“Multilingual” Deep Neural Network For Music Genre Classification

Jia Dai[1], *Wenju Liu*[1], *Chongjia Ni*[2], *Like Dong*[3], *Hong Yang*[3]

[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, China
[2] School of Mathematic & Quantitative Economics, Shandong Uni. of Finance & Economics, China
[3] Electric Power Research Institute of ShanXi Electric Power Co., China State Grid Corp, China

[1]{jia.dai, lwj}@nlpr.ia.ac.cn, [2]cjni_sd@sdufe.edu.cn, [3]djcasia@foxmail.com

Abstract

Multilingual deep neural network (DNN) has been widely used in low-resource automatic speech recognition (ASR) in order to balance the rich-resource and low-resource speech recognition or to build the low-resource ASR system quickly. Inspired by the idea of using multilingual DNN for ASR, we use a “multilingual” DNN (Multi-DNN) for music genre classification. However, we do not have “multilingual” in music, so we use the similar resource instead. In order to obtain the similar resource corresponding to small target database, the nearest neighbor (NN) algorithm is used to re-label the large similar database. Then the re-labeled large similar database is used to train a Multi-DNN, and the small target database is used to further adapt the trained Multi-DNN. By using the Multi-DNN approach, the DNN can be well trained, and be transferred to the small target database quickly. The experiments are evaluated on the benchmark databases, ISMIR database and GTZAN database, which are used as the large similar database and small target database respectively. The experiment results show that the proposed method can achieve 93.4% (10-fold cross-validation) average classification accuracy on GTZAN database, which outperforms the state-of-the-art best performance on this database.

Index Terms: multilingual, music genre classification, DNN

1. Introduction

In the information age, millions of music tracks are available on the Internet and it will be helpful if the large music collections are properly categorized. Automatic classification of music tracks is of great practical importance for information retrieval, music recommendation and on-line music access. The music genre is one of the most popular descriptions of the music content, and automatic music genre classification aims to correctly categorize an unknown recording to a specific music genre. It is useful especially in some personal applications such as automatic play-list generation [1].

Music genre classification mainly contains two steps: feature representation and classifier design. Several works have been done for classifying the music genre. Some of them focus on the representation of feature [2][3], and these methods generally aim to make the feature more discriminative. Others focus on the classification methods [4], which exploit the advantages of different classifiers and try to find a powerful classifier.

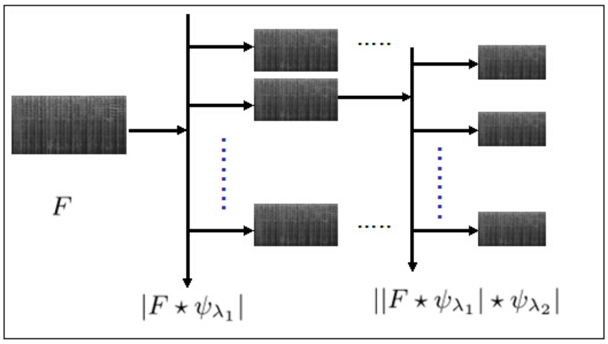
Although these methods have achieved a success in improving the whole music genre classification performance, there still exists some problems. One problem is how to build a classification system quickly. In practice, as the data updates very rapidly, much time may be required to train or update a system. If we can quickly build a new system using those old systems, we can save a lot of time. Another problem is that the lack of training data degrades the classification performance on some genres. It has been known that when using deep neural network (DNN) for classification, better performance can be achieved if we increase the number of hidden layers or hidden nodes in each layer within a certain range [5]. However, usually, large amount of data is needed to train a large DNN, while only a small amount of target data can be acquired in practice. There hasn’t been a method which is aimed at improving the performance of those low classification accuracy genres. In ASR, we can quickly build a recognition system on little data while having resource from other languages using multilingual neural network training or transfer learning [6][7][8], and the acoustic model trained on other languages can improve the performance of the target language. In [9], English corpus with Chinese accent is used to improve the German ASR on German with Chinese accent. Multilingual learning or transfer learning has been proved to be very useful in ASR and some other areas [10][11][12][13].

In ASR, a language with little training data can benefit largely from multilingual training on other languages. Inspired by this, we propose a “multilingual” DNN (Multi-DNN) model to improve the performance of music genre classification. However, we don’t have “other language” like in the ASR task. So, “the most similar resource” is used instead in music. When modeling, if we have a large similar database which has the same classes as target small database, we use it as “the most similar resource”, and then we quickly build a new system on it. Otherwise, we utilize the nearest neighbor (NN) algorithm to find “the most similar resource” data from a large similar database. Then “the most similar resource” is adopted for Multi-DNN training. At last, we use small target data for fine-tuning the Multi-DNN to get the final classification system. Using this system, the classification performances on genres which have poor classification accuracy are improved, and then contribute to the whole classification accuracy. A music genre which is not contained in large similar database, but using “the most similar resource” as Multi-DNN training improved the classification accuracy of this music genre.

To the best of our knowledge, this paper contributes the first multilingual DNN for the music genre classification, and it outperform state-of-the-art methods on GTZAN database[14].

2. Scattering Transform

The Mel-Frequency Cepstral Coefficient (MFCC) is widely used in many systems such as ASR, audio classification and so on. It is obtained by averaging spectrogram values over Mel-



F
$|F \star \psi_{\lambda_1}|$
$||F \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|$

Figure 1: The first order and second order scattering transform

frequency bands using Mel-scale filters $\psi_\lambda(\omega)$. The following is the Mel-frequency spectrogram:

$$M(\alpha, t, \lambda) = \frac{1}{2\pi} \int |\alpha(t,\omega)|^2 |\psi_\lambda(\omega)|^2 d\omega \qquad (1)$$

where, $|\alpha(t,\omega)|$ is the Fourier transform of α, α is a sample vector of a audio file, and λ is the center frequency of each $\psi_\lambda(\omega)$. It is effective over the length of a short time window 25 ms, and when using a window larger than 25 ms, the information lose becomes too important [15]. For music whose long time interval representation helps more for classification, the MFCC have a poor classification performance. Scattering transform is an extension of MFCC. It has been proved successful for music genres classification [15] [16]. It builds invariant, stable and informative signal representations and is stable to deformations. It is computed by scattering the signal information along multiple paths, with a cascade of wavelet modulus operators implemented in a deep convolution network (CNN). In the Figure 1 the first order and second order of the scattering transform are shown, which is much like the structure of CNN. And it can be expanded to the third order or more.

The first order scattering coefficients are first order moments of wavelet coefficient amplitudes, it can be computed as:

$$SC(\lambda_1) = |F \star \psi_{\lambda_1}| \qquad (2)$$

Second order scattering moments recover information lost by averaging spectrogram through computing the wavelet coefficients of each $|F \star \psi_{\lambda_1}|$, and their first order moment:

$$SC(\lambda_1, \lambda_2) = ||F \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \qquad (3)$$

where, $\psi_{\lambda_1}, \psi_{\lambda_2}$ are band-pass filters, F is a Fourier spectrogram, and $|F \star \psi_\lambda|$ is nearly equal to time averaging. For more detail about scattering transform, you can refer to [15].

3. Multi-DNN System

3.1. DNN

DNN is the latest hot topic in many fields [17] [18] [19]. In this paper, the architecture of DNN is a typical feed-forward neural network. It has one input layer, some hidden layers and one output layer. In our DNN, each hidden layer has 1024 neurons. The activation function of hidden layers is sigmoid function and the activation function of the output layer is soft-max function. The cross-entropy function is used as the objective function $J(W, b)$:

$$J(W, b) = \frac{1}{2} Y^T log f(W^T X + b) \qquad (4)$$

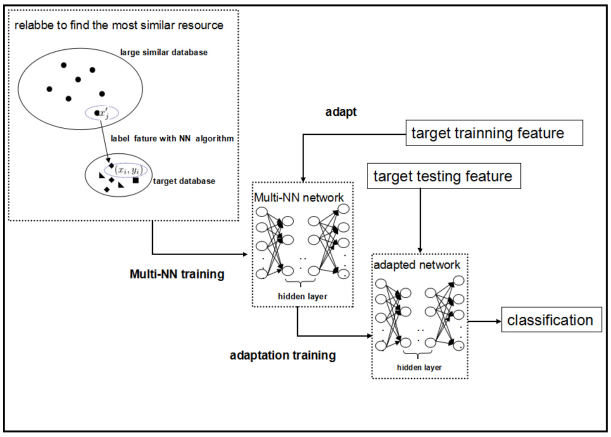


Figure 2: The architecture of Multi-DNN system for music genre classification

where (X, Y) is input sample feature, $f(\cdot)$ is the activation function, W is the weight matrix, and b is the bias vector.

3.2. Multi-DNN System

In this paper, we propose a Multi-DNN system for music genre classification. It is the cascade of DNN. Figure 2 is our Multi-DNN system structure for music genre classification. It contains three main stages.

In the first stage, we get "the most similar resource". If we already have a large similar database which consists of the same classes as the small target database, we use it as "the most similar resource" and then quickly build new system on it. We directly jump to the second stage, and skip the following of the first stage. Otherwise, if we have a large similar database which classes are different from the small target database, we relabel the large similar resource feature in order to find "the most similar resource". We don't use the original similar labels and relabel it, because those classes may be divided differently from target database, and some classes that we desire very much are not contained in large similar database. And sometimes in actual we cannot find the labeled database. So we re-label it for finding "the most similar resource" corresponding to each genre of small target database. We use NN algorithm to relabel the similar feature. First, we randomly select 10% of small target training feature, denoted by $\{(x_i, y_i) | i = 1, 2, ..., n\}$. x_i is the feature vector selected from target database and y_i is the genre label of feature vector x_i. Just as in the Figure 2. For feature vectors $\{x_j' | j = 1, 2, ..., m\}$ in the large similar database, we find the nearest target vector to x_j' using NN algorithm:

$$k_j = arg \min_i ||x_j' - x_i||, i = 1, 2, ..., n \qquad (5)$$

Where, k_j is the index of target feature vector nearest to x_j', $j = 1, 2, ..., m$, n is the number of feature vector in selected target database and m is number of sample vector in large similar database. The new genre label of x_j' is denoted by y_j':

$$y_j' = y_{k_j} \qquad (6)$$

Where, y_{k_j} is the genre label of x_{k_j}. Then (x_j', y_j') is a "the most similar resource" of the genre y_{k_j}.

In the second stage, we use "the most similar resource" feature to train a Multi-DNN. we use DNN with a random initialization. The structure of the DNN used here is described in section 3.1. Then the frame-level feature is adopted as the input of the DNN for Multi-DNN training. After Multi-DNN training, we get the trained Multi-DNN. This stage can be seen as a pre-train of the target DNN. Using a large similar database as a pre-train can make up for the deficiency that the small target data leads to the poor performance.

In the third stage, we use small target training feature to adapt the trained Multi-DNN. We use the trained Multi-DNN as the initial neural network of a new DNN, then use frame-level target training feature to train the new DNN. The old DNN and new DNN share the same network layers, which exploits the idea of multilingual neural network. The new DNN parameters are same as the old DNN. After we get the adapted Multi-DNN (trained new DNN), we get our Multi-DNN system. Then we use target testing feature as input for testing the performance of the system.

4. Music Genre Classification

4.1. Database and Experiment Setup

Two databases are used in the experiments, the GTZAN database [14] and ISMIR 2014 Genre database [20]. These two databases have been used as benchmarks for music genre classification by many researchers [21][22][23][24]. The ISMIR database is used as large similar resource database for Multi-DNN training. It contains 6 music genres. The time lengths of different tracks are different, and the total time length is about 100 hours. In our work, all ISMIR music tracks are used for Multi-DNN training without the original music genre labels. Before feature extraction, each audio file in ISMIR has been converted into a 22050Hz, 16 bit, and single channel WAV file. The GTZAN database is used as target database. It is divided into 10 music genres. The audio file format is 22050Hz, 16-bit, single channel WAV. The detail of the two databases are described as Table 1.

Table 1. *Database Description*

GTZAN		ISMIR	
genre	tracks	genre	tracks(train/test)
classical	100	Classical	320/320
jazz	100	Electronic	115/114
blues	100	Jazz/Blue	26/26
metal	100	Metal/Punk	45/45
pop	100	Rock/Pop	101/102
rock	100	World	122/122
country	100		
disco	100		
hiphop	100		
reggae	100		
total	1000	total	729/729

We first exact the feature on GTZAN and ISMIR using scattering transform as in section 2. The scattering transform can be computed by ScatNet [25]. In our work, we calculate first-order and second-order time scattering coefficients using a window of 370ms with half overlap. As having been proved to be useful for music genre classification, the octave bandwidth Q1=1 and Q1=8 [15] are both used. After that, we stack the feature. In other words, we use a window of 3 frames feature length and 1 frame shift, and then pull the 3 frames of feature within a window into a vector. We do this to make the feature consist context information. Finally we get our scattering feature, and our following work is based on this.

After preparing the feature, a randomized 10-fold cross-validation repeated 10 times. The GTZAN database is randomly divided into 10 folds, 9 folds of which are used for training and the remaining one is used to test the classification accuracy. For training and testing, we used frame-level feature as direct input. In real life, usually, we are interested in determining the genre of the whole music track (a clip of music), rather than the genres of the internal frames. So after training and testing, the majority vote is used to get the genre labels of all frames in a music clip. In this way, the genre label of the whole music clip is decided by the majority of the genre labels on the internal frames. The classification accuracy is evaluated 10 times on the 10 different combinations of training/testing sets. The overall classification accuracy is calculated as the average of 10 independent 10-fold cross-validations.

4.2. Baseline Systems

Support Vector Machine (SVM)[26] is a discriminative model with great generalization and more discriminative ability. It has been proved to be useful for classification in many fields. As one of the baseline models, the SVM is used. In this experiment we use SVM with radial basis function (RBF), and the best choice of c (the cost in RBF), g (the gama in RBF) are obtained by grid search algorithm on the GTZAN training feature. The result of SVM based baseline (baseline-SVM) is reported in the Table 2. Another baseline model used here is DNN. The DNN we used here is Karel's DNN implementation in kaldi [27] with random initialization. The structure of DNN is in section 3.1. No dropout function is used and learning rate is $8 * 10^{-6}$. The training step we use here is 1000. The result of DNN based baseline (baseline-DNN) is also reported in the Table 2.

The previous approaches for music genre classification on GTZAN database have made some successes. A robust music genre classification framework is proposed with locality preserving non-negative tensor factorization yields a 7.6% error rate [22]; deep scattering spectrum achieves an error of 8.1% with cascade of wavelet [15]; an automatic music genre classification approach based on long-term modulation spectral analysis of spectral as well as MFCC has got an error of 9.4% [28]. To the best of our knowledge, those are the best three results for music genre classification on the GTZAN database.

Table 2. *Classification result of different models*

model(hidden layers)	average accuracy
Panagakis and Kotropoulos [22]	92.4%
Andén and Mallat [15]	91.9%
Lee, Shih and Yu [28]	90.6%
baseline-SVM	88.5%
baseline-DNN(1024)	89.5%
baseline-DNN(1024-1024)	89.7%
baseline-DNN(1024-1024-1024)	90.1%
Multi-DNN(1024)	92.7%
Multi-DNN(1024-1024)	93%
Multi-DNN(1024-1024-1024)	**93.4%**

4.3. Multi-DNN System

In this experiment, The GTZAN database is used as small target database, and ISMIR database is used as large similar resource database. ISMIR database contains about 100 hours music genre data belongs to 6 classes, which is enough to build a bigger DNN network. As we know from the Table 1, some genres (e.g. "reggae") we desire very much is not contained

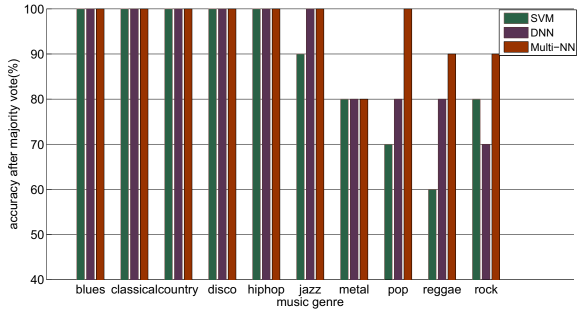


Figure 3: The detail classification performance of ten genres using different models. The result is one of ten-fold cross-validation, and it is obtained after majority vote. The hidden layers structure of DNN and Multi-DNN is 1024-1024-1024.

in ISMIR database. By re-labeling it using NN algorithm, we find "the most similar resource" to corresponding much desired GTZAN genres. Then use all "the most similar resource" for Multi-DNN training. The DNN and DNN parameters we used here for Multi-DNN training and adapting is just as the same as baseline-DNN.

As the number of hidden layers and the number of hidden units in each hidden layer effect the classification performance, we use DNN with different structures. The compared performance between Multi-DNN model and baseline models or the previous approaches can be seen in Table 1. From Table 1 we can see that our method gets an accuracy rate of 93.4%, which has 3.3% improvement compared to the baseline-DNN, and also outperforms the state-of-art best performance 92.4%.

4.4. Experiment Analysis

In this subsection, we analyze in detail why our Multi-DNN model improves the performance and how to build a new system quickly.

The first is to improve the performance. We improve the whole performance by improving the poor classification accuracy genre using "the most similar resource" for pre-training. Figure 3 is the detailed classification performance of 10 genres using different models. We can see from the Figure 3 that the accuracies of some genres are very low which affect the whole classification accuracy. Even we increase the size of the DNN, the performance cannot be enhanced. We use 3 hidden layers with 1024 nodes in each hidden layer, and the accuracy on those genres stays very low ("pop" or "reggae" in Figure 3). This is because those genres are easily to be misclassified to other genres. For these genres, if we want to improve the performance, we need more training data. But we may not so easily get the same genre data in real life. By using the idea of Multi-DNN, we use "the most similar resource" instead of original genre data. For example, "reggae" in GTZAN has a poor classification accuracy. Although the ISMIR database doesn't contain "reggae", we use nearest neighbor algorithm in the first stage to get the data more similar to "reggae". Then the data more similar to "reggae" will train as "reggae" for Multi-DNN training. From the result we know that the classification accuracy of "reggae" is improved and then contributes to the overall performance. This is similar to multilingual training in ASR that language training benefits from similar language. With this result, we can generalize this theory to other models. If we have a small database which limits the performance, we can use a similar bigger database to train a model. Then adapt this model using original small database.

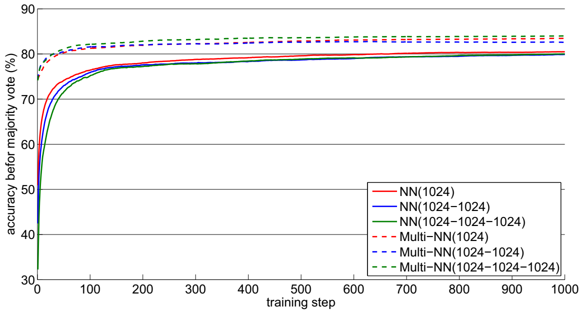


Figure 4: The performance between different training steps and different models. The result is one of ten-fold cross-validation, and it is obtained before majority vote in each music clip.

The second is for quickly building a classification system if we already have an old classification system. Then we can build a new system by adapting the new data to the original system using very little time. Figure 4 shows the performance between different training steps and different models. As shown in Figure 4, We can see that, after Multi-DNN training, our model can quickly achieve very good and stable performance through a few training steps. We use about 1000 training steps to get a low performance in baseline-DNN model training, but can use only about 100 steps to get a better performance in Multi-DNN adapting.

5. Conclusion

In this paper, we propose to use the Multi-DNN to improve the performance of music genres classification. This approach can utilize the large similar database to improve the classification performance of small target database. In order to overcome the dis-consistent label between large similar database and the small target database, we propose to use the NN algorithm approach to obtain the similar label for large similar database. The Multi-DNN can be pre-trained well by using "the most similar resource". After fine-tuning with the target training data, the Multi-DNN can obtain the better performance. It can get 93.4% classification accuracy rate, which is better than state-of-the-art top three methods.

This model also makes use advantage of DNN, which is easy to handle and train huge data. As many products are based on the vast amounts of on-line data, it can be generalized to many other systems in real life in many applications. For those music genres or similar task if we have some classes which have a relative low performance, we can use this model to train DNN on similar rich resource to improve the accuracy. Or, we can use this model to quickly build a new system if we already have an old system. It will save a lot of time and human resource. But there is also a disadvantage in Multi-DNN system: that is if we want to quickly build a new system based on an old one, the classes of the old system data must the same as new system. This may poses some limits on promoting the system more widely. In the future, we will devote to overcome this disadvantage.

6. Acknowledgements

This research was supported in part by the China National Nature Science Foundation No.91120303, No.61273267, No.61403370, No.90820011 and No. 61305027.

7. References

[1] Y. Song and C. Zhang, "Content-based information fusion for semi-supervised music genre classification," *Multimedia, IEEE Transactions on*, vol. 10, no. 1, pp. 145–152, 2008.

[2] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen, "Temporal feature integration for music genre classification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1654–1664, 2007.

[3] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, "Music genre classification based on ensemble of signals produced by source separation methods," *Intelligent Decision Technologies*, vol. 4, no. 3, pp. 229–237, 2010.

[4] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 282–289.

[5] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Interspeech*, 2011, pp. 437–440.

[6] F. Grézl and M. Karafiát, "Combination of multilingual and semi-supervised training for under-resourced languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, vol. 2014, no. 9. International Speech Communication Association, 2014, pp. 820–824.

[7] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[8] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3208–3215.

[9] N. T. Vu, Y. Wang, M. Klose, Z. Mihaylova, and T. Schultz, "Improving asr performance on non-native speech using multilingual and crosslingual information," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[10] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Language id-based training of multilingual stacked bottleneck features," 2014.

[11] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end." in *INTERSPEECH*, 2008, pp. 2711–2714.

[12] N. T. Vu, J. Weiner, and T. Schultz, "Investigating the learning effect of multilingual bottle-neck features for asr," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[13] F. Grezl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 359–364.

[14] 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.

[15] J. Andén and S. Mallat, "Deep scattering spectrum," *CoRR*, vol. abs/1304.6763, 2013.

[16] J. Anden and S. Mallat, "Multiscale scattering for audio classification," in *ISMIR*, 2011, pp. 657–662.

[17] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[18] J. Li, C. Niu, and M. Fan, "Multi-scale convolutional neural networks for natural scene license plate detection," in *Advances in Neural Networks–ISNN 2012*. Springer, 2012, pp. 110–119.

[19] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model." in *SLT*, 2012, pp. 234–239.

[20] Ismir 2004 genre dataset. [Online]. Available: http://ismir2004.ismir.net/genre_contest/index.html

[21] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *Multimedia, IEEE Transactions on*, vol. 8, no. 3, pp. 564–574, 2006.

[22] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music genre classification using locality preserving non-negative tensor factorization and sparse representations." in *ISMIR*, 2009, pp. 249–254.

[23] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *ISMIR*, 2005, pp. 34–41.

[24] I. Panagakis, E. Benetos, and C. Kotropoulos, "Music genre classification: A multilinear approach," in *ISMIR*, 2008, pp. 583–588.

[25] scattering transform toolbox in matlab. [Online]. Available: http://www.di.ens.fr/data/scattering/

[26] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[27] Karel's dnn implementation in kaldi. [Online]. Available: http://kaldi.sourceforge.net/dnn.html

[28] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *Multimedia, IEEE Transactions on*, vol. 11, no. 4, pp. 670–682, 2009.