Interspeech 2022
18-22 September 2022, Incheon, Korea

Oktoechos Classification in Liturgical Music Using SBU-LSTM/GRU

Rajeev Rajan, Ananya Ayasi

College of Engineering Trivandrum, Thiruvananthapuram
APJ Abdul Kalam Technological University, Kerala, India
rajeev@cet.ac.in, ananya.ayasi@gmail.com

Abstract

A distinguishing feature of the music repertoire of the Syrian tradition is the system of classifying melodies into eight tunes, called 'oktoēchos'. It inspired many traditions, such as Greek and Indian liturgical music. In oktoēchos tradition, liturgical hymns are sung in eight modes or eight colours (known as eight 'niram', regionally). In this paper, the automatic oktoēchos genre classification is addressed using musical texture features (MTF), i-vectors and Mel-spectrograms through stacked bidirectional and unidirectional long-short term memory (SBU-LSTM) and GRU (SB-GRU) architectures. The performance of the proposed approaches is evaluated using a newly created corpus of liturgical music in Malayalam. SBU-LSTM and SB-GRU frameworks report average classification accuracy of 88.19% and 87.50%, with a significant margin over other frameworks. The experiments demonstrate the potential of stacked architectures in learning temporal information from MTF for the proposed task.

Index Terms: liturgy, colour, timbral, stacked architecture, Bidirectional..

1. Introduction

Oktoēchos classification in liturgical music (music being used in worship) is addressed using deep learning frameworks in the paper. The system of singing the same lyrics in eight different melodies in an eight-week cycle is referred to as the'oktoēchos' [1]. Indian Orthodox church has imbibed this music system into its liturgy through its relationship with the Orthodox church in Syria (Antiochian liturgy). A distinguishing feature of the music repertoire of the Syrian tradition is the system of classifying melodies into eight tunes [1]. This musical tradition has been transferred to Indian liturgical music through centuries with hymns in the Malayalam language. Most of the hymns used for various feasts and occasions are musically composed under eight tunes.

1.1. Oktoēchos

Western Syriac music is based on the tradition prescribed in Syriac liturgical book, 'Bethgazzo', which contains a collection of Syriac chants and melodies. In oktoēchos, liturgical hymns are sung in eight modes (same lyrics but different modes), similar to the Greek liturgy. They are a group of eight adaptable melody types, known as eight 'colours' or 'niram' [2]. In their musical structure, they are very much related to the rāga [1] system of Indian music. However, in the scalar principle of the music, they are not at all equal [2]. Oktoēchos is considered as a cyclic system because it is performed in a cycle of eight weeks with two colours in a week, in the order 1-5; 2-6; 3-7; 4-8 then in reverse 5-1; 6-2; 7-3; 8-4.

[1] $Rāga$ is the fundamental melodic framework for both Carnatic and Hindusthani traditions

1.2. Related Works

Although there has been significant work in music genre classification, the proposed task of liturgical music genre classification is the first of its kind. A model capable of learning distinctive rhythmic structures for genre classification is proposed in [3]. In contrast with standard approaches, model-based distances between time series can take into account the structure of the songs by modelling the dynamics of the parameter sequence [4]. Spectrograms are also employed for music genre classification [5]. The i-vector based statistical feature is explored in [6]. In [7], a multi-step LSTM is proposed for recognizing music genres. The divide-and-conquer approach to classify ten genres of music in their work resulted in an accuracy of 50.00%. A deep bidirectional transformers-based masked predictive encoder approach for genre classification is attempted in [8]. Recent approach [9] uses a transformer classifier to analyze the relationship between different audio frames for the task of genre classification. Stacked architectures have been attempted for forecasting network-wide traffic with missing data in [10]. Regarding multimodal approaches found in the literature, most of them combine audio, and song lyrics [11, 12] through a fusion framework. The proposed task is similar to genre classification, but sharing the textual content across modes is one of the specific traits of the oktoēchos genre system. The proposed work explores the potential of stacked neural network architectures to capture the long-range dependency in learning temporal patterns. Backward dependencies by combining LSTM and BiLSTM enhances the feature learning process in the proposed framework.

The rest of the paper is organized as follows; Section 2 describes the proposed system. The performance evaluation is discussed in Section 3 followed by the analysis of results in Section 4. Finally, the paper is concluded in Section 5.

2. System Description

2.1. Feature Extraction

It has already been proven that timbral and rhythmic features are useful in genre classification task [13]. Timbral and rhythmic features have been computed as musical texture features in the front-end. Mel-frequency cepstral features (MFCC) and low-level timbral feature-set (T_{LF}), are computed as timbral feature-set. MFCCs are widely employed in numerous perceptually motivated audio classification tasks [14], as predictors of perceived similarity of timbre [15] Spectral centroid, spectral roll-off, spectral flux, and spectral entropy [16] are extracted as low-level timbral feature set. Besides, tempo, pulse clarity, event density [17] are also computed as rhythmic cues (R_F).

10.21437/Interspeech.2022-136

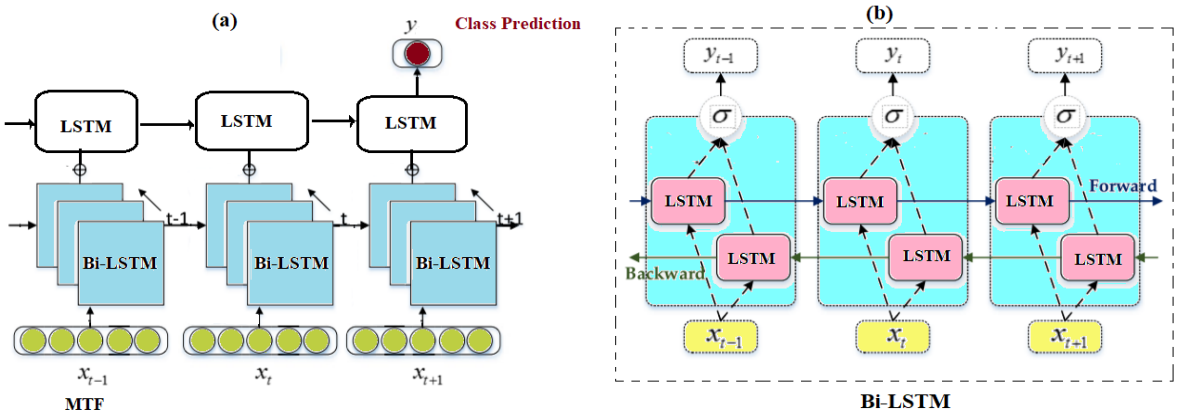

Figure 1: *(a) Stacked Bi-directional Unidirectional LSTM architecture. (b) Single Bi-directional LSTM unit of Bi-LSTM block in (a) [10]*

Event density represents the number of events per unit time in the music piece [17]. This feature-set is computed using MIR-Toolbox.

Motivated by the experimental works in [18, 19], i-vector-DNN framework has also been experimented here. In i-vector system [20], the high dimensional GMM super vector space (generated from concatenating the mean values of GMM) is mapped to a low dimensional space called total variability space. The target utterance GMM is adapted from a universal background model (UBM) using Eigen voice adaptation. The target GMM super vector can be viewed as a shifted version of UBM. Formally, a target GMM super vector M can be written as:

$$M = m + Tw \tag{1}$$

where m represents the UBM super vector, T is a low dimensional rectangular total variability (TV) matrix, and w is termed as i-vector. Using training data, the UBM and TV matrix is modelled by expectation maximization. 100 dimensional i-vectors (i_{MFCC}) are computed per song from MFCC using Alize tool kit [21]. Since Mel-spectrogram has already been utilized well for music genre classification tasks [22, 23], we experimented with mel-spectrogram-CNN framework also. Mel-spectrogram is computed with frame size of 40 ms and hop size of 10 ms using 128 bins.

2.2. Stacked Bidirectional and Unidirectional LSTM and GRU

To the best of our knowledge, the stacked network architectures introduced in the paper for genre classification is the first of its kind. LSTM architectures with several hidden layers can progressively build up a higher level of representations of sequence data, and thus, work more effectively [24, 25]. In a stacked multi-layer LSTM architecture, the output of a hidden layer will be fed as the input into the subsequent hidden layer. In this study, we employ a deep architecture named stacked bidirectional and unidirectional LSTM network (SBU-LSTM) to classify octoechos music from musical texture features. SBU-LSTM contains a Bi-LSTM layer as the first feature-learning layer and a LSTM layer as the last layer. Multiple LSTM or Bi-LSTM layers as middle layers are optional. Bi-LSTMs explore both forward and backward dependencies on feature vectors for the classification task. The proposed architecture for SBU-LSTM and Bi-LSTM are shown in Fig 1(a) and (b), respectively. The temporal dependencies among the feature vectors are captured during the learning process. Since Bi-LSTM contains more learnable parameters, the architecture of stacked Bi-LSTMs has the potential to perform better in various sequence data processing applications. The SBU-LSTM architecture employed for the experiment is shown in Table 1.

Table 1: *SBU-LSTM architecture for the experiment*

Sl no.	Output Size	Description
1	(46,512)	BiDLSTM, 512 hidden units
2	(46, 512)	Drop out (0.25)
3	(46,1024)	BiDLSTM, 1024 hidden units
4	(46, 1024)	Drop out (0.25)
5	(1024)	LSTM, 1024 hidden units
6	(8)	Dense (8 hidden units)

Table 2: *SB-GRU architecture for the experiment*

Sl no.	Output Size	Description
1	(46,256)	BiGRU, 256 hidden units
2	(46, 256)	Drop out (0.25)
3	(512)	BiGRU, 512 hidden units
4	(512)	Drop-out (0.25)
5	(8)	Dense

The forward layer output $\overrightarrow{h_t}$ is iteratively calculated based on positive ordered inputs $[x_1, x_2 \dots x_T]$ and the backward layer output, $\overleftarrow{h_t}$ is iteratively calculated using the reversed ordered inputs. The Bi-LSTM layer generates output element y_t at each step t based on the combination of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ by using the following equation

$$y_t = \oplus(\overrightarrow{h_t}, \overleftarrow{h_t}) \tag{2}$$

where $\oplus$ can be chosen as average, summation, multiply or concatenate functions. The output of a Bi-LSTM layer can be represented by a vector $[y_1, y_2 \dots y_T]$. The class prediction is done from a dense layer following the LSTM layer.

Similar to the stacked Bi-LSTM, we also experimented with stacked bi-directional GRU (SB-GRU). However, compared with Bi-GRU, the SB-GRU has more hidden layers,

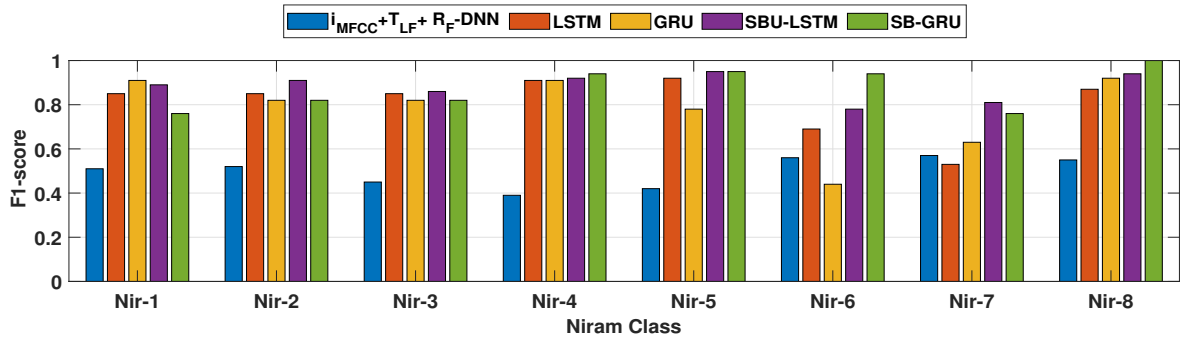

Figure 2: *Class-wise F1-score for various phases.*

which can capture long-term memories for the prediction of the class in the proposed framework. The SB-GRU architecture employed for the experiment is given in Table 2.

2.3. Baseline classifiers

The performance of the proposed architecture is compared with that of DNN, CNN, *vanilla*-LSTM and GRU. DNN is implemented with six hidden layers, which uses 64, 128, 256, 512, 1024, 2048 nodes in successive layers with a dropout of 0.25. The network is trained with a batch size of 32 for 150 epochs by AdaMax algorithm. Mel-spectrogram-CNN [26] and *vanilla*-LSTM [7] have already been used for genre classification. CNN has six convolution layers, followed by max-pooling. We use filters with a very small 3×3 receptive fields, with a fixed stride of one and increase the number of filters by a factor of 2 after every layer. Global max-pooling is done, which is then fed to a fully connected layer. The training is done with 100 epochs by optimizing the categorical cross-entropy between predictions and targets using Adam optimizer, with a learning rate of 0.001. *vanilla*-LSTM and GRU are implemented with two LSTM and GRU layers with a drop out of 0.25, respectively.

3. Performance Evaluation

A database was created in a studio environment, and it consists of eight nirams (colours), with 384 audio tracks with a duration of 30 to 45 s per file. A total of 15 professional singers in the age group 12 to 50 participated in the data recording, and the whole session was recorded at 44.1kHz. All the singers were very much familiar with the singing modes in 'oktoec̄hos'. Malayalam hymns were collected from the liturgical book of the Indian Orthodox church. 60% files of the dataset are used for training, 5% is used for validation and the rest for testing.

Table 3: *Overall classification accuracy. The best system is highlighted in red and RNN variants are entered as bold entries.*

Sl.No	Feature	Approach	Accr.(%)
1	$i_{MFCC} + T_{LF} + R_F$	DNN	50.00
2	Mel-spectrogram (MS)	CNN [26]	52.60
3	**MFCC +** T_{LF}**+** R_F (MTF)	**LSTM [7]**	**81.94**
4	**MFCC +** T_{LF}**+** R_F (MTF)	**GRU**	**79.16**
5	**MFCC +** T_{LF}**+** R_F (MTF)	**SBU-LSTM**	**88.19**
6	**MFCC +** T_{LF}**+** R_F (MTF)	**SB-GRU**	**87.50**

MFCCs (39 dim comprising 13 dim MFCC, it's delta and

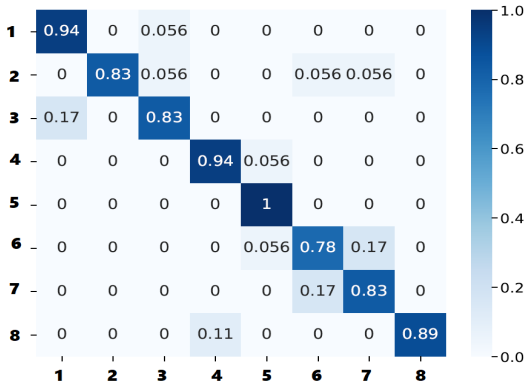

Figure 3: *Normalized confusion matrix for MTF-SBU-LSTM*

delta-delta features), timbral (T_{LF}, four dim) rhythmic (R_F, 3 dim) are frame-wise computed with a frame width of 40 ms and hop size of 10 ms and fused to obtain 46-dim MTF. In the i-vector experimental phase, 100-dim i-vectors are computed using 128 mixture GMM from MFCC using Alize toolkit [21]. UBM is trained using features derived from the auxiliary database comprising audio file other than the files in the corpus. An Auxiliary database comprising 300 audio files (duration 25-35ms) of the liturgical music category is prepared in a studio environment. The songs from the training data are used for modelling the total variability matrix T by Eigen voice adaptation. In the fusion scheme, track level aggregated timbral (T_{LF}) and rhythmic (R_F) features are concatenated with track-level computed i-vectors. SB-LSTM and SB-GRU models are trained by minimizing categorical cross-entropy using the Adam optimizer. Training is done for 100 epochs with an initial learning rate of 0.001. Bi-LSTM/GRU layer generates output element y_t using concatenation.

4. Results and Analysis

The results are tabulated in Table 3. As per the table, the average classification accuracy of 50.00%, 52.60%, 81.94%, 79.16%, 88.19% and 87.50% are reported for i-vector-DNN framework, Mel-spectrogram-CNN, LSTM, GRU, SBU-LSTM, SB-GRU, respectively. Among various schemes, SBU-LSTM reports best performance with 7% improvement over *vanilla*-LSTM approach. Even though the SB-GRU model reports an accuracy of 87.50%, SB-GRU followed by unidirectional GRU resulted in a slightly less accuracy (86.16%). The superior performance

Table 4: *Metrics for experiments. MS and MTF denote Mel-spectrogram and MFCC+T_{LF}+R_F, respectively*

Colour	i_{MFCC}+T_{LF}+R_F-DNN			MS-CNN			MTF-LSTM			MTF-GRU			MTF-SBU-LSTM			MTF-SB-GRU		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Nir-1	0.48	0.56	0.51	0.42	0.50	0.45	0.93	0.78	**0.85**	1.00	0.83	**0.91**	0.85	0.94	**0.89**	1.00	0.61	**0.76**
Nir-2	0.46	0.61	0.52	0.52	0.68	0.59	0.77	0.94	**0.85**	0.80	0.84	**0.82**	1.00	0.83	**0.91**	0.76	0.89	**0.82**
Nir-3	0.54	0.39	0.45	0.70	0.35	0.47	0.77	0.94	**0.85**	0.80	0.84	**0.82**	0.88	0.83	**0.86**	0.69	1.00	**0.82**
Nir-4	0.46	0.33	0.39	0.69	0.58	0.63	0.94	0.89	**0.91**	1.00	0.83	**0.91**	0.89	0.94	**0.92**	1.00	0.89	**0.94**
Nir-5	0.40	0.44	0.42	0.54	0.68	0.60	0.86	1.00	**0.92**	0.64	1.00	**0.78**	0.90	1.00	**0.95**	0.90	1.00	**0.95**
Nir-6	0.52	0.61	0.56	0.55	0.63	0.59	0.79	0.61	**0.69**	0.67	0.33	**0.44**	0.78	0.78	**0.78**	1.00	0.89	**0.94**
Nir-7	0.59	0.56	0.57	0.47	0.42	0.44	0.67	0.44	**0.53**	0.65	0.61	**0.63**	0.79	0.83	**0.81**	0.81	0.72	**0.76**
Nir-8	0.60	0.50	0.55	0.44	0.37	0.40	0.81	0.94	**0.87**	0.86	1.00	**0.92**	1.00	0.89	**0.94**	1.00	1.00	**1.00**
Macro	0.50	0.50	**0.50**	0.54	0.53	**0.52**	0.82	0.82	**0.81**	0.80	0.79	**0.78**	0.89	0.88	**0.88**	0.90	0.88	**0.87**

of stacked architecture shows the learning capability through forward and backward dependencies on temporal data. It is worth noting that the stacked architectures outperform other approaches with significant margins.

The important music elements can be captured well by i-vectors [18] and may potentially be benefited for music genre classification. A possible cause of the less accuracy in the given i-vector experimental set-up may potentially be due to the inability to capture the rhythmic-temporal dynamics well with the given UBM framework. Besides, aggregation of musical texture features to track-level features might have deteriorated the performance. For the CNN framework, the result improved as the number of layers increased up to six and then saturated due to overfitting. As n increases, the model grows in-depth, and the upper layers find efficient feature representations that are invariant to small perturbations leading to better model generalization. The authors [27] emphasize the need for data augmentation schemes, in the visual representation-based approaches for the genre classification task. CNN needs a large size of data to achieve better results since it is not successful enough for less data [28]. By comparing i-vector modelling, time-frequency processing and temporal processing, it is reasonable to say that time pattern capturing scheme has the potential to recover more relevant information from temporal embedded musical traits [29]. The stacked approach is promising for the given task, improving when the dynamics are not taken into account.

Class-wise classification accuracy of all nirams are greater than 78% for SBU-LSTM. Nirams 1, 4 and 5 report accuracy greater than 90%. Besides, class-wise F1-score can be examined from the bar plot given in Fig. 2. The significant improvement in accuracy for stacked architectures over *vanilla* counterparts can be observed in the plot. The normalized confusion matrix of best performing SBU-LSTM is given in Fig. 3. The performance metrics precision, recall and F1 score for all the six approaches are given in Table 4. From Table 4, it can be seen that the average F1 measure of 0.50, 0.52, 0.81, 0.78, 0.88 and 0.87 are reported for i-vector-DNN, CNN, LSTM, GRU, SBU-LSTM and SB-GRU, respectively. The higher values of the stacked approach show the significance of the adopted scheme.

Fig.4 visualizes the output vectors produced by the snippets for the last dense layer of the trained SBU-LSTM using t-SNE. Good clustering (as represented with colour) and a general separation of different classes for SBU-LSTM can be observed. Furthermore, it is worth noting that stacked recurrent neural network variants on MTF perform better than the CNN-time-frequency representation framework, without any data augmentation schemes. To summarize, the results show the efficacy

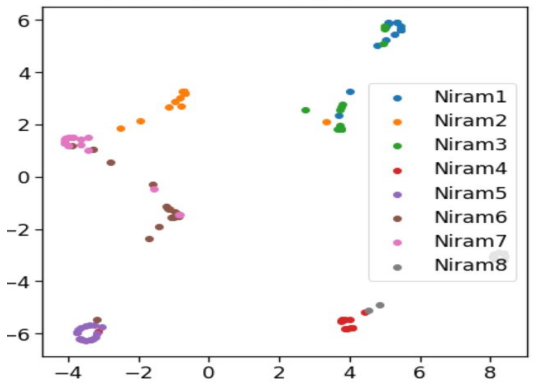

Figure 4: *Visualization of the output vectors produced by the snippets for the last dense layer of the trained SBU-LSTM using t-SNE*

of SBU-LSTM and SB-GRU in learning temporal patterns as compared to *vanilla*-LSTM/GRU.

5. Conclusion

Oktoēchos classification is addressed using stacked LSTM/GRU. The proposed approaches are evaluated using a liturgical music corpus. The evaluation shows the potential of MTF-SBU-LSTM framework by reporting 88.19% average classification accuracy. Since the Greek liturgy and Gregorian chant also share similar musical traits with Syrian tradition, the musicological insights observed can potentially be applied to those traditions as well.

6. References

[1] J. Palackal, "Oktoechos of the syrian orthodox churches in south india," *Ethnomusicology*, vol. 48, pp. 229–250, 2004.

[2] P. Vysanethu, "Musicality makes the malankara liturgy musical (moran etho 2)," *St.Ephrem Ecumenical Research Institute, Kottayam, Kerala, India*, 2004.

[3] M. Pesek, A. Leonardis, and M. Marolt, "An analysis of rhythmic patterns with unsupervised learning," *Applied Science*, pp. 1–22, 2020.

[4] D. Garcia-Garcia, J. Arenas-Garcia, E. Parrado-Hernandez, and F. Diaz-de Maria, "Music genre classification using the temporal structure of songs," *in Proc. of IEEE Int. Workshop on Machine Learning for Signal Processing*, 2010.

[5] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," *in Proc. of IEEE*

Int. Conference on Acoustics, Speech and Signal Processing, pp. 2392–2396, 2017.

[6] J. Dai, W. Xue, and W. Liu, "Multilingual i-Vector Based Statistical Modeling for Music Genre Classification," in *Proc. Interspeech*, 2017, pp. 459–463.

[7] C. Tang, K. L. Chui, Y. K. Yu, Z. Zeng, and K. H. Wong, "Music genre classification using a hierarchical long short term memory (lstm) model," in *in Proc. of Int. Workshop on Pattern Recognition*, 2018.

[8] L. Qiu, S. Li, and Y. Sung, "DBTMPE: Deep bidirectional transformers-based masked predictive encoder approach for music genre classification," *Mathematics*, vol. 9, p. 530, 03 2021.

[9] Y. Zhuang, Y. Chen, and J. Zheng, "Music genre classification with transformer classifier," *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, pp. 155–159, 2020.

[10] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values," *Transportation Research Part C: Emerging Technologies*, vol. 118, pp. 1–14, 09 2020.

[11] S. Oramas, F. Barbieri, O. Nieto, and X. Serra, "Multimodal deep learning for music genre classification," *Trans. Int. Society for Music Information Retrieval*, vol. 1, pp. 4–21, 2018.

[12] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal music mood classification using audio and lyrics," *in Proc. of Seventh IEEE Int. Conference on Machine Learning and Applications*, pp. 688–693, 2008.

[13] B. K. Baniya, D. Ghimire, and J. Lee, "Automatic music genre classification using timbral texture and rhythmic content features," *Proc. of 17th Int. Conference on Advanced Communication Technology*, pp. 434–443, 2015.

[14] J. Seppanan, "Computational models for musical meter recognition," Masters Thesis, Tampere University of Technology, Department of Information Technology, 2015.

[15] G. Richard, S. Sundaram, and S. Narayanan, "An overview on perceptually motivated audio indexing and classification," *in Proc. of the IEEE*, vol. 101, no. 9, pp. 1939–1954, 2013.

[16] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," *in Proc. of the 26th Annual Int. ACM Conference on Research and development in information retrieval*, pp. 282–289, 2003.

[17] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari, "Multi-feature modeling of pulse clarity: Design, validation and optimization." *in Proc. of the 9th Int. Conference on Music Information Retrieval*, pp. 1–5, 2008.

[18] J. Dai, W. Xue, and W. Liu, "Multilingual i-vector based statistical modeling for mmusic genre classification," *in Proc. of Interspeech*, pp. 459–463, 2017.

[19] J. Zhong, W. Hu, F. Soong, and H. Meng, "DNN i-vector speaker verification with short, text-constrained test utterances," *in Proc. of Interspeech*, pp. 1507–1511, 2017.

[20] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[21] J.-F. Bonastre, F. Wils, and S. Meignier, "Aliźe, a free toolkit for speaker recognition," *in Proc. of Interspeech*, vol. 1, pp. 737–740, 01 2005.

[22] M. Sukhavasi and S. Adappa, "Music theme recognition using CNN and self-attention," *preprint arXiv:1911.07041*, 2019.

[23] D. Ghosal and M. H. Kolekar, "Music genre recognition using deep neural networks and transfer learning." *in Proc. of Interspeech*, pp. 2087–2091, 2018.

[24] J. N. Graves, A. and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM." *in Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 273–278, 2013.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[26] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Convolutional neural networks approach for music genre classification," *in Proc. of Int. Symposium on Computer, Consumer and Control*, pp. 399–403, 2020.

[27] C. Liua, L. Fengb, G. Liuc, H. Wangd, and S. Liub, "Bottom-up broadcast neural network for music genre classification," *Pattern Recognition Letters*, pp. 1–7, 2019.

[28] M. Kaya and S. H. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, pp. 1–26, 2019.

[29] J. Pons and X. Serra, "Randomly weighted cnns for (music) audio classification," *in Proc. of IEEE Int. Conference on Acoustics, Speech and Signal Processing*, pp. 336–340, 2019.