Music Genre classification using a hierarchical Long Short Term Memory (LSTM) model

Chun Pui Tang, Ka Long Chui, Ying Kin Yu, Zhiliang Zeng, Kin Hong Wong
Department of Computer Science and Engineering, The Chinese University of Hong Kong
Hong Kong
khwong@cse.cuhk.edu.hk

ABSTRACT

This paper examines the application of Long Short Term Memory (LSTM) model in music genre classification. We explored two different approaches in the paper. (1) In the first method, we used one single LSTM to directly classify 6 different genres of music. The method is implemented and the results are shown and discussed. (2) The first approach is only good for 6 or less genres, so in the second approach, we adopted a hierarchical divide-and-conquer strategy to achieve 10 genres classification. In this approach, music is classified into strong and mild genre classes. Strong genre includes hiphop, metal, pop, rock and reggae because usually they have heavier and stronger beats. And mild classes include jazz, disco, country, classic and blues because they tend to be softer musically. We even divide the sub-classes into sub-subclasses to help with the classification. First we classify an input piece into strong or mild class. Then for each subclass we further classify them until one of the ten final classes is identified. For the implementation, each subclass classification module is implemented using a LSTM. Our hierarchical divide-and-conquer idea is built and tested. The average classification accuracy of this approach for 10-genre classification is 52.975% which is higher than the state of the art approach (46.87% accuracy) of using a single convolution neural network. From our experimental results, we show that this hierarchical scheme improves the classification accuracy significantly.

KEYWORDS

Computer Music, LSTM, Music Genre classification

ACM Reference Format:
Chun Pui Tang, Ka Long Chui, Ying Kin Yu, Zhiliang Zeng, Kin Hong Wong. 2018. Music Genre classification using a hierarchical Long Short Term Memory (LSTM) model. In *Proceedings of (ICMR18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Nowadays, machine learning has been widely applied to many different fields, for examples healthcare, marketing, security and information retrieval. Artificial neural networks is one of the most effective techniques that are good at solving classification and prediction problems. In this project, we apply an artificial neural network to music genre classification. Our target is to classify music in different genres, for exmaple 6-10 different genres.Our algorithm is very useful for the user to search for their favorite music pieces and has great commercial potential. The applications of machine learning techniques to classification is not as common as that the image classification. Tao et. al. [1] have created a deep learning model that can identify the music from at most 4 different genres in a dataset. Matan Lachmish [2] adopted the Convolutional Neural Network (CNN) to tackle the problem. However, their results are not satisfactory. Its accuracy would eventually drop as the number of music genres increases. In this project, we make use of the Long short-term memory (LSTM) model instead of CNN in music genre classification. We are able to train a model to classify music from 6 to 8 different genres. Furthermore, we adopt a divide-and-conquer scheme to further improve the accuracy. Firstly, we divide the music into strong and mild classes. Genres of strong classes include hiphop, metal, pop, rock and reggae. The mild class includes jazz, disco, country, classic, and blues . An LSTM classifier is trained to categorize music into these two classes. Then the music is further classified into subclasses. Under the strong subclass , there are sub-strong1 (hiphop, metal, rock), and sub-strong2 (pop, reggae). For the mild class, the sub-classes include sub-mild1 (dico, country) and sub-mild2 (jazz, classic, blues). From our experimental results, we show that this hierarchical scheme improves the classification accuracy.

The remaining sections of this paper are arranged as follows. In section 2, we introduce the background of our work. In Section 3, the theory used is discussed. In Section 4, we discuss the implementations and reults of our LSTM approaches. The conclusion is found in Section 5.

2 BACKGROUND

Classification is one of the most important applications in machine learning. Recently, a popular tool called Tensorflow [3] has demonstrated a number of examples about neural network applications, such as character and object recognition. One of them is the classification of iris into three iris species from their images. The Iris species classification system contains only 50 samples of each species and the approach is able to reach an accuracy over 90% . Research related to music is interesting and has many commercial applications. Machine learning can provide elegant solutions to music signal processing like beat detection, music emotion recognition and chord recognition. Yann Bayle [4] published a survey on deep learning in music. Figure 1 shows the related fields and their percentages in the current computer music research literature. We

ICMR18, 11-14 June 2018, ,Yokohama,Japan

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

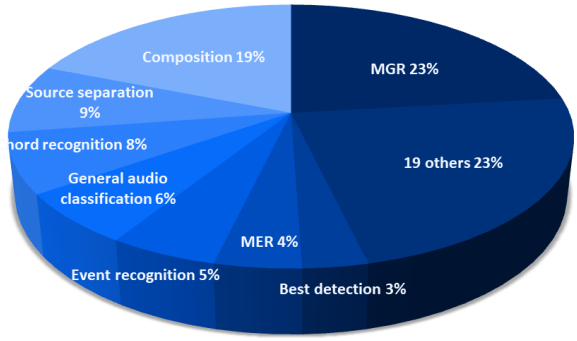

Figure 1: The popularity of deep learning for different kinds of music research [4]

Table 1: Results of the system described in [1]

Number of Genres	Testing set
2 genres (Classic, Metal)	98.15%
3 genres (Classic, Metal, Blues)	69.16%
4 genres (Classic, Metal, Blues, Disco)	51.88%

found that composition and audio classification are the most common. For examples, a project on environmental sound classification is reported in [5]. Also, the system BachBot by Feynman Liang [6] is a music composition robot which uses LSTM to create J.S. Bach like music pieces.

In this paper, we are interested in applying machine learning to music genre classification. Feng et. al. [1] devised an algorithm to classify music into in 2 to 4 genres. Their results are summarized as below.

From Table 1, we can see that the accuracy of classifying 2 genres is 98.15%. However, when the number of genres is increased to 4, the accuracy is reduced by 17%. Another work tackling the same problem is proposed by Matan Lachmish [2]. Their approach uses the convolutional neural network model. They achieved an accuracy of 46.87% in classifying music into 10 different genres. In this paper, we are going to use the LSTM model to solve the music classification problem. To the best of our knowledge, we believe that we are one of the first group to try to tackle this problem using LSTM.

3 THEORY

Figure 2 illustrates the overall structure of our project. We use the Gtzan [7] music dataset to train our system. We apply the Librosa library [8] to extract audio features, i.e. the Mel-frequency cepstral coefficients (MFCC), from the raw data. The extracted features are input to the Long Short-Term Memory (LSTM) neural network model for training. Our LSTM are built with Keras [9] and Tensorflow[3].

3.1 Mel frequency cepstral coefficients(MFCC)

MFCC features are commonly used for speech recognition, music genre classification and audio signal similarity measurement. The computation of MFCC has already been discussed in various paper, for example [10]. We will focus on how to apply the MFCC data for

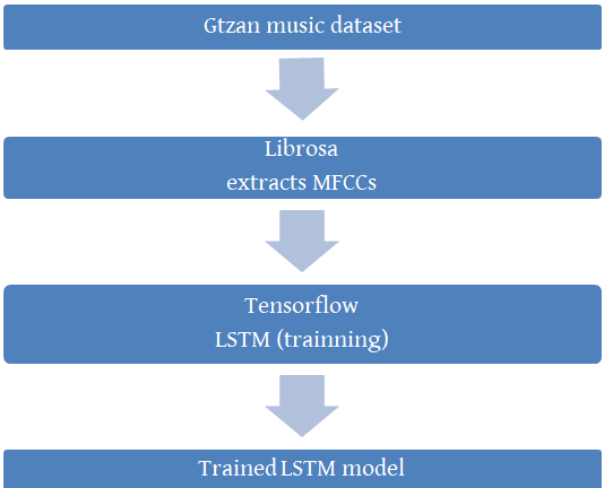

Figure 2: Overview of music genre classification process

```
y, sr = librosa.load(file)
mfcc = librosa.feature.mfcc(y=y, sr=sr, hop_length=hop_length, n_mfcc=13)
```

Figure 3: Librosa MFCC feature extraction

```
array([[ -5.229e+02,  -4.944e+02, ...,  -5.229e+02,  -5.229e+02],
       [  7.105e-15,   3.787e+01, ...,  -7.105e-15,  -7.105e-15],
       ...,
       [  1.066e-14,  -7.500e+00, ...,   1.421e-14,   1.421e-14],
       [  3.109e-14,  -5.058e+00, ...,   2.931e-14,   2.931e-14]])
```

Figure 4: Sample outputs of Librosa

our application. In practice, we use the Librosa library to extract the MFCCs from the audio tracks. Figure 4 shows the output of the program. It is a 2-D array. One dimension represents time while the other dimension represents the different frequencies.

3.2 The Long Short Term Memory Network (LSTM)

Approaches such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are popular machine learning framework. The LSTM network used in this project is a subclass of RNN. RNN is different from the traditional neural networks. It can memorize the past data and be able to predict with the help of the information stored in the memory. Moreover, LSTM solves the RNN long term dependencies problem. Although RNN model can make use of the past information to predict the current state, the RNN model may fail to link up the information when the gap between the past information and the current state is too large. The details of the long-term dependencies has been discussed in [11]. Figure 5 reveals the structure of a typical LSTM model [11]. Figure 6 shows the configuration of our LSTM network. The network has 4 layers. A LSTM can be formulated mathematcially as follows:

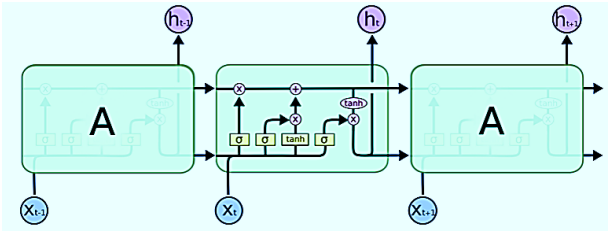

Figure 5: A typical LSTM model contains four interacting layer [11].

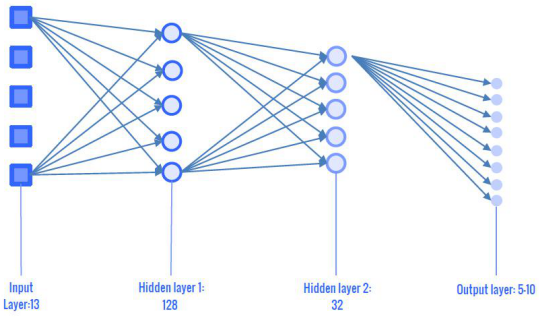

Figure 6: The LSTM network used in our music genre classification problem

Table 2: The design of our LSTM network in experiment1

Input Layer(I)	13 MFCCs features obtained as input
Hidden Layer(II)	128 neurons
Hidden Layer(III)	32 neurons
Output Layer(IV)	6 output corresponding to 6 different genre of music

$$
\begin{aligned}
u_t &= \tanh(W_{xu} * x_t + W_{hu} * h_{t-1} + b_u) : \text{update eq.} \\
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i), \text{input gate eq.} \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f), \text{forget gate eq.} \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o), \text{output gate eq.} \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \text{cell state} \\
h_t &= \tanh c_t \odot o_t, \text{cell output} \\
output_{class} &= \sigma(h_t * W_{outpara})
\end{aligned} \tag{1}
$$

where W_{xu}, W_{xi}, W_{xf}, W_{xo} and W_{hu}, W_{hi}, W_{hf}, W_{ho}, $W_{outpara}$ are weights, and b_u, b_i, b_f, b_o are biases to be computed during training. h_t is the output of a neuron at time t. $\odot$ denotes pointwise multiplication. $\sigma()$ denotes a sigma function and tanh() represents the tanh function. The input x_t is the MFCC parameters at time t. $output_{class}$ is the classification output. In first version of our approach, we use 6 output nodes that corresponds to 6 music genres.

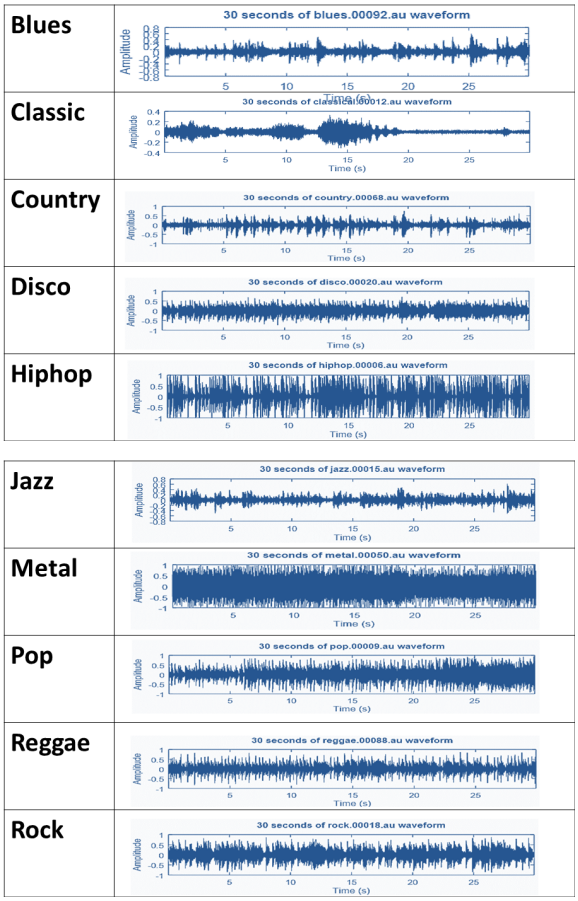

Figure 7: Sample waveforms of different music genres.

4 IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1 Our dataset

We used the GTZAN dataset [7] that contains of various samples of the ten music genres in our experiments. The genres are blues, classic, country, disco, hip-hop, jazz, metal, pop, Reggae and rock. Each genre includes 100 soundtracks of 30 seconds long in .au format. We randomly chose samples from the dataset for training and testing. Using the script written by Kamil Wojcicki [12], we created the waveforms of the soundtracks and compared their similarity. Samples of the waveforms are shown in Figure 7. We use a scheme of 30% data for testing and 70% data for training, and the testing and training data are not overlapped. We compared the waveforms of 10 different genres, and found that Blues is similar to Jazz and Country. Rock is similar to Pop and Reggae. So we decided to use music from the classic, hip-hop, jazz, metal, pop and reggae group to form the six genres for training in our first experiment.

4.2 Preprocessing

Before we can use the data in the GTZAN dataset, we need to preprocess the signals so that they can be input to the Long Short

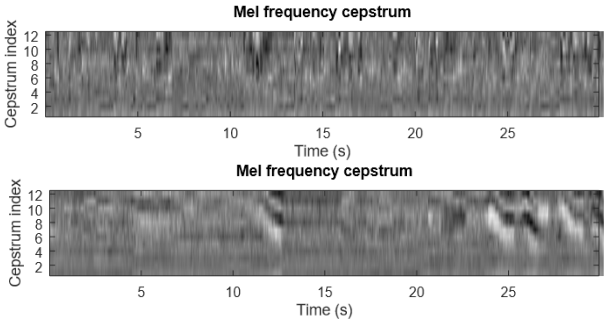

Figure 8: Visualization of Mel frequency cepstrum

Term Memory (LSMT) model. MFCC is a good representation of music signals. It is one of the best indicators of the 'brightness' of the sound. It is able to measure the timbre of the music according to Schubert et. al. [13]. We used the Librosa library [8] to transform the raw data from GTZAN into MFCC features. In particular, we chose the frame size as 25ms. Each 30-second soundtrack has 1293 frames and 13 MFCC features, C1 to C13, in experiment1 (14 MFCC features, C0 to C13, in expeirment2). Figure 8 shows some examples of the Mel frequency cepstrum plots of the music signals in the database. In the following discussion, expeirment1 is used to demonstrate how to process data to achieve classification using LSTM. For example, how epochs may affect classification results. Experiment2 is our actual classification work of classifying different genres of music.

4.3 Experiment 1 : LSTM for 6-genre classification

We divided the GTZAN dataset into three sub-sets for training, validation and testing. There are 420 audio tracks in the data set for training, 120 for validation and 60 for testing. Each audio track lasts for 30 seconds. We set the batch size that defines the number of samples to be propagated through the network for training as 35. We tested the model by using different Optimizer, Epoch, and kept other conditions constant. Firstly, we applied the Adam optimizer with 5, 10, 20, 50, 100, 200, 400 Epochs. For each case, we performed 4 trials. The accuracy and loss of validation and testing were recorded. The results are shown in Figure 9. We can see that the accuracy and loss are improving within 20 Epochs. At 20, the test accuracy reaches the maximum and the loss is minimized. When it is over 20, we cannot see any significant improvement on the accuracy and loss values. On the contrary, both of these values increase along with the number of Epochs. In fact, the accuracy of the testing set increases along with the epoch number and eventually exceed 90% around 400 epochs. It may be due to the effect of overfitting.

4.4 Discussion of experiment 1 of our LSTM method for 6-genre classification

- Accuracy of experiment 1: We obtained an accuracy of around 0.5 to 0.6. There are still some rooms for improvement. With more training samples, we may be able to achieve an accuracy of 0.6 to 0.7. Another problem is that while the number of epochs grows, loss due to overfitting occurs.

Epoch	Validation accuracy	Validation loss	Testing accuracy	Testing loss
5	0.5250	1.3172	0.4500	1.3332
5	0.5000	1.3262	0.5333	1.2054
5	0.4333	1.4384	0.4833	1.2220
5	0.5083	1.2990	0.4333	1.3837
10	0.5583	1.0426	0.4833	1.1811
10	0.5500	1.2642	0.4333	1.3320
10	0.5000	1.1596	0.4333	1.1585
10	0.5333	1.1805	0.5167	1.2791
20	0.6250	1.0420	0.6167	1.0461
20	0.5750	1.1938	0.5667	1.2148
20	0.5667	1.119	0.5167	1.1090
20	0.5500	1.1478	0.5833	1.3487
50	0.6250	1.1019	0.5333	1.2493
50	0.5917	1.3085	0.5500	1.6167
50	0.5833	1.3688	0.5333	1.4257
50	0.5250	1.5030	0.5167	1.3428
100	0.5250	1.6511	0.5167	1.7444
100	0.5917	1.6230	0.5000	1.8762
100	0.6250	1.4189	0.5833	1.3538
100	0.5583	1.4247	0.6333	1.1702
200	0.5583	1.9404	0.5833	2.0062
200	0.5833	1.8259	0.6000	1.5157
200	0.5333	1.9841	0.4500	1.8396
200	0.5833	1.7292	0.4666	2.1839
400	0.5917	2.1531	0.5333	1.9813
400	0.5583	2.2721	0.5000	2.0047
400	0.5833	2.0219	0.5667	1.8939
400	0.5750	2.0028	0.5333	2.1367

Figure 9: The results of the proposed LSTM model. Classification accuracy and loss are shown in the table.

Epoch	Avg. Validation Accuracy	Avg. Validation Loss	Avg. Testing Accuracy	Avg. Testing Loss
5	0.4917	1.3452	0.4750	1.2861
10	0.5354	1.1617	0.4667	1.2377
20	0.5792	1.1259	0.5709	1.1797
50	0.5813	1.3206	0.5333	1.4086
100	0.5750	1.5294	0.5583	1.5362
200	0.5646	1.8699	0.5250	1.8864
400	0.5771	2.1125	0.5333	2.0042

Figure 10: The results of the proposed LSTM model. Classification accuracy and loss are shown in the table.

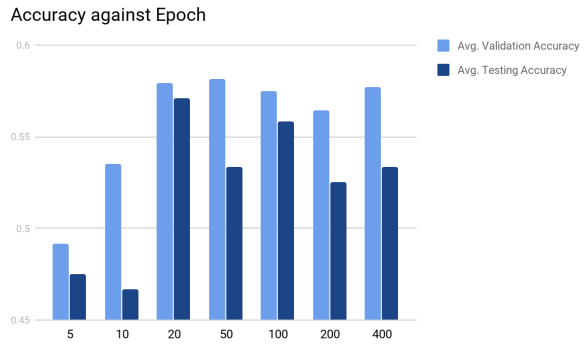

Figure 11: Accuracy against the number of epochs

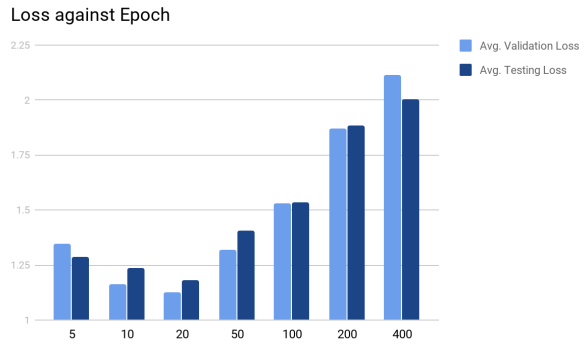

Figure 12: Avgerage loss against the number of epochs

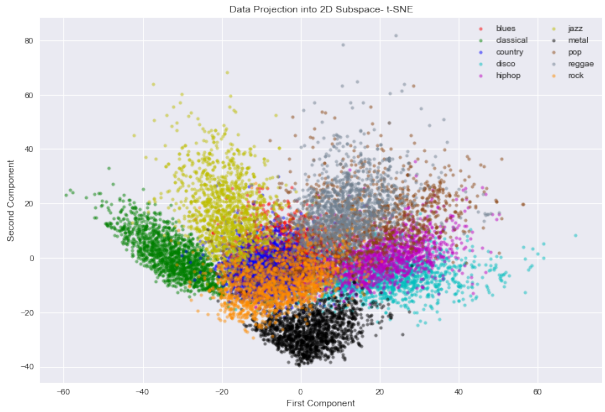

Figure 13: Classification of music genre in GTZAN [14]

,

- Difficulties and Limitations: The major limitation is the small training data size. It leads to low accuracy and overfitting. Although some genre such as metal, are outstanding and easy to be recognized, it is hard to classify some other genre that are quite similar. Other work, which also used GTZAN dataset, has shown that there is a overlapping of some features among different genres. From Figure 13, we see that the data points of pop music overlap with other genres. It is reasonable because pop song can include many other genre features.

4.5 Experiment 2 : The divide-and-conquer approach for 10-genre classification

A divide-and-conquer scheme as mentioned in the theory section is used. In our scheme, we applied 7 LSTMs. Then a multi-step classifier involving all these 7 LSTM classifiers were used in the classification to achieve 10-grenre classification. The division of samples for training, and testing is the same as that in experiment 1 above (30% data for testing and 70% for training). The LSTM classifiers involved are shown as below.

- LSTM1: It classifies strong (hiphop metal pop rock reggae) and mild (jazz disco country classic blues) music.

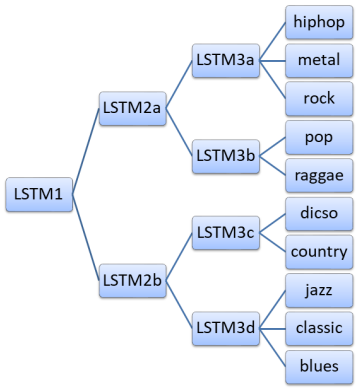

Figure 14: Tree diagram of our approach

,

- LSTM2a: It divides the music into Sub-strong1 (hiphop, metal and rock) and Sub-strong2 (pop and reggae) classes. During training, only music samples of hiphop, metal, rock, pop and reggae are involved.
- LSTM2b: It categorizes music into Sub-mild1(disco and country) and Sub-mild2 (jazz, classic and blues) groups. We used samples only from of disco, country, jazz, classic and blues for training.
- LSTM3a: It classifies music into hiphop, metal and rock. Only music from hiphop, metal and rock class are involved.
- LSTM3b: It differentiates pop music from reggae
- LSTM3c: It differentiates disco music from country.
- LSTM3d: It recognizes jazz, classic and blues.

The proposed multi-step classifier involves the above 7 LSTMs. In the testing stage, the input music is first classified by LSTM1 to find if it is strong or mild. Then according to the result, either LSTM2a or LSTM2b is applied. Finally, LSTM3a, 3b, 3c or 3d is used to classify the music into the target categories according the results obtained in the previous level. Results of this experiment are shown in Table 3. A tree diagram showing the relation of the different levels of LSTMs is shown in Figure 14

Table 3: Results of experiment 2. The accuracy of each LSTM components in the proposed multi-step classifier

LSTM classifiers	Accuracy	Epochs
LSTM1 (strong, mild)	80.0%	35
LSTM2a (sub-stong1, sub-strong2)	81.6%	20
LSTM2b (sub-mild1, sub-mild2)	81.6%	35
LSTM3a (hiphop, metal, rock)	74.6%	40
LSTM3b (pop, reggae)	88.0%	20
LSTM3c (disco, country)	78.0%	20
LSTM3d (jazz, classic, blues)	84.0%	40

4.6 Discussion of experiment 2 of using a divide-and-conquer with LSTM method for 10-genre classification

Using the results in table 3, we can generate a table for the 10 genre classification results. Assume the unknown is passed to the LSTM1 (strong, mild) and then to LSTM2a (sub-strong1, sub-strong2) or LSTM2b(sub-mild1, sub-mild2). Finally it is handled by LSTM3a, LSTM3b, LSTM3c or LSTM3d. The predicted results is shown in table 3. It shows it is comaprable to the resulst in [2].

Table 4: Results of experiment 3. The accuracy of each of the 10 genres using the proposed multi-step classifier

LSTM classifiers	Accuracy of our method	Method in [2]
hiphop, metal, rock	48.7%	Not applied
pop, reggae	57.45%	Not applied
disco, country	50.92%	Not applied
jazz, classic, blues	54.84%	Not applied
average of above	52.975%	46.87%

The above accuracies of each of the 10 genres is calculated by combining the results of three classifiers. For example the accuracy of hiphop (or metal, rock) is calculated by the accuracy of LSTM1*LSYM2a*LSTM3a = 80%*81.6%*74.6%=48.7% . The result of all genres are shown in table 4. Our method for 10-genre classification is compared to the one proposed by Matan Mlachmish [2], it shows our average accuracy is 52.975% which is higher than 46.87% reported in the [2].

5 CONCLUSION

In conclusion, our testing result shows our Long Short-Term Memory (LSTM) model has potential for a good engine for building a music genres classifier. First we have achieved an accuracy of 50-60% for a direct approach of using LSTM for 6-genres. In another experiment we use a divide-and conquer approach for 10 Genres and our result is from 48.7% to 57.45% (average = 52.975%) accuracy, which is better than the state of the art approach of 46.87%. In this project, we have successfully used a recursive neural network for a music genre classification problem and we will work on refining the approach for better accuracy and efficiency.

REFERENCES

[1] Tao Feng. Deep learning for music genre classification. https://courses.engr.illinois.edu/ece544na/fa2014/Tao_Feng.pdf. Accessed: 16- Jov- 2018.

[2] Mlachmish. Music genre classification with cnn. https://github.com/mlachmish/MusicGenreClassification/blob/master/README.md. Accessed: 16- Jov- 2018.

[3] Google. Tensorflow: Tensorflow is an open source software library for machine intelligence. https://www.tensorflow.org. Accessed: 16- Jan- 2018.

[4] Yann Bayle. Dl4m: Deep learning for music. https://github.com/ybayle/awesome-deep-learning-music/blob/master/README.md. Accessed: 16- Jan- 2018.

[5] Raman Arora and Robert A Lutfi. An efficient code for environmental sound classification. *The Journal of the Acoustical Society of America*, 126(1):7–10, 2009.

[6] Feynman Liang. Bachbot: a research project utilizing long short term memory (lstms) to generate bach compositions. https://github.com/feynmanliang/bachbot/. Accessed: 16- Jan- 2018.

[7] GTZAN. Gtzan genre data set. http://marsyasweb.appspot.com/download/data_sets/. Accessed: 16- Jov- 2018.

[8] Librosa. Librosa : a python package for music and audio analysis. https://librosa.github.io/librosa/. Accessed: 16- Jan- 2018.

[9] Keras. Keras: The python deep learning library. https://keras.io/. Accessed: 16- Jan- 2018.

[10] Md Sahidullah, Sandipan Chakroborty, and Goutam Saha. Improving performance of speaker identification system using complementary information fusion. *arXiv preprint arXiv:1105.2770*, 2011.

[11] Christopher Olah. Htk mfcc matlab: Mel frequency cepstral coefficient feature extraction that closely matches that of htk's hcopy. URLhttp://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png. Accessed: 16- Jov- 2018.

[12] Kamil Wojcicki. Htk mfcc matlab: Mel frequency cepstral coefficient feature extraction that closely matches that of htk's hcopy. URLhttps://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab?requestedDomain=true. Accessed: 16- Jov- 2018.

[13] Emery Schubert, Joe Wolfe, and Alex Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the international conference on music perception and cognition, North Western University, Illinois*, pages 112–116. sn, 2004.

[14] Dmytro Perekrestenko. Visualisation of gtzan dataset sparse representation. https://lts2.epfl.ch/blog/perekres/2015/04/27/visualisation-of-gtzan-dataset-sparse-representation/. Accessed: 16- Jov- 2018.