Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 99–109

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12–13 September 2019

Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks

Adiyansjah[a], Alexander A S Gunawan[a,*], Derwin Suhartono[a]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

With commercial music streaming service which can be accessed from mobile devices, the availability of digital music currently is abundant compared to previous era. Sorting out all this digital music is a very time-consuming and causes information fatigue. Therefore, it is very useful to develop a music recommender system that can search in the music libraries automatically and suggest suitable songs to users. By using music recommender system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of the music that has been heard previously. Our research would like to develop a music recommender system that can give recommendations based on similarity of features on audio signal. This study uses convolutional recurrent neural network (CRNN) for feature extraction and similarity distance to look similarity between features. The results of this study indicate that users prefer recommendations that consider music genres compared to recommendations based solely on similarity.

Keywords: Music Recommender System; Convolutional Recurrent Neural Network; Similarity Distance;

* Corresponding author. Tel.: +62-21- 5345830.
E-mail address: aagung@binus.edu

10.1016/j.procs.2019.08.146

1. Introduction

Music is one of the popular entertainment media in the digital era. Music is considered as the work of human creativity to express ideas and emotions in the form of sounds that consist of melody, harmony and rhythm. Music can be categorized into several genres, such as pop, rock, jazz, blues, folk etc. Listening to music in the digital age is easier because of the features on the smartphone that can play music offline and online [1]. Nowadays, the availability of digital music is very abundant compared to the previous era, so to sort out all this digital music is very time consuming and causes information fatigue. Therefore, it is very useful to develop a music recommender system that can search music libraries automatically and suggest songs that are suitable for users.

Music streaming applications like Spotify and Pandora have features to recommend music to users. These features can help to get a list of appropriate music from the popular music libraries based on music that has been heard previously. This makes the recommender system [2] play an important role in maintaining the streaming music business. Music recommendations are done by looking for similarities from one music to another or by giving preference from one user to another [3]. The challenge of music recommender system is to create a system that can continually find attractive new music which understand the users' preferences in music [4]. This requires that the music personalized recommender system should effectively reflect the personal preferences. It needs adjustments to achieve personalized recommendations for the needs of different audiences. Therefore, the music personalized recommender system is a more complicated than the general recommender system. It is necessary to consider user needs comprehensively and combines the music feature recognition and audio processing technologies to extract the music features. The objective of this paper is to implement a personalized recommender system, which has practical significance and great vale in research.

To reach the objective, our research approach is based on comparing the similarity of features on audio signal. This approach can be considered as content-based music recommendation, where the recommendations is based on perceptual resemblance of what was previously heard by the user. This approach requires the definition of similarity metric, which used to measure the similarity between audio signals [3]. We use convolutional recurrent neural networks (CRNNs) for feature extraction and similarity distance to look similarity between features. CRNNs is combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). RNNs were designed to work with time series data especially for temporal sequence prediction problems. RNNs have the most successful results when working with word and paragraph sequences in natural language processing. CNNs are especially suited for predicting high level music features such as chords and beats, because they allow hierarchical structure consisting of intermediate features on multiple timescales. Our approach was inspired from Choi et al works [5, 6]. When developing music genre classification system, Choi et al [6] conducted a study to combine CNNs and RNNs. In the study, they compared the performance of several CNNs architectures with CRNNs for classifying music genres. The model takes as an input the spectrogram of music frames and analyzes the image using CRNNs. The model output is a vector of predicted genres for the song. The main result of the study is that the accuracy of CRNNs is slightly higher than CNNs methods which combining frequency and time domains and using the same number of parameters. In their previous paper [5], they compared three audio representations for automatic tagging, that is: STFT, MFCC, and Mel-spectrogram. CNNs is used to train high level music features for automatic tagging. The results of the study concluded that CNNs is very effective mainly when use Mel-spectrograms as audio representation compared to STFT and MFCC. We follow these results in using Mel-spectrograms for audio representation and CRNNs for feature extraction. Finally, recommendations will be made by comparing the similarities of the features. Our research consists of several stages (see Fig. 1), that is:

1. Beginning stage, comprises problem identification and literature study
2. Collecting music data.
3. Data cleaning and selection.
4. Preprocessing audio.
5. Modeling neural network.
6. Modeling recommender system.
7. Designing, implementing and evaluating the music recommender system application.

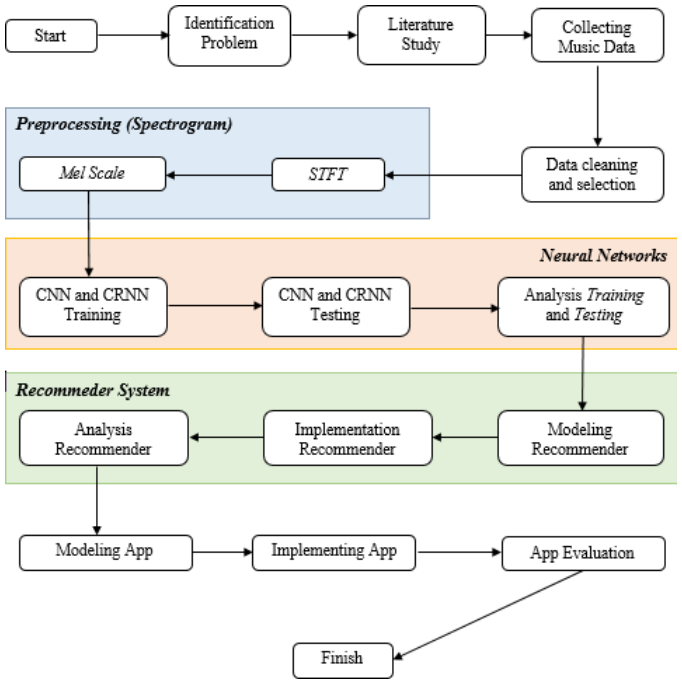

Fig. 1. Steps of the music recommender system research

The remainder of the paper is composed as follows: first we discuss music recommender system in section 2, and then is followed by explanation of CRNNs, in section 3. In section 4, we discuss implementation of music recommender system based on CRNNs. Furthermore, we report the experiment results in music recommender system based on metrics distance and respondent opinions in section 5. Finally, we concluded our work with suggestions for the future research in section 6.

2. Music Recommender System

Recommender System [7] is a software tool and algorithm that gives recommendations for items that is most interesting to a user. Recommendations is related to many kinds of real applications, such as what commodities are purchased, what songs is listened, or what latest news is read. On the other hand, there is a transformation of the recorded commodity music, specifically when Apple bought Beats Music in 2014 [8]. Recently, the business model in music industry is changing from being dependent on commodity sales to a model based on subscriptions and streaming. With the new business model in music industry, the availability of digital music currently is abundant compared to previous era. Therefore, the role music recommender system for the music providers is essential. It can predict and then offer the appropriate songs to their users, consequently the music providers can increase user satisfaction and sell more diverse music.

Generally, music recommender system can be divided into three main parts [9], that is: (i) users, (ii) items and (iii) user-item matching algorithms. Firstly, to differentiate user's music tastes, we can develop user modelling based on user profiling such as geographic region, age, gender, life styles, and interests. User modelling will model the difference in user profile to determine their choices of music. For example, Bu et al [10] used social media information to give more accurate music recommendations. Second, item profiling comprises of three kinds of metadata, that is: editorial, cultural and acoustic. They can be used in a variety of recommender system, for example Bogdanov et al [11] exploited genre metadata to increase the listeners satisfaction. Finally, the matching algorithm should be able to automatically recommend personalized music to listeners. There are two main approaches of matching algorithm, that is collaborative filtering and content-based filtering.

Collaborative filtering uses the collaborative power of the available assessment by users to make recommendations. It assumes if different users rate music items similarly or have similar behavior, they will rate on other music items similarly [12]. The main challenge in collaborative filtering methods is the sparse assessment matrix because most users only see a small part of all music libraries, consequently most assessments are not determined. On the other hand, content-based filtering uses the features of the music items to make recommendations. In here, "content" means information in the items. Content-based approach typically employ a two-stage approach [13], (i) extract traditional audio content features, and (ii) predict user preferences. Many researches have been focused on extracting and comparing the acoustic features such as timbre, rhythm in finding audio perceptual similarity. Both collaborative and content-based filtering will result in personalized recommendations in the form of item rankings. Our approach can be regarded as content-based music recommendation because it is based on comparing the similarity of features on audio signal. We give music recommendations based on the similarity of perceptions of the preferred music that the user has previously heard.

3. Convolutional Recurrent Neural Networks

In this section, we will explain our research steps in detail, with the focus to describe Convolutional Recurrent Neural Networks (CRNNs) architecture.

3.1. Collecting Music Data

Data is collected by downloading available datasets. The dataset comes from the Free Music Archive (FMA) [14] music collection that is legal to download. For the research, it is used fma_medium dataset which data volume is 22 GB and total collections are 25,000 music in mp3 format. This dataset is imbalance with 16 genres of music and duration of each music is 30 seconds. For our research, we only used a part of the FMA dataset by choosing the good quality records and maintaining the balance of data for each genre.

3.2. Data Cleaning and Selection

First, data that has been collected is cleaned up by discarding music with low audio quality and music which has wrong genre label. The cleaning process is done easily by sorting the music based on the creator, because music with the same creator has the same audio quality in the dataset. We also removed the music which formed of two or more genres, such as combination of pop and folk. Finally, we chose only seven genres for maintaining the data balance, that is: classical, electronic, folk, hip-hop, instrumental, jazz and rock. The following table is the amount of data for each music genres after the process of data cleaning and data selecting.

Table 1. Total data in each genre

Music Genre	Total Data
Classical	1000
Electronic	1000
Folk	1000
Hip-Hop	1000
Instrumental	1000
Jazz	1000
Rock	1000

3.3. Audio Preprocessing

We need a suitable audio representation as input for neural networks architecture. Therefore, data in the form of audio signal is converted into spectrogram image. First, to do this conversion, the audio signal is processed with a certain sampling rate. Next, we applied Short Time Fourier Transform (STFT) at each sampling rate with a certain window length and hop length, thus the frequency of the audio signal can be calculated at a specific time. To smooth

the frequency and avoid the spectral leakage, a window function is used. Finally, the results from STFT are converted into Mel scale. Mel-spectrograms is considered the most effective representation compared to other audio representations [5]. Figure 2 show the result of preprocessing step. Parameters which used in the research as follows: the sampling rate is 22050, the window length is 2048 with hop length is 512 and the chosen window filter is Hanning function.

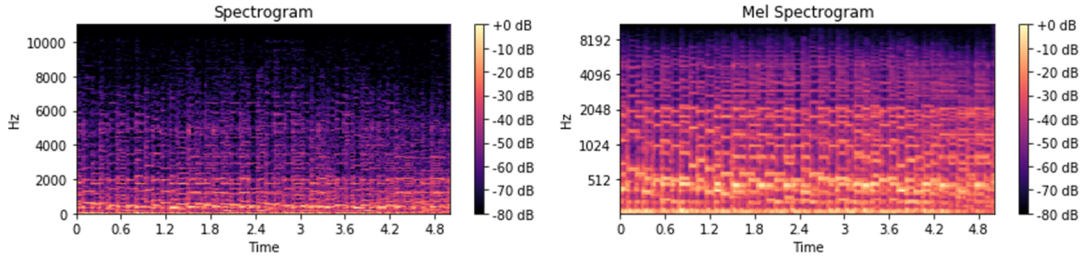

Fig. 2. Spectrogram and Mel-Spectrogram

3.4. Modeling Neural Networks

To classify the music genres, we used two architectures (see Figure 3 and 4), that is: Convolutional Neural Networks (CNNs) and Convolutional Recurrent Neural Network (CRNNs). CRNNs is used because the model can extract the important features for the prediction results. Not only looking at frequency features on the spectrogram, CRNNs also can look at time sequence patterns. Finally, feature vectors that produce before the classification layer can be used as a basis for recommendations.

The following is an explanation of the training and testing stages on the CNNs and CRNNs models. First we divided Mel-spectrogram data into training, validation, and testing dataset, which the proportion is 8:1:1 respectively. Because the model is binary classification, labelling is done by using one hot encoding. It can be done by giving a value of 1 to a chosen music genre and 0 to other genres. In the training step, it is used the ADAM optimizer with parameters $\alpha = 0.001$, $\beta_1 = 0.9$, dan $\beta_2 = 0.999$. The training model is binary classifier and the loss function is binary cross-entropy. Parameters $batch\_size = 50$ and $epoochs = 100$. The batch size is not too large, so that the processing calculation does not exceed the RAM capacity. For both CNNs and CRNNs architectures, the input dimensions of Mel-spectrogram image are 96 × 1366. Figure 3 shows illustration of CNN structure with dimensional change for each layer. The following are the components of the CNNs structure:

- Five layers of convolutional with kernel 3×3, feature maps (47-95-95-142-190), stride 1 and using padding to maintain input dimensions.
- Each convolutional using batch normalization and ReLU activation.
- Max-pooling layers using kernel $((2 \times 4)$-(2×4)-(2×4)-(3×5)-$(4 \times 4))$ and with same stride.
- The output layer is sigmoid function.

The CRNNs architecture used two layers of RNN with Gated Recurrent Units (GRU) to summarize 2D temporal patterns from the results of four CNN layers. CNNs was first performed on this model, which was used for extracting local features. The results of sub-sampling are feature maps with a large N × 1 × 15 (number of feature maps × frequency × time) that will be used for two layers of RNNs. Figure 4 shows illustration of CRNN structure with dimensional change for each layer. The following are the components of the CRNNs structure:

- Four layers of convolutional with kernel 3 × 3, feature maps (68-137-137-137), stride 1 and using padding to maintain input dimensions.
- Each convolutional using batch normalization and ReLU activation.
- Max-pooling layer for each convolutional layer with kernel $((2 \times 2)$-(3×3)-(4×4)-$(4 \times 4))$ and same stride.
- Each convolutional layer using dropout with rate 0.1

- Two layers of GRU with 68 feature maps. In the first layer, input data is in the form of sequences and outputs in the form of sequences. In the second layer, the input data is in the form of a sequence and the output is in the form of a single value.
- The output layer is sigmoid function.

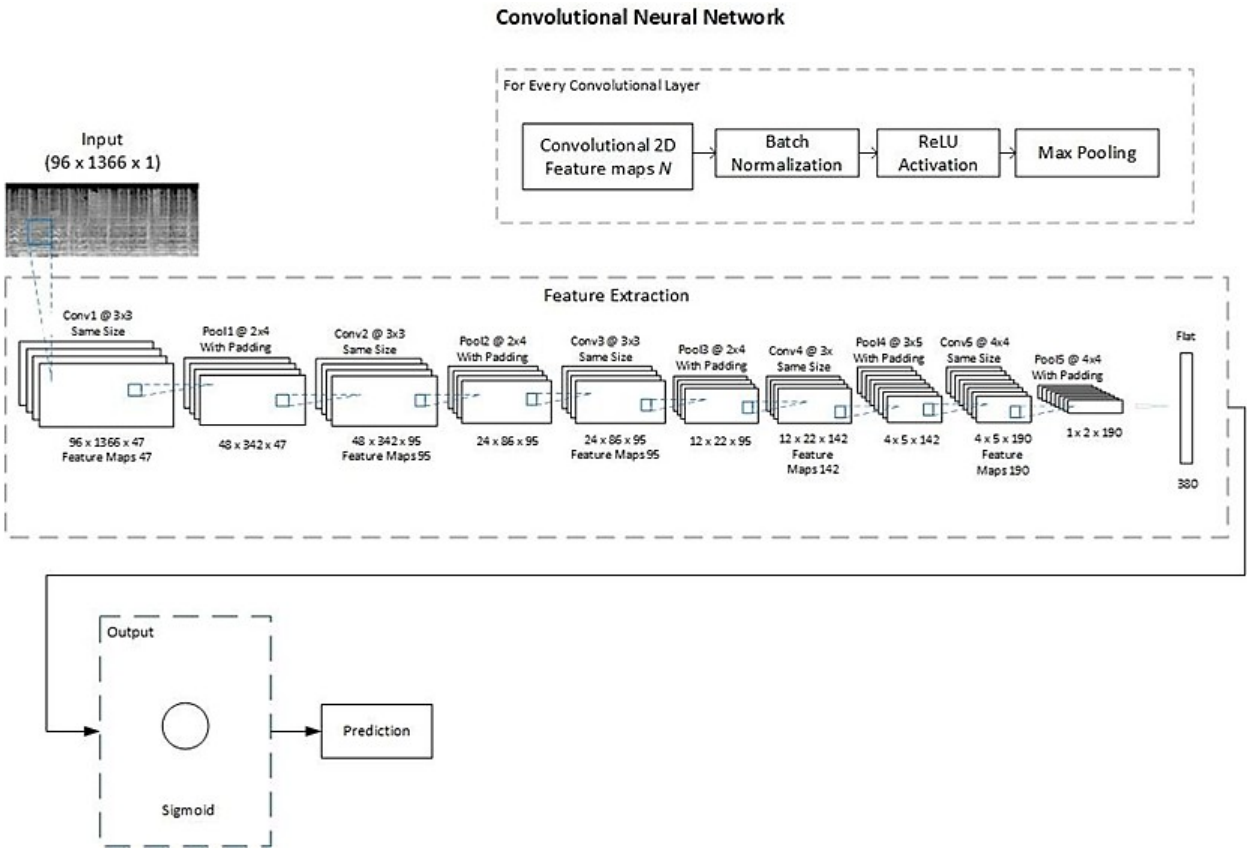

Fig. 3. Structure of CNN

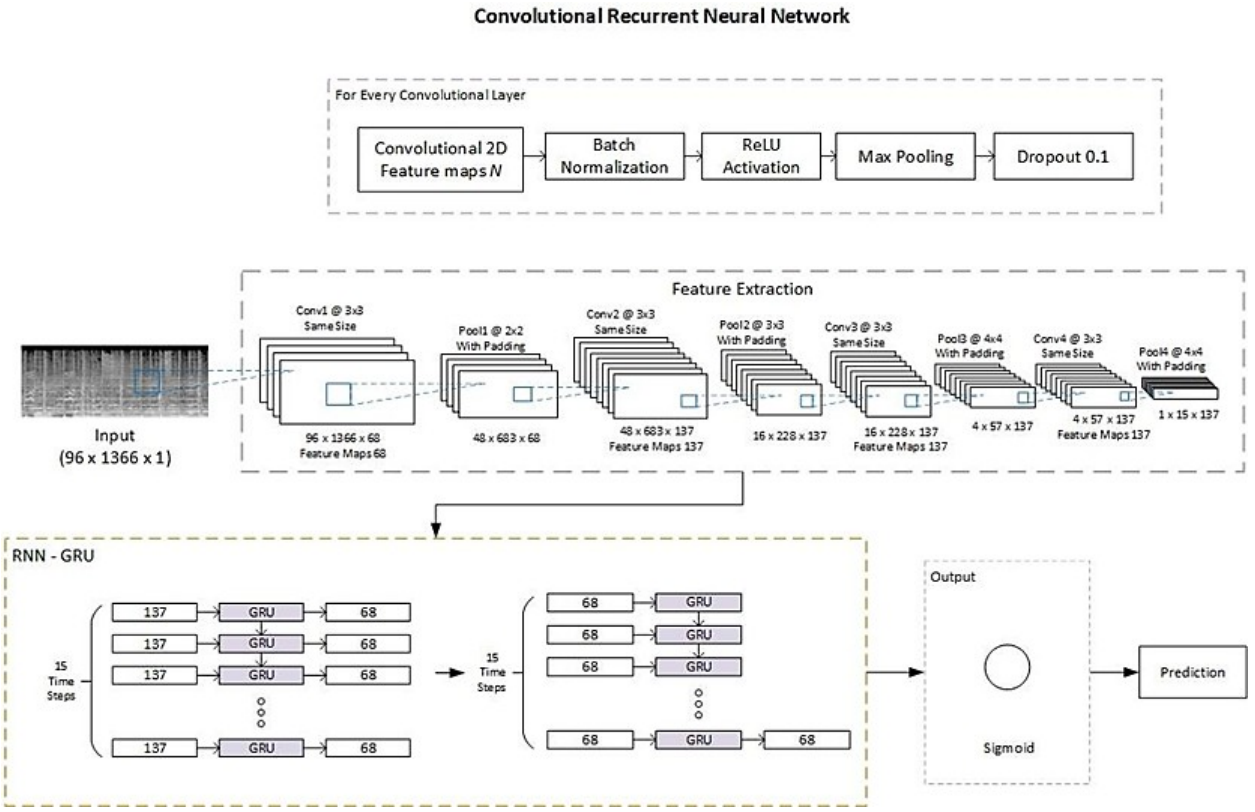

Fig. 4. Structure of CRNN

4. Implementation of Music Recommender System

The recommender system is done by calculating cosine similarity of extraction features (equation 1) from one music to another music. The extraction features are in vector form; thus, it is possible to calculate their distance. First, we chose one music for each genre as the basis for the recommender system. Next the prediction of the basis music genre is calculated based on neural networks. The feature vectors that produce before the classification layer are used as a basis for recommendations. After the basis music features are obtained, cosine similarity calculations are performed on other music features. To calculate the similarity of the two music with a number of features equal to N, where the first music has a feature vector $x = [x_1, x_2, \dots, x_n]$ and the second music has a feature vector $y = [y_1, y_2, \dots, y_n]$, here is the formulation to calculate cosine similarity from both music:

$$cos\,\theta = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

In the numerator, the calculation is done by calculating dot product of both vectors and in the denominator, the calculation is done by calculating the vector lengths. The obtained value of cosine similarity is between -1 to 1. By sorting the values from the largest to the smallest, the recommendations can be made by choosing several music with the largest cosine similarity. In this research, the number of recommendations is set to be five music.

In our experiments, the recommender system uses two methods. The first method only uses the value of cosine similarity, while the second method uses both the value of cosine similarity and information of music genre. The music which selected by the user is used as the basis music for recommendations. The features of basis music are extraction vector is obtained from the best genre prediction model in previous step. Next, the values of cosine

similarity are sorted from the largest to the smallest value. Finally, the first five music with the greatest value is used as recommendations.

Our music recommender application is developed based on Python programming language and using several key libraries, that is: Tensorflow, Keras, Librosa and Kivy. To get recommendations from the music that is playing, user can click the corresponding three-button button to the right of the music title and switch to the "Get relevant" button. The results of recommendations are shown in Figure 5 below.

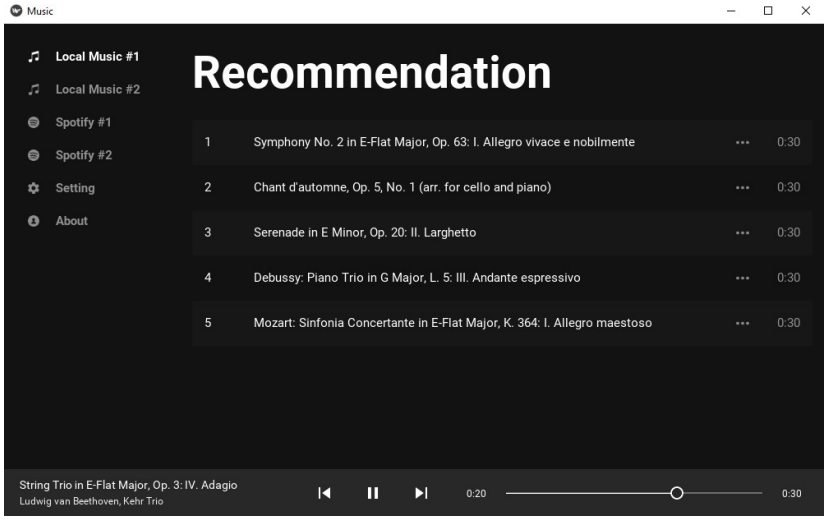

Fig. 5. Music recommender application

5. Experiment Results

5.1. Evaluation of Music Genre Classification Models

To evaluate the performance of binary classification, we usually use Receiver Operator Characteristics (ROC). However, when dealing with imbalance datasets or skewed data, precision-recall (PR) will provide more information about algorithm performance [15]. For first evaluation, the precision and recall in testing data of both CNNs and CRNNs model are shown in Table 2. The following are the results of music genre classification when classification threshold is 0.5.

Table 2. Precision and Recall of Testing Data.

Genre	Precision		Recall	
	CNN	CRNN	CNN	CRNN
Classical	0.632	0.804	0.800	0.750
Electronic	0.621	0.729	0.640	0.700
Folk	0.647	0.629	0.750	0.830
Hip-Hop	0.902	0.845	0.740	0.870
Instrumental	0.537	0.532	0.790	0.750
Jazz	0.835	0.833	0.810	0.800
Rock	0.635	0.616	0.940	0.930
Average	**0.687**	**0.712**	**0.781**	**0.804**

From Table 2, precision values explain how well the model predict the true music genre. For the genre of classical and electronic music the CRNNs method is superior to the CNNs method with differences in values of 0.172 and 0.108, while for the genre of hip-hop music, the CNNs method is better than the CRNNs method with a difference in

value of 0.057. For other music genres the CNNs method is slightly better than the CRNNs. While recall values explain when given the true music genre, then how often when the model evokes the true conditions. The CRNNs method is superior to the genre of electronic, folk, and hip-hop with a difference in values of 0.06, 0.08, and 0.13 respectively. While the CNNs method is superior to classical and instrumental music genres with a difference of 0.05 and 0.04. For other music genres the CNNs method is slightly better with a difference in the value of 0.01. Table 2 shows that CRNNs in general give better performance, indicating the effectiveness of its hybrid structure to extract the music features.

Table 3 describe the experiment results for the true positive rate and the false positive rate when using classification threshold is 0.5. By definition, true positive rate is the same to the recall value. While false positive rate explains when the number of negative music genres wrongly categorized as positive. The CRNNs method is better in classical, electronic, and instrumental genres. As for the genre of folk, hip-hop, and rock the CNNs method is better. For the genre of jazz music, the values of the two methods are the same. Table 3 one more shows that CRNNs in general give better performance than CNNs method.

Table 3. True Positive Rate dan False Positive Rate of Testing Data.

Genre	True Positive Rate		False Positive Rate	
	CNN	CRNN	CNN	CRNN
Classical	0.800	0.750	0.047	0.018
Electronic	0.640	0.700	0.070	0.046
Folk	0.750	0.830	0.073	0.088
Hip-Hop	0.740	0.870	0.014	0.029
Instrumental	0.790	0.750	0.121	0.118
Jazz	0.810	0.800	0.029	0.029
Rock	0.940	0.930	0.096	0.104
Average	**0.781**	**0.804**	**0.064**	**0.061**

To summarize the classifier performance, we use ROC curve at all possible thresholds. The ROC curve in Table 4 is created with true positive rate and false positive rate. Furthermore, F1 scores can be used to see the performance of the model by combining the precision and recall values. Table 5 shows the F1 scores of each music genre. From both Table 4 and 5, the CRNNs model that considers both the frequency features and time sequence patterns has overall better performance. Although for several genres, its performances are similar to the CNNs method.

Table 4. ROC Curve of Testing Data.

Method	AUC ROC							
	Classical	***Electronic***	***Folk***	***Hip-Hop***	***Inst***	***Jazz***	***Rock***	***Avg***
CNN	0.95	0.90	0.93	0.95	0.91	0.94	0.97	**0.935**
CRNN	0.96	0.95	0.93	0.97	0.91	0.92	0.97	**0.944**

Table 5. F1 Score of Testing Data.

Method	F1 Score							
	Classical	***Electronic***	***Folk***	***Hip-Hop***	***Inst***	***Jazz***	***Rock***	***Avg***
CNN	0.706	0.631	0.695	0.813	0.640	0.822	0.758	**0.723**
CRNN	0.780	0.714	0.716	0.857	0.622	0.816	0.741	**0.749**

5.2. Evaluation of Music Recommender System

To evaluate our music recommender system, we conduct a listening experiment [11] with 30 participants. The experiments are based on user responses to given music recommendations. In our experiments, the recommender system implements two methods. The first method only uses the value of cosine similarity, while the second method

uses both the value of cosine similarity and information of music genre. The procedure of experiment is as follow: the participants chose five their preferred music, then our music recommender system gives 5 recommendations respectively. Next, the participants evaluated the music recommender system by telling like or dislike to each recommendation. Finally, we performed statistics procedure to see the significance of the differences in respondents' responses to the two methods.

For the first method, the average of user's like with the music recommendations is 3,327 (in scale 5) and for the second method is 3,557. The average value of the second method is slightly better than the first method. When we viewed by the user's response based on the sequence of music recommendations, the first rank recommendation is not necessarily favoured by the user as a recommendation even though the music in other ranks is preferred. Therefore, we concluded that five music recommendations are independent and do not affect each other. To see the significance level between the average value of each method, the t-test calculation is used. By using alpha=0.05, here are the results of the t-score calculation:

$$H_0: \mu_1 - \mu_2 = 0 \tag{2}$$

$$H_1: \mu_1 - \mu_2 \neq 0 \tag{3}$$

$$\bar{x}_1 = 3.327, \qquad \bar{x}_2 = 3.557 \tag{4}$$

$$s_1^2 = 0.2895, \qquad s_2^2 = 0.2269 \tag{5}$$

$$t\ stat = -4.743, \qquad t\ Critical\ two\ tail = 2.037 \tag{6}$$

From the above calculation, it can be seen that -4.743 <-2.037, the value of t stat is smaller than the critical value. Therefore, we can conclude that the second method which considering music genre information is better than the first method. In addition, the calculation of feature vectors for the recommendation system takes ± 14 minutes for 726 audio data or about 1.2 seconds to create feature vector on one music. Furthermore, the recommendation process takes about 5 seconds for 726 music. There is significant difference in speed for both methods.

6. Conclusions

The following are our conclusions based on experiment results. First, music recommender system should consider the music genre information to increase the quality of music recommendations. Second, CRNNs that considers both the frequency features and time sequence patterns has overall better performance. It indicates the effectiveness of its hybrid structure to extract the music features. Based on our analyses, we can suggest for future research to add other music features in order to improve the accuracy of the recommender system, such as using tempo gram for capturing local tempo at a certain time.

Acknowledgments

This work is supported by Bina Nusantara University, as a part of Penelitian Terapan BINUS Grant 2019.

References

1. Stafford SA. Music in the Digital Age: The Emergence of Digital Music and Its Repercussions on the Music Industry. The Elon Journal of Undergraduate Research in Communications Vol. 1 No. 2. 2010.
2. Aggarwal CC. Recommender Systems: The Textbook. 1st ed.: Springer; 2016.
3. Oord Avd, Dieleman S, Schrauwen B. Deep content-based music recommendation. Advances in Neural Information Processing Systems 26 (NIPS 2013). 2013.
4. O'Bryant J. A survey of music recommendation and possible improvements. In ; 2017.

5. Choi K, Fazekas G, Sandler M. Automatic Tagging using Deep Convolutional Neural Network. arXiv eprints arXiv:1606.00298. 2016.
6. Choi K, Fazekas G, Sandler M. Convolutional Recurrent Neural Networks For Music Classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016.
7. Ricci F, Rokach L, Shapira B. Recommender Systems Handbook: Springer Science; 2010.
8. Arditi D. Digital Subscriptions: The Unending Consumption of Music in the Digital Era. Popular Music and Society. 2017.
9. Song Y, Pearce M, Dixon S. A Survey of Music Recommendation Systems and Future Perspectives. In The 9th International Symposium on Computer Music Modeling and Retrieval (CMMR); london. p. 395-410.
10. Bu J, Tan S, Chen C, Wang C, Wu H, Zhang L, et al. Music recommendation by unified hypergraph: combining social media information and music content. In MM'10 Proceedings of the 18th ACM international conference on Multimedia; 2010. p. 391-400.
11. Bogdanov D, Herrera P. How Much Metadata Do We Need in Music Recommendation? A Subjective Evaluation Using Preference Sets. In 12th International Society for Music Information Retrieval Conference (ISMIR 2011); 2011. p. 97-102.
12. Ekstrand MD, Riedl JT, Konstan JA. Collaborative Filtering Recommender Systems. Foundations and Trends in Human–Computer Interaction. 2010; 4(2): p. 81–173.
13. Wang X, Wang Y. Improving Content-based and Hybrid Music Recommendation using Deep Learning. In Proceedings of the 22nd ACM international conference on Multimedia; 2014; Orlando. p. 627-636.
14. Defferrard M, Benz K, Vandergheynst P, Xavier Bresson. FMA: A Dataset For Music Analysis. In 18th International Society for Music Information Retrieval Conference; 2017.
15. Davis J, Goadrich M. The Relationship Between Precision-Recall and ROC Curves. In ICML'06 Proceedings of the 23rd international conference on Machine learning; 2006. p. 233-240.