Automatic music genre classification using deep learning technologies

Chao Xiong
KTH, Royal Institute of Technology
cxiong@kth.se

Yu Hu
KTH, Royal Institute of Technology
hu3@kth.se

Yuxia Wang
KTH, Royal Institute of Technology
yuxia@kth.se

Wenqi Rong
KTH, Royal Institute of Technology
wenqir@kth.se

Abstract

Machine learning and deep learning technologies have contributed greatly to solving problems in various fields, in which music genre classification is a challenging and attracting one. This project aims to apply different deep neural networks (DNN) to the task of automatic music genre classification, by means of several music features. Based on Mel-Frequency Cepstral Coefficients (MFCC), delta MFCC and delta-delta MFCC features extracted from the music clips, we build models based on deep learning networks, including Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). Considering the relatively small dataset and the large feature dimension, data augmentation method is also implemented to alleviate the influences of overfitting. Experimental results show that our CNN and LSTM both perform well for this task, hitting the classification accuracy at 61.5% and 68% respectively. Furthermore, the extended MFCC features improve the accuracy again up to about 72%.

1 Introduction

Content-based musical information processing and retrieval [1] has received much attention for the past decades, in which music genre classification is one of the main research topics. A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions [2], including instrumental and vocal tones. In applications, music genre classification can be used in music labelling, similar music recommendation based on music genre or help information. With the help of music recognition function built in the applications, users can retrieve and match the music from the song library within tens of seconds.

However, music classification and annotation by person can be time-consuming and expensive. With the increasing popularity of machine learning technologies, more and more research work have been done based on machine learning algorithms, so does the music genre classification. Termens [3] gives an exhaustive study on automatic music genre recognition in his doctoral thesis, including analyzing multiple feature descriptors and various machine learning models.

At the same time, it is very important to select and extract proper features which can represent the audio signals for the task of automatic recognition or classification. G. Tzanetakis [4] proposed a method for automatic music genres classification based on timbral texture, rhythmic content and pitch content of music signals. H. Zhou et al. [5] examined the spectral features MFCC for speech and music discrimination, in which improved MFCC features are used based on dynamic changes of the MFCC information.

The recent deep learning technologies have the advantages of visual presentation of image information, for example, CNNs do a great job on image classification. H. Bahuleyan explored the music genre classification based on CNN network [6], in which he also proved that among the multiple features, MFCCs contribute significantly to the task of music genre classification. Since MFCC features can be seen as spectral images, it is possible to do the music classification based on CNN networks. At the same time, LSTM model, as a subclass of Recurrent Neural Network (RNN), is also a dark horse in deep neural network field. Different form the traditional neural networks, LSTM can memorize the past data and predict with the help of the memorized information. Tang C P et al. [7] trained an LSTM model to classify music genres with divide-and-conquer approach and hit the average accuracy at 52.975% on 10 genres, proving LSTM is also a competitive contestant in the game of music genre classification.

The main tasks of this project are (1) to explore and evaluate the music genre classification performance on different deep neural networks including MLP, CNN and LSTM, (2) besides the mostly-used MFCC, to introduce new music features namely delta and delta-delta MFCC to the above mentioned networks and analyze if they could help in music genre classification task, and (3) to examine if data augmentation can improve the network performance in our case.

This project report is organized as follows: Section 2 presents the dataset and music features expression, the data augmentation method as well as the design of our deep neural networks; Section 3 introduces the data preparation and network model settings in detail; Section 4 describes the the results achieved in the two-stage experiments before and after the data augmentation; Finally discussion and conclusions are stated in the last section.

2 Method

Music genre classification based on deep learning technologies involves two parts of work: (1) Feature selection and extraction, in which the selected features should represent the musical features properly; (2) Network selection and design, where the network's characteristic and the detailed architecture should be considered.

2.1 Dataset and feature selection

The public GTZAN dataset from Kaggle [8] is chosen for experiments in this project. There are 10 music genres in this dataset, each of which is composed of 100 music clips. They are all 22050 Hz monophonic 16-bit audio files in .au format, with the length of 30 seconds. The genres are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock.

For feature selection, besides the most commonly used MFCC features, we also take into account other features including delta MFCC and delta-delta MFCC, which are potentially helpful for music classification.

2.1.1 MFCC features

According to the results of previous research [4, 6], the spectral MFCC coefficients are the main features for speech or audio signals processing, which represent the audio information better than the features in time domain, so we extract MFCC features to do this work. Table 1 gives the parameters when extracting MFCCs from the audio clips. Considering both the expressive capability and the computation complexity, the number of the MFCC coefficients is set to 20. Figure 1 and Figure 2 show the MFCC features for pop and reggae music. It is clear that the MFCC feature graphs are different for these two music genres.

Table 1: Parameters setting for MFCCs

Sampling rate	22050Hz
Window size (nfft)	2048
Hop length	512
Num of MFCCs	20

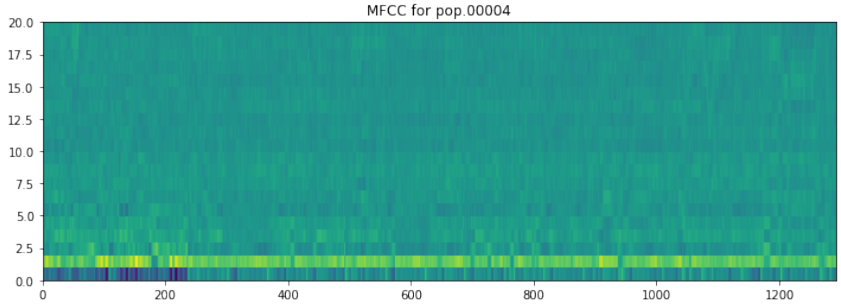


Figure 1: MFCC for pop music.

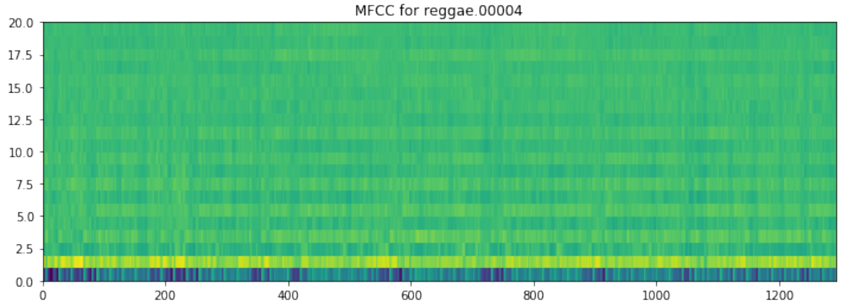


Figure 2: MFCC for reggae music.

2.1.2 Delta MFCC and delta-delta MFCC features

MFCC describes the instantaneous, spectral envelope shape of the audio signal. However, audio signals are time-variant signals and in a constant flux, MFCC may not represent enough information on this. So as done in phoneme recognition or voice activity detection [9], we compute the first and second order derivative of MFCC values, that is delta MFCC and delta-delta MFCC. These features are combined with MFCC as the new features, which can introduce more non-instantaneous features. Figure 3 shows the two features for the same clip of pop.00004.

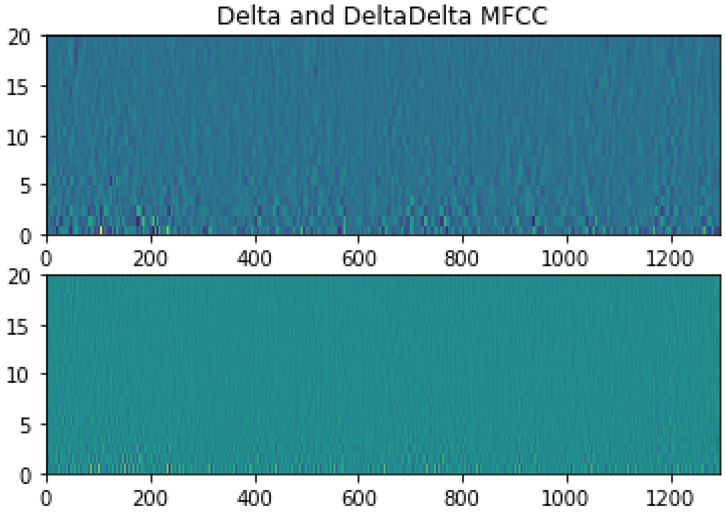


Figure 3: Delta MFCC and delta-delta MFCC for pop.00004.

2.2 Data augmentation

Training a neural network with a small dataset like GTZAN may cause the network memorize training samples, which leads to overfitting. Data augmentation is a popular technique to reduce overfitting of training model by increasing the amount of data.

In our experiment, each original song is split into multiple clips, the duration of each clip is 3 seconds. The motivation is that people can usually perceive the genre of a song within a few seconds [4](This method also has been adopted in [13]).

2.3 Network models and architectures

This project is inspired by the previous project, which classified the music genres using multi layer perceptron (MLP) network [10], but the classification performance of the model is not satisfying. Here more deep neural networks like CNN and LSTM are explored. As a baseline comparison, the 4-layer MLP is also implemented.

2.3.1 CNN

As a class of deep neural networks, CNN is most commonly applied to visual analyzing. Since MFCC can be expressed in the form of images, it is reasonable to hypothesize that CNN can better perceive the hidden regularity while adopting the MFCC frequency domain image as network input. The convolution layer in CNN can reduce the number of free parameters, allowing the network to be deep [11] while pooling layer can reduce the dimensions of the data.

VGG-16 [12] was firstly selected as reference since it is a widely-used network. However, according to our experiments on original VGG-16, the architecture was found oversized for our project, leading to long training time and little improvement on performance. Thus a three-convolution-layer CNN, as shown in Figure 4, is chosen instead of the original one. In Figure 4, the input shape is (129,20,1), which is the shape of resized data we used in later experiment. The output is the number of music genres.

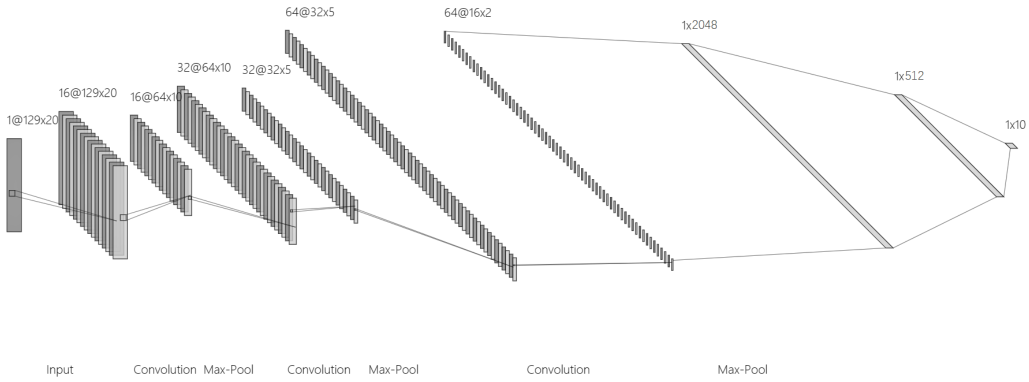


Figure 4: CNN architecture in this work.

2.3.2 LSTM

LSTM is also frequently used in the field of deep learning for sequential data including videos and speeches. It is worth trying to implement LSTM to see if long-term memory helps on this genre classification work. The LSTM architecture in this work is shown in Figure 5. The input should have either 20 or 60 dimensions depending on what kind of features are chosen and the sequence length is 129 for split data or 1290 for the original one.

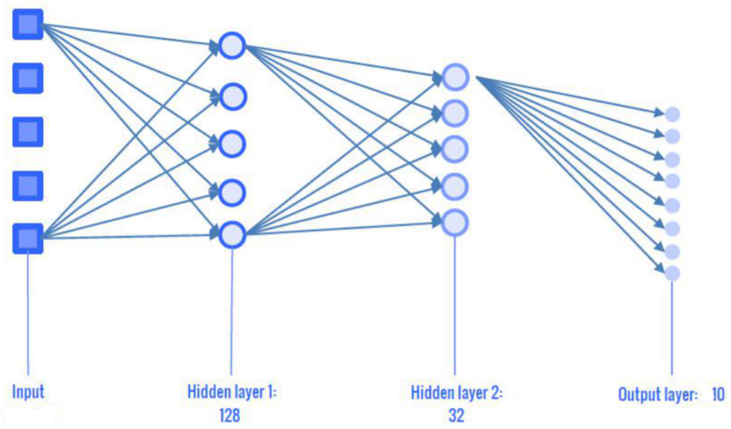


Figure 5: LSTM architecture in this work.

3 Experiments

In this work, firstly, the DNN models are built based on the MFCC and the extended features with delta information. Secondly, considering the potential issues caused by the relatively small size of our dataset, a bigger dataset from the resized data are used.

3.1 Model building

As a baseline model, MLP is implemented using the same settings as in [10]. That is, there are 4 hidden layers, 256 nodes in each layer and a softmax layer on the top of hidden layers, which calculates the cross-entropy between the predictions and an one-hot encoding of the ten classifications. However, to compare with other DNN models, the dimension of features are different with the reference, we compute 20 MFCC coeffients for the whole time step of each music clip. The MFCC dataset is 1000*25800 and the dataset from MFCC with delta is 1000*77400.

For CNN, we used 3 convolution2D layers, the activation function applied is ReLU (Rectified Linear Unit) for a non-linear operation. Then the pooling layer is added to reduce dimensionality and the chosen pooling type is MaxPooling. For each convolutional block including the activation and the MaxPooling layer, we added dropout layer to prevent overfitting, the dropout rate is 0.25 for each convolutional layer. L2 regularization is also added to penalize excessively high weights, with lambda set as 0.0001. The optimizer is Adam algorithm.

For LSTM, as shown in Figure 5, there are two LSTM layers, whose cell sizes are 128 and 32, and the last layer is a full connected layer with softmax activation. The optimizer is Adam as well, by using L2 regularization.

3.2 Data preparation

3.2.1 Original Data

From the GTZAN dataset, by using the library of Librosa, 1000 MFCC features for 1000 music clips can be extracted. Each feature has the dimension 20*1290 as shown in Figure 1, in which 20 is the number of MFCC coefficients, and 1290 is the number of frames for each clip.

For the delta MFCC and delta-delta MFCC features, they have the same dimension with the original MFCC features. For model training, these features are concatenated with MFCC vectors, and the final dimension of each feature image is 60*1290.

For both cases, the feature data are split into three parts: training set, validation set and test set. The split setting is shown in Table 2. Then by using the library of StandardScaler from sklearn, the dataset is standardized to zero mean and unit variance.

Table 2: Dataset splitting for experiment

Training set	Validation set	Test set
60%	20%	20%

3.2.2 Resized data

Similar to the enframing process in MFCC extraction, we spilt original data (songs of 30 seconds) into clips of 3 seconds with 50% overlapping. In total, the size of data set has been increased by 18 times. Now, one sample has 129*20 dimensions, which accelerates batch training a lot. But simultaneously it also brings inconvenience that the input of the model turns to a 3s clip while our task is to classify a 30s song. To fix this issue, we define a custom predict method that we split a song into 18 clips first like the process above and after feeding 18 corresponding features to a model, we take the mode of 18 outputs of these clips as the final prediction of a song.

4 Results

To evaluate the performance of the models, the training and validation accuracy and the prediction accuracy on test data are calculated, and further the classification results are analyzed with confusion matrix.

4.1 Results based on original dataset

Table 3 gives the results for different models with two kinds of features. We can see that for the MLP network, the validation accuracy and test accuracy are no more than 50% for ten music genres. And with delta information added to MFCC, there is no much improvement for the accuracy.

Table 3: The performance for different models and features

Model	Feature	Training acc	Val acc	Test acc
MLP	MFCC	0.9083	0.535	0.475
MLP	MFCC+Delta	1.0	0.47	0.49
CNN	MFCC	0.9683	0.555	0.515
CNN	MFCC+Delta	0.9767	0.565	0.525
LSTM	MFCC	1.0000	0.545	0.500
LSTM	MFCC+Delta	1.0000	0.500	0.560

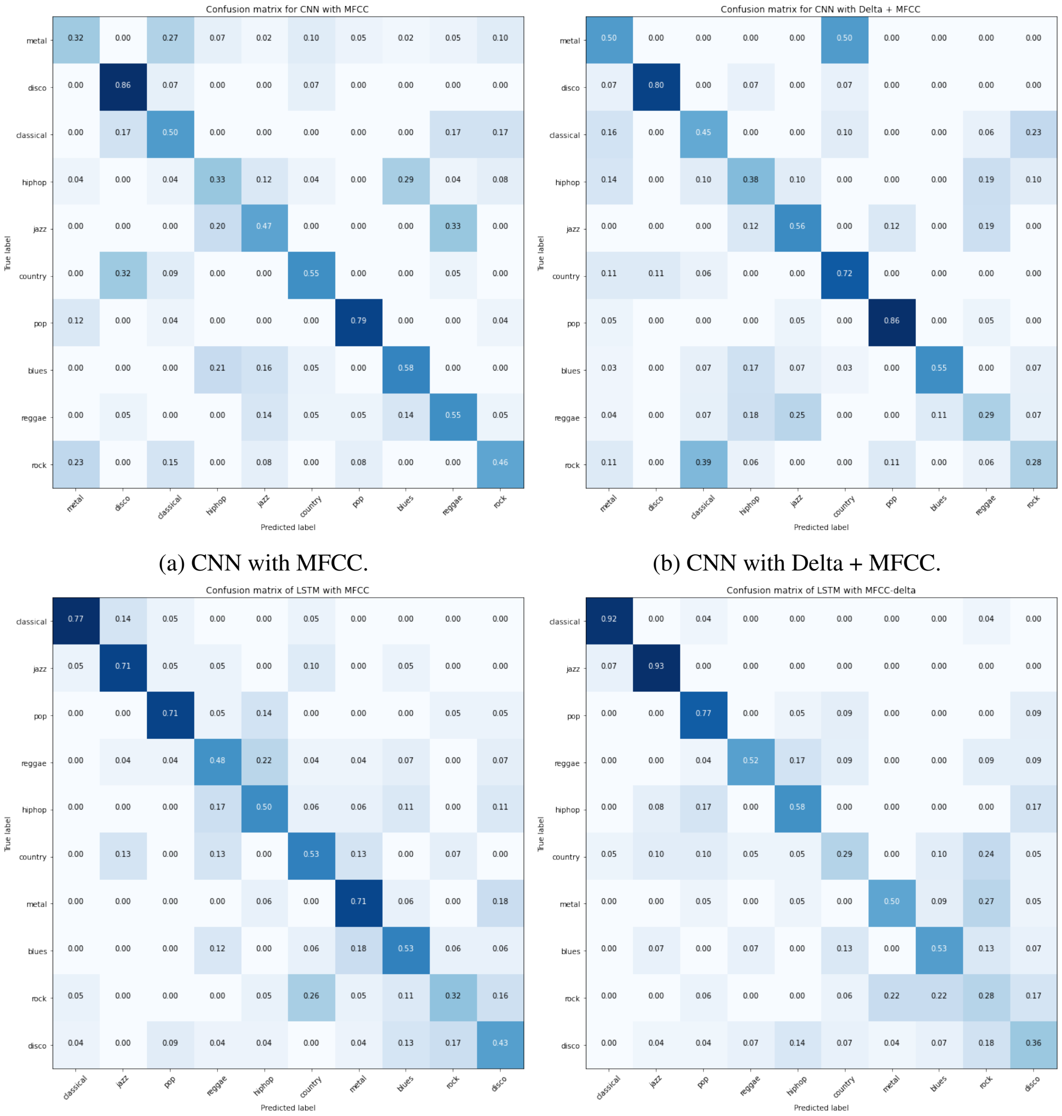
The potential reason that both networks are not robust enough is that the feature space is large, while the training dataset is small (600 samples), especially when MFCC features are extended by two kinds of delta features, the feature space is 3 times of the original size, but the training set is still 600, which may lead to overfitting in this case.

On the other hand, the extended MFCC with delta information contributes only a little to the classification performance, which may be due to the same reason of small training set.

4.2 Results based on data augmentation

Table 4 gives the new results based on data augmentation.The reason why CNN have lower accuracy on training set than the one on test is that the network has many drop-out layers which are only activated in the training process.As shown in the models above, LSTM performs better than CNN when MFCC features are used only. With models built on extended MFCC features, both networks achieve a test accuracy at about 72%, which means they perform similarly on this classification task. It also shows that there is an evident improvement on the test accuracy for all situations compared to the original data. With only MFCC features, the accuracy increases by about 18% and with extended features, the test accuracy increases by about 16-20% for two networks.

Looking into the confusion matrices in Figure 7, over 70% music clips are classified well, in which some genres can get an accuracy over 90%, even to 100%. For blues music, the classification



(a) CNN with MFCC.

(b) CNN with Delta + MFCC.

(c) LSTM with MFCC.

(d) LSTM with Delta + MFCC.

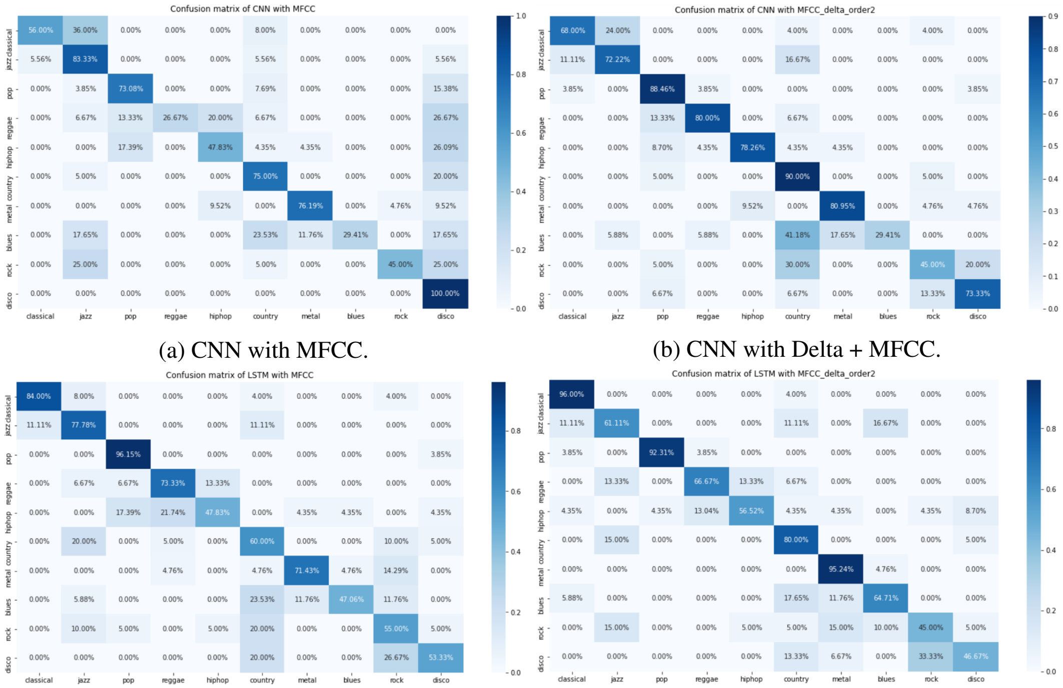
Figure 6: Confusion matrix for CNN and LSTM with two kinds of features

performance of CNN is not satisfying, with the accuracy lower than 30%. LSTM performs better in this case, with the accuracy over 45% for all genres.

The extended MFCC features contribute more to the classification performance based on this dataset, which is consistent with our expectation. The accuracy increases by about 10% for CNN and 4.5% for LSTM networks. The evident improvement is on reggae music for CNN network, the accuracy increases from 26.67% to 88.46% based on extended MFCC information, together with improvements on other genres.

Table 4: The performance for different models and features on split data

Model	Feature	Training acc	Val acc	Test acc
LSTM	MFCC	0.7748	0.685	0.68
LSTM	MFCC+Delta	0.6400	0.635	0.725
CNN	MFCC	0.5468	0.61	0.615
CNN	MFCC+Delta	0.5876	0.635	0.715



(a) CNN with MFCC.

(b) CNN with Delta + MFCC.

(c) LSTM with MFCC.

(d) LSTM with Delta + MFCC.

Figure 7: Confusion matrix for CNN with two kinds of features

5 Discussion and Conclusions

In this project, we aim to explore and propose the methods for automatic music genre classification. Then CNN and LSTM models are selected and trained on both MFCC and extended MFCC features.

The results show that, for ten genres of music, all the models in our experiment perform better than the reference report [10]. For CNN and LSTM although these two models give better performance than the baseline MLP model, the classification accuracy is still not satisfying. One potential reason is overfitting issues due to the small training data set with the large feature dimensions.

Further study is done based on the technology of data augmentation, which can enlarge the dataset by splitting the music clips. Experimental results prove that data augmentation improves the classification performance evidently. As we expected, the extended MFCC features with delta information from this resized dataset also contribute fairly to the final classification performance. On the other hand, LSTM gives higher test accuracy than CNN based on the MFCC features only, but they have similar test accuracy based on the extended MFCC features, which explains that both CNN and LSTM do a good job on music genre classification given enough feature representation.

In the future work, more features can be examined for this task. For instance, as suggested in course lab, in deep learning, another way to represent the time-variant signals is stacking several consecutive feature vectors together. In implementation, 7 MFCC features can be stacked together at each time step. Considering the current relatively small size of data, it is not good to extend the dimension of the feature space again. But this study can be done on new larger datasets. Another work we can do is to evaluate our method in a larger dataset, which can prevent the potential overfitting issues due to a small training dataset.

References

[1] Rainer Typke, Frans Wiering, Remco C. Veltkamp. (2005) A SURVEY OF MUSIC INFORMATION RETRIEVAL SYSTEMS. *Proceedings of 6th International Conference on Music Information Retrieval*, London, UK, 11-15 September.

[2] Samson J. Genre. Grove Music Online[J]. *Oxford music online*, [Accessed May 2020].

[3] Enric Guaus i Termens. (2009) *Doctoral thesis:Audio content processing for automatic music genre classification: descriptors, databases, and classifiers.* Universitat Pompeu Fabra, Barcelona.

[4] George Tzanetakis and Perry Cook (2002) Musical Genre Classification of Audio Signals *IEEE Transactions on Speech and Audio Processing* .10(5):293 - 302

[5] Huiyu Zhou, Abdul Sadka and Richard M. Jiang. (2008) *Feature extraction for speech and music discrimination* .IEEE International Workshop on Content-Based Multimedia Indexing.

[6] Hareesh Bahuleyan (2018) Music Genre Classification using Machine Learning Techniques *arXiv:1804.01149* [Accessed May 2020].

[7] Tang, Chun Pui, et al. (2018) *Music genre classification using a hierarchical long short term memory (LSTM) model* Third International Workshop on Pattern Recognition. Vol. 10828.

[8] https://www.kaggle.com/carlthome/gtzan-genre-collection. [Accessed May 2020]

[9] https://wiki.aalto.fi/display/ITSP/Deltas+and+Delta-deltas. [Accessed May 2020]

[10] Konstantin Sozinov, Albina Shilo. (2017) *Music genre classification using Deep Neural Network*, DT2119 course project, KTH.

[11] Habibi, Aghdam, Hamed (2017) *Guide to convolutional neural networks : a practical application to traffic-sign detection and classification* Heravi, Elnaz Jahani. Cham, Switzerland.

[12] Simonyan, Karen, and Andrew Zisserman. (2014) *Very deep convolutional networks for large-scale image recognition* arXiv preprint arXiv:1409.1556.

[13] Wang L., Zhu H., Zhang X., Li S., Li W. (2020)*Transfer Learning for Music Classification and Regression Tasks Using Artist Tags*. Proceedings of the 7th Conference on Sound and Music Technology (CSMT).

Appendix

Here, we answer questions from peer review and describe which part of the report has been modified.

Regarding the criteria of relevance for the learning outcomes,the suggestion we received is doing some transfer learning on another dataset to see the performance on language classification. We think that's a good insight and if time permitted we could do further research. So far we did music genre classification, which is well suited for the course content.

Then for the literature study part, we modified our report following the suggestions we received. We did find there are a lot of related work on music genre classification, but our work was inspired by the previous work, we would like to do this work by using similar methods and also related to the course content. Due to limited space, we included some main related work in introduction section, where we mentioned some previous studies had been done on our topic. We also added motivation of using delta and delta-delta MFCC in introduction. About why we didn't describe MLP in detail, the reason is that MLP is a very basic model and is just a baseline in our experiment. In this report, we focused on CNN and LSTM models and gave detailed illustrations for these two models.

About the novelty, we combined deep neural networks with feature extraction. We applied what we learned from this course: MFCC, delta MFCC and delta delta MFCC. Those methods of feature extraction are combined with deep learning models: MLP, CNN and LSTM. Here, we concatenate the delta MFCC and delta-delta MFCC image with the orignal MFCC image as a new image for CNN network, we think it is a new way to do this work. Also regarding the future work on dynamic features, we got very good results in lab3 with MLPs (over 10% improvement for phoneme recognition), so there is a reason to examine that in this topic, but better with a large dataset.

As for the criteria of correctness, some wrong expressions about LSTM had been removed. We also realised the problem with that test accuracies are higher than validation or training accuracies. Therefore, we reran our experiments and got similar results. There are two reasons: one is that we trained less epochs (5 epochs) than before to alleviate the problem of overfitting; the other is that we did augmentation for the traning dataset, but for validation and test, we need to evaluate it on the original size, so we took the mode of 18 outputs to get the final accuracy which will get better results than on the resized data. Besides, we explained the parameters of data splitting in "Resized data" section, we think it's more natural to talk about data splitting when we resized the dataset in the experiment and we don't want to repeat it again if we put this part in dataset section earlier as suggested.

Regarding the review of "The authors achieved 90-100% training accuracies, but only half of that on the test and validation datasets", we did realize this issue. For instance, even though L2 regularization and dropout have been applied for CNN, they helped not much. Therefore, we considered the method of data augmentation since the dataset is small.

And the last suggestions we get are about the clarity of presentation, we added information the readers are interested to know about. For example, we added more details on experimental setup and data preparation. About the plots of loss and accuracy on training and test data, we did that but didn't put them in the report. Instead, we made a table to summarise the accuracy of each model. We think it's a good way to show and compare the results when there is no too much space left.

Above all, thanks to the reviews, we refined our experimental work and the report.