features in [11]. The features learned from unlabelled audio data are shown to perform very good on multiple music classification tasks. Li et al. used CNNs to extract musical pattern features in audio [7]. Their work proved that CNNs had potential capacity to capture informative features from the variations of musical patterns with minimal prior knowledge needed. However, their experimental results showed that the proposed models did not generalise very well to unseen testing data [7]. Zhang et al. employed CNNs with $k$-max pooling layers for semantic modelling of music [8]. The proposed method could produce more robust music representations by adding more layers. Zhang et al. built a hierarchical architecture for extracting invariant and discriminative audio representations [12]. Sander et al. 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[13].

Motivated by the recent success of using CNNs in other fields [10, 13, 7], we propose two ways to improve music genre classification accuracy using convolutional neural networks in this paper.

## 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 we need to carefully design the net. In this Section, we describe two different network architectures (Figure 1 and Figure 2) that will be investigated in our experiments.

As can be seen from Figure 1 and Figure 2, the input of the nets is Short Time Fourier Transform (STFT) magnitude spectrum, which is usually used to represent the timbre texture of music [14]. Our nets consist of two major parts: a stack CNN module used as the feature extractor to learn mid and high level features from the spectrograms, and a fully connected module (i.e. the dense layers) used as the classifier. CNNs are biologically-inspired variants of multilayer perceptrons. We will describe the architectures of the two nets in details below.

### 3.1. The network with both max and average pooling

Figure 1 shows the architecture of the first neural network (referred to as nnet1 below) used in our experiments. It contains 10 layers, including the input layer and the soft-

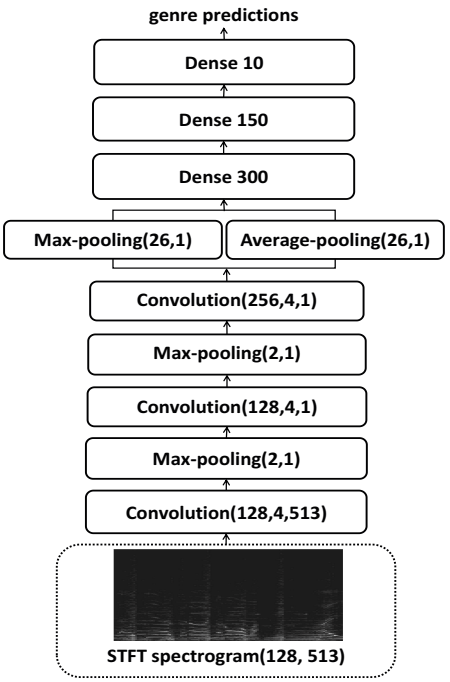


Figure 1: *The architecture of the first neural network we used (nnet1).*

max output layer. The input layer is the STFT of the input audio signal. As will be explained later, it contains about 128 frames and each frame has 513 frequency bins. The first convolutional layer has 128 different kernels of equal size. During convolution, the kernel surveys a fixed $4 \times 513$ region in the input STFT spectrogram, multiplying the input value with its associate weights in the kernel, adding the kernel bias and passing the result to the activation function. After each convolution, the kernel hops 1 step forward along the input. The 2nd and 3rd convolutional layers function very similarly to the 1st convolutional layer, with 128 and 256 feature maps respectively. Their kernel size is $4 \times 1$ and their hop size is 1. Each kernel has connections with all the feature maps in the lower layer.

CNNs exploit timely-local correlation by enforcing a local connectivity pattern between the input and the CNN neurons. The inputs of a unit in layer $m$ are from a subset of units in layer $m-1$, units that are timely contiguous to each other. In our case, each STFT frame spans 23 ms on the audio signal with 50% overlapping with adjacent frames. Thus the first convolutional layer (i.e. 2nd layer) detects basic musical patterns appear in about 127 ms. Upper convolutional layers capture musical patterns in longer windows, meaning that the neuron becomes more global. In addition, since the kernel is shared across on a feature map, it allows useful features to be detected regardless of their position in the spectrum. This weight sharing mechanism also increases learning effi-

ciency by greatly reducing the number of free parameters to be learnt. The constraints on the model enable CNNs to achieve better generalisation on lots of classification tasks.

Following each convolutional layer, except the last one, a max-pooling operation with a hop size 1 is applied as a process of non-linear subsampling. The max-pooling enables CNNs to look at non-overlapping regions of the audio signal and output the maximum value. By eliminating non-maximal values, it reduces computation for upper layers. Also, it provides a form of translation invariance. Purposely, both the max- and average-pooling operations across the entire time axis are used after the 3rd convolutional layer in order to provide more statistical information to the following layers.

The last three layers are dense layers with 300, 150 and 10 hidden units respectively, which are used as a classifier to automatically classify the input audio into different genres. The output of the last layer is the probabilities of different genres.

Rectified linear units (ReLUs) [15, 16] are used as the activation function in all convolutional and dense layers except for the top layer where the softmax function is applied instead. The ReLU activation function is defined as $f(x) = max(0, x)$. Compared with the $sigmoid$ function, ReLU does not saturate at 1 and the partial derivative of the activation function with respect to the model parameters is never 0, as long as the neural is active. During training, regularisation techniques such as Dropout is usually used to prevent the model from overfitting.

### 3.2. The Residual Network

Figure 2 shows the architecture of the second neural network (referred to as nnet2 below) used in our experiments. It's similar to the nnet1. The biggest difference between the two networks is the shortcut connections from the output of the first convolutional layer to the output of the third convolutional layer.

This network is inspired by the concept of residual learning proposed by He et al. [10]. Suppose the complicated function learnt by the stacked layers is $H(x)$, then it is equivalent to hypothesise that the net can asymptotically approximate the residual functions, i.e., $H(x) - x$ (assuming that the input and output are of the same dimensions). So, instead of approximating $H(x)$, we explicitly let these layers approximate a residual function $F(x) := H(x) - x$. The original function thus becomes $F(x) + x$. The authors claim that 1) residual learning makes it easier to optimise a deeper net; and 2) the network can gain accuracy from increased depth. We only use a single residual block for the following two reasons: 1) to avoid overfitting since the training data used in our experiments are limited; 2) to facilitate a fairer comparison by making nnet1 and nnet2 have the same number of layers. Similarly, we use global temporal max- and average-pooling after the residual block. The remaining parts of the network, used as a classifier, have the same settings as the first network.

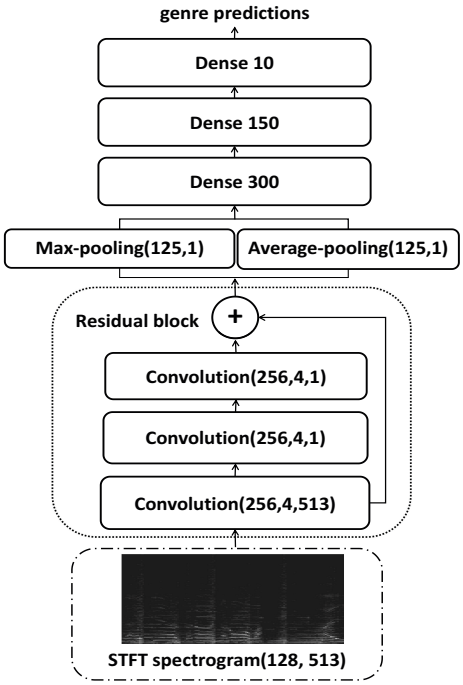


Figure 2: *The network architecture of nnet2.*

We also use a technique called batch normalisation (BN) [17] to speed up the training process and make the final model more robust. The BN operation provides a method of reducing internal covariate shift and thus allows us to use a much larger learning rate. In particular, BN can be expressed as

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \quad (1)$$

$\hat{x}^{(k)}$ is the normalised $k$-th dimension of the input, and the parameters $\gamma^{(k)}$ and $\beta^{(k)}$ are learnt along with the original model parameters. We refer the readers to [17] for the details of the BN technique.

## 4. Experiments and Results

In this Section, we report the experiments used to evaluate the methodologies described in Section 3.

### 4.1. Dataset

The dataset we used is GTZAN dataset, which was collected by Tzanetakis and Cook [18]. The GTZAN dataset has been widely used as a benchmark for music genre classification [19]. There are 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, all the songs are split into 8/1/1 train, validate and test splits. The number of songs for different genres in the train, validate and test sets is balanced. Evaluation on this dataset was carried out in a 10-fold cross validation manner. 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 firstly cut every song excerpt (about 30 seconds) into smaller music clips (3 seconds) with 50% overlap. We found that this improved the classification accuracy. Then, as in [3], we calculate FFTs on frames of length 1024 with an overlap of 50% and use the absolute value of each FFT frame. The output for each frame is a 513 dimensional vector.

When training the networks in all experiments, we used Adadelta [20] as the optimiser with the default learning 1.0. The loss function we chose was categorical cross-entropy. We also used the dropout technique with 0.2 dropout rate to alleviate the overfitting problem. In nnet2, the output of the first convolutional layer has 256 feature maps and each map is a 125 dimensional vector, while each map of the output of the third convolutional layer is a 119 dimensional vector. Before the component wise adding operation, we used zero padding to make sure that the two vectors are of the same dimension. We used mini batches of 50 samples and we shuffled the samples after each epoch.

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 genre classification accuracies of the neural networks described in Section 3 are reported in Table 1. For comparison, we also presented results achieved with neural networks that used only max-pooling or average-pooling. The neural network with only max-pooling is equivalent to the one used in [8]. But in their network, the DNN after CNN contains only one hidden layer. In addition, they only used the neural networks to extract features.

Table 1: *Results for nnet1 and netted with different pooling methods.*

| Methods | Accuracy |
|---|---|
| nnet1(max-pooling) | 79.9% |
| nnet1(average-pooling) | 84.4% |
| **nnet1** (max- and average-pooling) | **84.8%** |
| nnet2(max-pooling) | 85.0% |
| nnet2(average-pooling) | 81.9% |
| **nnet2** (max- and average-pooling | **87.4%** |

Table 2: *Genre classification results on GTZAN.*

| Methods | Features | Accuracy |
|---|---|---|
| **nnet1** | **STFT** | **84.8%** |
| **nnet2** | **STFT** | **87.4%** |
| KCNN(k=5)+SVM [8] | mel-spectrum, SFM, SCF | 83.9% |
| DNN (ReLU+SGD +Dropout) [3] | FFT (aggregation) | 83.0% |
| Multilayer invariant representation [12] | STFT with log representation | 82.0% |

As can be seen, if only a single pooling operation is used, the first neural network works best with max-pooling, while the second one works best with average-pooling. That means we need to choose the right pooling operation for the network to achieve the best performance. However, combining both max- and average-pooling always improve the classification accuracy, especially for the second neural network. In addition, the residual learning method significantly improves the classification accuracy.

In table 2, we compare with previous results on the GTZAN data set. The performance of nnet1 is slightly better than KCNN. Note that nnet1 with only max-pooling is almost the same with KCNN (K stands for $k$-max, [8]). But in [8], the extracted features using KCNN were fed into SVM for classification. The nnet2 result outperforms all listed previous results.

## 5. Conclusions and Future Work

In this paper, we have investigated the effectiveness of using CNNs for music genre classification. Our experimental results show that the following two ways are effective to improve music genre classification with CNNs: 1) combining max- and average-pooling to provide more statistical information to higher level neural networks; 2) using shortcut connections to skip one or more layers, a method inspired by residual learning.

In the future, we'll try to fuse new methods such as multi-scale convolution and pooling [21] with residual learning (e.g. inception-resnet [22]). 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.

## 6. Acknowledgement

This work is supported in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2011A010801005, Grant 2014B010111003, Gant 2014B010111006 and the Fundamental Research Funds for the Central Universities under Grant 2015ZJ009.

# 7. References


[1] N. Auguin, S. Huang, and P. Fung, "Identification of live or studio versions of a song via supervised learning," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, 2013, pp. 1–4.

[2] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and adaboost for music classification," *Machine learning*, vol. 65, no. 2-3, pp. 473–484, 2006.

[3] 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*. IEEE, 2014, pp. 6959–6963.

[4] 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*. IEEE, 2015, pp. 1–6.

[5] 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*. IEEE, 2014, pp. 96–102.

[6] 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, 2013.

[7] 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.

[8] 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 *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 379–386.

[9] 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 *Computer Science and Electronic Engineering Conference (CEEC), 2014 6th*. IEEE, 2014, pp. 35–40.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[11] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[12] 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*. IEEE, 2014, pp. 6984–6988.

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

[14] S. Lippens, J. P. Martens, and T. De Mulder, "A comparison of human and automatic musical genre classification," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 4. IEEE, 2004, pp. iv–233.

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

[16] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, IEEE transactions on*, vol. 10, no. 5, pp. 293–302, 2002.

[19] B. L. Sturm, "An analysis of the gtzan music genre dataset," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*. ACM, 2012, pp. 7–12.

[20] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[21] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*. Pontifícia Universidade Católica do Paraná, 2013, pp. 116–121.

[22] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.