I.J. Information Technology and Computer Science, **2021, 2, 1-14**
Published Online April 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2021.02.01

Comparative Analysis of Three Improved Deep Learning Architectures for Music Genre Classification

Quazi Ghulam Rafi, Mohammed Noman, Sadia Zahin Prodhan, Sabrina Alam
American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: {quazirafi1, mohammednomanno1, sadiaprodhan, sabrina.alam.397}@gmail.com

Dip Nandi
American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: dip.nandi@aiub.edu

Received: 13 September 2020; Accepted: 05 November 2020; Published: 08 April 2021

Abstract: Among the many music information retrieval (MIR) tasks, music genre classification is noteworthy. The categorization of music into different groups that came to existence through a complex interplay of cultures, musicians, and various market forces to characterize similarities between compositions and organize collections is known as a music genre. The past researchers extracted various hand-crafted features and developed classifiers based on them. But the major drawback of this approach was the requirement of field expertise. However, in recent times researchers, because of the remarkable classification accuracy of deep learning models, have used similar models for MIR tasks. Convolutional Neural Net- work (CNN), Recurrent Neural Network (RNN), and the hybrid model, Convolutional - Recurrent Neural Network (CRNN), are such prominently used deep learning models for music genre classification along with other MIR tasks and various architectures of these models have achieved state-of-the-art results. In this study, we review and discuss three such architectures of deep learning models, already used for music genre classification of music tracks of length of 29-30 seconds. In particular, we analyze improved CNN, RNN, and CRNN architectures named Bottom-up Broadcast Neural Network (BBNN) [1], Independent Recurrent Neural Network (IndRNN) [2] and CRNN in Time and Frequency dimensions (CRNN- TF) [3] respectively, almost all of the architectures achieved the highest classification accuracy among the variants of their base deep learning model. Hence, this study holds a comparative analysis of the three most impressive architectural variants of the main deep learning models that are prominently used to classify music genre and presents the three architecture, hence the models (CNN, RNN, and CRNN) in one study. We also propose two ways that can improve the performances of the RNN (IndRNN) and CRNN (CRNN-TF) architectures.

Index Terms: Music information retrieval, music genre classification, deep learning, Convolutional Neural Network, Recurrent Neural Network, Convolutional - Recurrent Neural Network.

1. Introduction

In recent times, with the rise of popularity of music streaming services such as Spotify, Apple Music, Amazon Music, the necessity of structuring and organization of music is of utmost importance for applications, namely, music auto-tagging or music recommendation. These streaming services have resulted in an exponential increase of contents, and the categorization or organization of such a big library of contents is a daunting task. Genre is one of the ways to organize and classify music content. Hence, it is vital to develop a robust and accurate music genre classification system to help in the automatic organization of these content. However, music genre classification remains a nontrivial task as the genres are loosely defined. Despite the wide use of terms such as rock, pop, or jazz, there remains no clear description to distinguish between them. Such challenges and the need for a music genre classification system has created opportunities for re- searchers worldwide and has attracted many researchers in recent times.

Deep learning architectures, namely CNN, RNN, and CRNN, are in use for music genre classification, but there remains a lack of comparisons of the three architectures for the purpose. Furthermore, the architectures are improving to overcome their shortcomings. Therefore, it is vital to have a comparison of the three architectures, CNN, RNN, and CRNN to classify the music genre. For this reason, we read and analyzed the previous works of researchers who previously used the deep learning architectures for the purpose and sorted the improved architectures of CNN, RNN, and CRNN. We sorted the architectures according to the classification accuracy for music genre classification of relatively longer music tracks (29-30 seconds). Among the sorted architectures we selected BBNN, IndRNN and CRNN-TF since as far as our knowledge and findings BBNN and CRNN-TF achieved the highest classification accuracy among the

architectures of CNN and CRNN, where as IndRNN achieved remarkable classification accuracy with immense efficiency among the RNN architectures. All the three architectures achieved classification accuracy of more than 90%. We then discussed and analyzed the selected architectures and proposed two ways to improve the RNN and CRNN architectures.

In this study, we reviewed and discussed BBNN [1], IndRNN [2], and CRNN-TF [3] three improved architectures of CNN, RNN, and CRNN respectively, that previously used to classify genres of relatively longer clips ($29-30$ seconds) and achieved remarkable results. It is to be noted, as we focused on a particular niche of MIR tasks, that is, on music genre classification, we focused on the use of CRNN-TF for the purpose and ignored the use of the architecture for music auto-tagging, it's transfer-ability from the work of Wang et al. [3]. In the later part of the study, we suggested two possible ways to improve the performance of IndRNN and CRNN-TF.

The organization of the rest of the study is as follows; section 2 contains the literature review, in section 3 we discuss the BBNN architecture of Liu et al., section 4 holds a discussion of the IndRNN architecture of Wu et al., and in section 5 Wang et al.'s CRNN-TF architecture is discussed. In section 6 , we analyze the outcome of the three architectures, BBNN, IndRNN, and CRNN-TF, and propose ways to improve the IndRNN and CRNN-TF architectures for better music genre classification accuracy. Finally, section 7 holds the conclusion of the study and reflects light to future work.

2. Literature Review

Classification of music genres is a multi-class classification task. In other words, into one of three or many classes (or genres), the music is classified. Such tasks involve two steps, feature extraction, followed by classification. The success of such tasks relies heavily on the extraction of relevant features. In the past, researchers have depended on hand- crafted feature extraction for MIR tasks and developed classifiers based on them. Casey et al. outlined the problems of content-based music information retrieval and explored methods using audio cues, such as query by humming, audio fingering, etc, and other cues, namely music notion and symbolic representation [4] Mermelstein used Mel-frequency cepstrum (MFCC) to measure the distance of the source if sound for speech recognition [5]. In MIR tasks information retrieval from images is also vital as information from music signal can be represented as images. Ojala et al. introduced Local Binary pattern (LBP) as an efficient texture descriptor that labeled the pixels of images by neighbor pixel thresholding and considered the result as a binary number [6]. In 2002, mixtures of Gaussians model and k-nearest neighbor (KNN) was used along with three hand-selected features (timbral texture, rhythmic content, and pitch content) for music genre classification (MGC) and achieved an accuracy of 61% , which in comparison to average human accuracy of 70% was a remarkable success [7]. However, the popularity of the use of hand-crafted feature extraction has decreased in recent times because this process imposes a significant blockade as expertise in the relevant field is required to obtain hand-crafted features. This requirement limits the generalization of MGC, as in different environments, the considered features change.

In 2011, from spectrograms or images which were generated from audio signals using short-term Fourier transform (STFT), textural features were extracted [8]. This and the rise in popularity of a parallel processing architecture named Graphics Processing Unit (GPU) made the use of deep learning models for feature extraction and classification tasks, hence MIR tasks of different music tracks feasible. Deep learning is such a technique that enables systems to learn by example and has proven to enable systems to understand complex perception tasks with maximum precision. The deep learning process includes two phases, training and inception. Labeling of large data and identification of matching characteristics takes place during the training phase. On the other hand, in the inception phase, a concluding decision is made, and new unexposed data are labeled using previously learned knowledge. For long deep learning, models are in use with great success for different Computer Vision (CV) tasks, such as image classification [9], object detection [10, 11], image caption [12], facial expression recognition [13], image recognition [14] and so on. Being inspired by the success of the application of deep learning models for various CV tasks, researchers being able to represent the audio signal as a spectrogram have applied different deep learning models such as CNN, RNN, and CRNN for various MIR tasks and have achieved state-of-the-art performance.

CNN has been widely used for various MIR tasks such as music recommendations [15], automatic tagging [16], feature learning [17], and so on. A popular approach of using CNN for MIR tasks involves using a spectrogram as an input to the CNN and extract patterns in 2D by applying convolving filter kernels. In 2010, for music genre prediction Li et al. developed a CNN using raw Mel-frequency cepstral coefficients (MFCC) as input [18]. For MGC, a CNN was used to capture temporal information, and another to capture timbral relations in the frequency domain [19]. The use of CNN for MGC has inspired many researchers such as Senac et al. [20] who used the filter dimensions of CNN in such a way so that it is interpretable time and frequency. In their experiment, they used eight features chosen along with dynamics, tim- bre, and tonality dimensions as inputs of CNN and achieved global accuracy of 89.6% against 87.8% for 513 frequency bins of a spectrogram. The experiment proved that music features are more efficient for MGC than a huge number of spectrogram frequency bins. Bahuleyan conducted a comparative study on the performance of deep learning models requiring a spectrogram as input, specifically VGG-16 (a robust CNN architecture) and machine learning classifiers that need is trained with hand-selected features for music genre classification [21]. In his experiments,

the CNN architecture outperformed the feature-engineered models. Yang et al. used a Mel-scale spectrogram as input to his proposed CNN architecture, which applied the output of duplicated convolutional layers to different pooling layers to produce information for music genre classification [22]. Though the CNN architecture obtained a remarkable accuracy of 90.7%, the performance suffered in a significant amount when it came to correct classification of the country genre, and it suggested that the use of 3 seconds of raw audio as input have caused the loss of performance. But a major setback for most of the CNN-based music classification models is the requirement of data-augmentation and large datasets for the training of the models, as the models often require to learn large parameters. A study in 2019 solved this problem to a certain extent by the introduction of a novel CNN architecture named Bottom- up Broadcast Neural Network (BBNN) [1]. The architecture was designed to handle multi-scale of audio features and preserve lower-level features, which was usually lost in the previous architectures even though it contained critical information. In BBNN low-level information was transmitted to the decision-making layers, which preserved the crucial low- level information and hence, resulted in greater classification accuracy. Furthermore, BBNN required only a few parameters to learn compared to other CNN architecture, this made a small dataset without any data-augmentation techniques adequate for the training of the BBNN.

Recurrent Neural Network (RNN) is another popular deep learning model designed to recognize patterns in data and takes time and sequence into account during the process. RNN is widely used for sequential data and can manipulate a long-term relationship that is present in the data. But, despite the capacity to manipulate long-term relationships, due to gradient vanishing and exploding problems, vanilla RNN struggles to learn long-term patterns [23]. But through the emerging of Gated Recurrent Unit (GRU) [24] and Long Short-Term Memory (LSTM) [25], the issues of vanilla RNN have been solved. Zhang et al. used GRU in the RNN layer of their system that can classify music genre in real-time by listening to music for just 0.5 seconds with an accuracy of 64% and suggested that LSTM-RNN can boost the accuracy to near 80% by treating the MFCC average and covariance as a time series with time-step 0.5 seconds [26]. In 2016 Dai et al. used LSTM-RNN to classify music into different genres [27]. In the study, LSTM-RNN was used to extract features from the scatter spectrogram of audio input. The extracted features and segment representation of the initial frame were combined to obtain the fusional segment feature, which achieved an accuracy of 89.71%. Both GRU and LSTM were used to classify music and achieved 92% and 89% accuracy on the GTZAN dataset by Jukubik [28]. GRU and LSTM solve gradient vanishing and exploding problems of vanilla RNN, but they both are still susceptible to gradient decay as they both use sigmoid and hyperbolic tangent functions. Also, the network struggles to work on a long-time scale. In other words, the network struggles to deal with long clips. Wu et al. addressed the problem with the net- work and introduced IndRNN for MGC tasks as IndRNN can learn long-term dependencies better than GRU and LSTM [2]. They adjusted the time-based back-propagation to solve the problems of vanilla RNN and used scattering transformation for data pre-processing to keep the loss of information to a minimum.

CRNN is a hybrid model that uses CNN to extract local features, and RNN acts as a temporal summarizer, that is, aggregates the features. In 2015 for the first time, this hybrid structure was proposed [29]. Choi et al. used success- fully used a CRNN architecture that was made of 2-layer GRU-RNN on top of 4-layer CNN for music tagging [30]. Later Choi's CRNN architecture was used to map the music genre, where the architecture achieved a 0.893 AUC-ROC index [31]. But in this architecture, RNN was used for the extraction of spatial dependency of music signal in its time dimension only, ignoring its frequency dimension. Wang et al. introduced CRNN-TF for MGC tasks that extracted spatial dependencies not only in the time dimension but also in frequency dimensions of music signal in multiple directions [3]. In the study, the CNN part of the architecture used 4-layer CNN the same as that used in Choi's [30] but used different convolutional and pooling operators. For the RNN part, multi-directional RNN was used to generate sequences that was later fed to a Grid LSTM to extract spatial dependency.

To the best of our knowledge, no researchers have done a comparative analysis of all of the three deep learning models, CNN, RNN, and CRNN, to classify music genre specifically. Nor, much work explored the use of deep learning models for the purpose. Scaringella et al. discussed typical techniques for extracting features, namely, timbre, melody or harmony and rhythm for the MIR tasks, three paradigms (expert systems, supervised and unsupervised clustering) for genre classification using these features, were some of the mentionable outcomes of the music genre classification con- test of MIREX 2005, and some emerging techniques of mu- sic genre classification [32]. Corrêa et al. surveyed different approaches for music genre classification that considers the symbolic representation of music data [33]. On the other hand, Fu et al. explored the audio-based music classification [34]. Although the study discusses CNN for music classification in short, but ignores the other two vital and emerging deep learning models that are robust for the purpose, that is, RNN and CRNN. Chillara et al. developed multiple models (Spectrogram based CNN models and Feature-based models) for MGC tasks and compared them [35]. The study found Spectrogram based CNN models outperform the models of the other type. Among the CNN based models (CNN, CRNN, CNN-RNN), CNN performed the best. Kumar et al. used of various machine learning algorithms to compare Fast Fourier Transform (FFT) and MFCC feature extractors for MGC tasks and later constructed LSTM-RNN based classifier using MFCC data for training [36]. The LSTM-RNN classifier achieved an accuracy of 86%. Along with a comparison of MFCC and FFT, the study provides a comparative idea of the different machine learning algorithms and shades light on the potential of deep learning algorithms such as RNN for the purpose but does not focus on the other two, CNN and CRNN.

3. Bottom-Up Broadcast Neural Network (BBNN) [1]

Liu et al. used a novel CNN architecture named Bottom- up Broadcast Neural Network (BBNN) for music genre classification. The BBNN consisted of a Broadcast Module (BM) which was made up of densely connected inception blocks that perceived the feature maps with different scales and extracted information hidden in the time-frequency of the audio signals simultaneously from different scales. Furthermore, to transform low-level information to the decision layer, BBNN interconnected the building blocks, which ensured the optimum maintenance of the low-level information that otherwise would have been lost in most other CNN architectures. Unlike other CNN architectures, the BBNN architecture had a few parameters which omitted the requirement of data-augmentation, which in turn reduced the requirement of large datasets for the training of the model.

3.1. Construction of the Used BBNN

The BM of the BBNN consisted of $L=3$ identical stacked, densely interconnected Inception blocks. Such architecture allowed each block to receive information from all the previous blocks and hence made the network less vulnerable to frequency-shifts in a spectrogram. When X_{SL} was the output, the input of the $l-th$ block, that is, $l=1,\ldots,L$,

$$X_l = f_I\left(\left[X_{SL}, X_1, \ldots, X_{l-1}\right]\right), \tag{1}$$

where, $[X_{SL}, X_1, \ldots, X_{l-1}]$ represented the summation of feature maps which the Inception blocks $0,\ldots,l-1$ had produced and f_I represented the composite function of all operation of an Inception block. Each Inception block had convolutions of $1\times1, 3\times3, 5\times5$ filter sizes with two strides. After that, 1×1 convolutions calculated the previous reductions. Before each convolution to enhance the generalization ability of the network, an extra layer was used, which comprised of Batch Normalization (BN), followed by rectified linear activation (ReLU). Each inception blocks had layers of stacked convolutions and BN with occasional max-pooling layers of 3×3 stride 2. Max-pooling was used to reduce the resolution of the grid to half. Feature maps of $k_0+k\times(l-1)$ were present in each block, where k_0 represented the number of channels in the input X_{SL} and k represented the growth rate of the BM, which was 128.

All the layers of the used BBNN had four parts, namely, the shallow feature extraction layer, BM, transition layer, and decision layer. Overall, the BBNN aimed to learn all parameters of a composite function $F(\cdot|\Theta)$, where Θ represented all the parameters. The composite function, $F(\cdot|\Theta)$ maps input X_0 to the output p, which is a genre and was represented by,

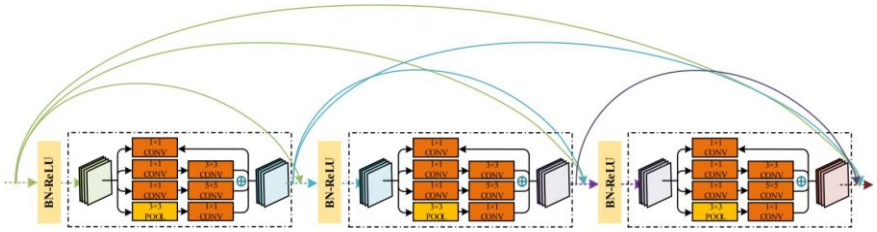

Fig.1. Used Broadcast Module [BM], sourced from [1]

$$\begin{aligned} p &= F\left(X_0|\Theta\right) \\ &= f_{DL}\left(f_{TL}\left(f_{BM}\left(f_{SL}\left(X_0|\theta_{SL}\right)|\theta_{BM}\right)|\theta_{TL}\right)|\theta_{DL}\right), \end{aligned} \tag{2}$$

where $f(\cdot)$ represented a composite function of the corresponding part of the network. In the shallow layer, within a short time, local frequency information was extracted by a small receptive field followed by a BN, and ReLU functions that activated the local features. An added max-pooling operation filtered the dominant frequency of Mel-spectrogram and to enable the architecture to achieve some invariance to translation. Already discussed each BM layer then received the extracted local information. Shreds of evidence that support contextual "time-frequency signatures" were gathered from the information, an essential indication for music genre identification. The structure of the BM significantly reduced the need for down-sampling. Despite the transition layer, by down-sampling, the size of feature-maps and the number of channels were reduced. A BN, ReLU activation, convolution, and average-pooling made the transition layer. In the final decision layer, global average pooling [28] took the average of each feature map to form a resulting vector and

fed it to a softmax log-loss function, which produced a distribution over genre labels.

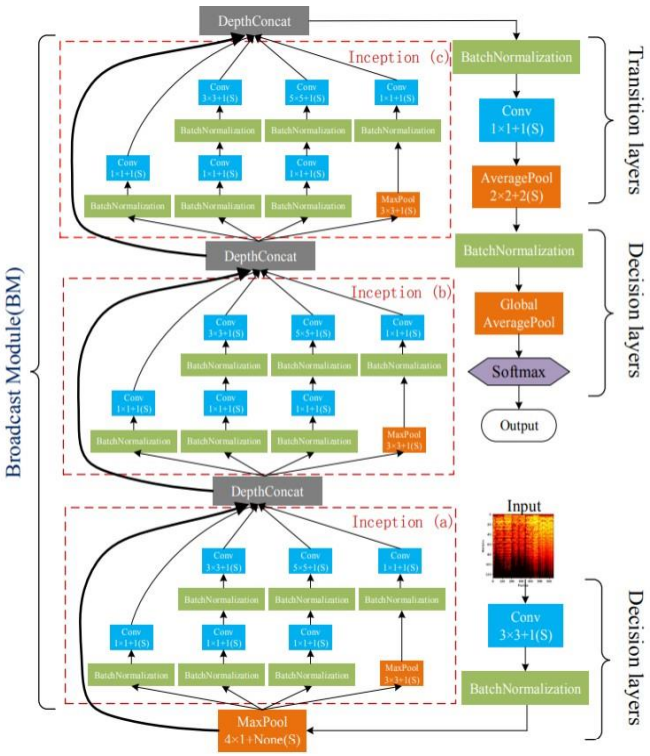

Fig.2. The architecture of the used BBNN, sourced from [1]

Table 1. Configuration of the used BBNN (Each convolution layer shown in the table corresponds to the sequence BN-ReLU-Conv), sourced from [1]

Type	Layers	Output Size	Filter Size/Stride (Number)	Params
SL	Convolution	647 x 128 x 32	3 x 3/1(32)	320
	Max Pool	161 x 128 x 32	4 x 1/*None*	
BM	Inception (a), top	-	[1 x 1/1(32)conv]*3,[3 x 3/1maxpool] *1	3,168
	Inception (a), bottom	161 x 128 x 160	[3 x 3/1(32)conv] * 1, [5 x 5/1(32)conv]*1 [1 x 1/1(32)conv]*1	35,936
	Inception (b), top	-	[1 x 1/1(32)conv]*3, [3 x 3/1maxpool] * 1	15,456
	Inception (b), bottom	161 x 128 x 288	[3 x 3/1(32)conv]*1, [5 x 5/1(32)conv] *1 [1 x 1/1(32)conv]*1	40,032
	Inception (c), top	-	[1 x 1/1(32)conv]*3, [3x3/1maxpool] * 1	27,744
	Inception (c), bottom	161 x 128 x 416	[3 x 3/1(32)conv]*1, [5 x 5/1(32)conv] *1 [1 x 1/1(32)conv]*1	44,128
TL	Convolution	161 x 128 x 32	1 x 1/1(32)	13,344
	Max Pool	80 x 64 x 32	2 x 2/2	
DL	Global Average Pool	1 x 1 x 32	-	
	Softmax	1 x 1 x 10	-	330
Total Params				**180,458**

3.2. Performed Experiment

BBNN, along with six different deep learning models and one traditional model, was trained and tested using GTZAN [37], Ballroom [38], and Extended Ballroom [39] datasets. The preprocessing program Librosa [40] transformed the files of a dataset into a Mel-spectrogram of size 647×128 (30 seconds audio), which was the input to the BBNN. The ADAM optimizer [41] trained the architecture, also the three datasets used a batch size of 8 for 100 epochs. The researchers also used an early stopping mechanism in the training phase. Fig. 3 shows that BBNN converged to a low loss both in training and verification sets. 10-fold cross-validation, by randomly partitioning training, testing, and validation sets into $8/1/1$ proportion, was used to evaluate the genre classification accuracy.

4. Independent Recurrent Neural Network (Indrnn) [2]

For sequential data such as music signals, the use of RNN is wide. Although GRU and LSTM, the popular variants of RNN, successfully deal with the gradient vanishing and exploding problem of vanilla RNN, the architectures are still susceptible to gradient decay in a deep network. Furthermore, these architectures cannot work on a long-time scale, such as long music clips. Considering these, Wu et al. used Independent Recurrent Neural Network (IndRNN) for music genre classification, as it could learn long-term dependencies better than LSTM and RNN. Gradient vanishing and exploding were solved by adjusting time-based gradient backpropagation. Scattering transforms for data pre-processing minimized the information loss. Furthermore, the use of ReLU as an activation function makes the structure trained IndRNN more robust.

4.1. Construction of the Used IndRNN

The used IndRNN had three parts, starting with the scattering transform, which pre-processed the dataset with preliminary feature extraction. Then, A 5-layer IndRNN with tagged data along with the ReLU function trained the data. Finally, softmax completed the classification of music genres.

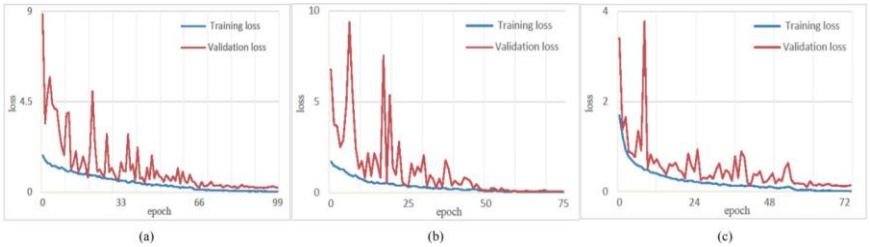

Fig.3. Loss curves of training and validation obtained on (a) GTZAN, (b) Ballroom, and (c) Extended Ballroom datasets, sourced from [1]

Scattering transform performs better than MFCC and Mel spectrogram on a large time scale, and the information loss is also less. Mel spectrogram loses information, which scattering transform with the help of a cascade of wavelet decomposition, and modulus operators recover. $\left|x*\psi_{\lambda_1}\right|*\phi(t)$ was the calculated value of Mel-frequency spectral coefficient [42], where x represented an audio signal, ψ_{λ_1} represented the wavelet and $\phi(t)$ represented a low pass filter. Wavelet modulus coefficients recovered high frequencies removed by the low pass filter. A local translation-invariant descriptor $S_0x(t)=x*\phi(t)$ was obtained by a time-average operation on a signal x and removed the high-frequencies, which can be recovered by wavelet modulus transform $|W_1|$,

$$|W_1|x=\left(x*\phi(t),\left|x*\psi_{\lambda_1}(t)\right|\right), \tag{3}$$

Wavelets of the same frequency resolution as Mel-frequency filters were defined. Moreover, the average unit was used in the transform to make the wavelet modulus coefficient invariant to the translation. The first-order of scattering coefficients were represented by,

$$S_1x(t,\lambda_1)=\left|x*\psi_{\lambda_1}\right|*\phi(t), \tag{4}$$

The second wavelet modulus transformed $|W_2|$ for each $|x*\psi\lambda_1|$,

$$|W_2|\left|x*\psi_{\lambda_1}\right|=\left(\left|x*\psi_{\lambda_1}(t)\right|*\phi,\left|x*\psi_{\lambda_1}\right|*\psi_{\lambda_2}\right), \tag{5}$$

here, the operation of λ_2 on modulus coefficients recovered the lost high frequencies. The coefficients passed through the same low pass, used in the first layer, to ensure the invariance to time shifts.

The second and the n-order scattering coefficients were represented by,

$$S_2x(t,\lambda_1,\lambda_2)=\left|\left|x*\psi_{\lambda_1}\right|*\psi_{\lambda_2}\right|*\phi(t), \tag{6}$$

and,

$$S_n x(t,\lambda_1,\lambda_2,\ldots,\lambda_n) = \left|\left|\left|x*\psi_{\lambda_1}\right|*\ldots\left|\psi_{\lambda_n}\right|*\phi(t),\right.\right. \tag{7}$$

respectively.

IndRNN, similar to [43], was used. The Hadamard product processed the recurrent input of IndRNN,

$$h_t = \sigma\left(Wx_t + \mathbf{u}\odot h_{t-1} + b\right), \tag{8}$$

here, h_t defined as the hidden status at time step t, h_{t-1} was the unseen state, $\mathbf{u}$ was the recurrent weight, W was the weights, b was the basis, σ was the activation function, and $\odot$ represented Hadamard product. The

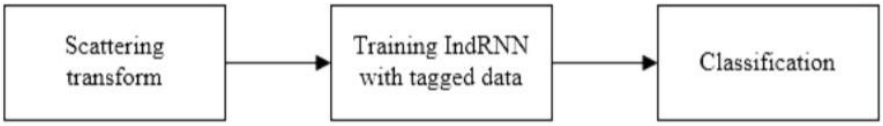

Fig.4. The architecture of the IndRNN used, sourced from [2]

formula indicated that in this architecture, at time step t, each neuron accepted information present at the moment along with the information in its hidden layer at time step $t-1$ and allowed each neuron of the architecture to process the time and space model. The h_t and h_{t-1} were independent of each other, which suggested W extracted spatial dependencies of the input, while $\mathbf{u}$ extracted temporal characteristics. Layers of IndRNN were stacked to make connections between the neurons, and each IndRNN was further stacked to make a deep IndRNN network. For the $n-th$ neuron $h_{n,t}$ the gradient back-propagated to time step t, which could be represented as,

$$\begin{aligned}\frac{\partial J_n}{\partial h_{n,t}} &= \frac{\partial J_n}{\partial h_{n,T}}\frac{\partial h_{n,T}}{\partial h_{n,t}} = \frac{\partial J_n}{\partial h_{n,T}}\prod_{k=t}^{T-1}\frac{\partial h_{n,k+1}}{\partial h_{n,k}} \\ &= \frac{\partial J_n}{\partial h_{n,T}}\prod_{k=t}^{T-1}\sigma'_{n,k+1}u_n = \frac{\partial J_n}{\partial h_{n,T}}u_n{}^{T-1}\prod_{k=t}^{T-1}\sigma'_{n,k+1},\end{aligned} \tag{9}$$

where the objective at time step T was J_n and $\sigma'_{n,k+1}$ the derivative of the activation function. The gradient of activation was within a definite range, and the formula as a whole suggested the exponential term of the scalar value u_n, which was only involved by the gradient. So, IndRNN depended on the value of recurrent weights, which was adjusting the exponential part, that is, keepings $u_n{}^{T-1}\prod_{k=t}^{T-1}\sigma'_{n,k+1}$ in a particular range, gradient vanishing, and exploding were solved.

4.2. Performed Experiment

For the evaluation of the architecture GTZAN dataset [37] was used. The audio inputs were of 30 seconds, which converted to mono by sampling at 16 kHz. The dataset was shuffled randomly and placed into ten folders, of which nine were used to train and 1 to test. Using 10-fold-cross-validation performance was evaluated, and for final test accuracy, an average accuracy of 10 times it was used. Dropout was set at 0.5, while the learning rate was at $1e-5$. Early stopping was applied when the curve convergence was stable.

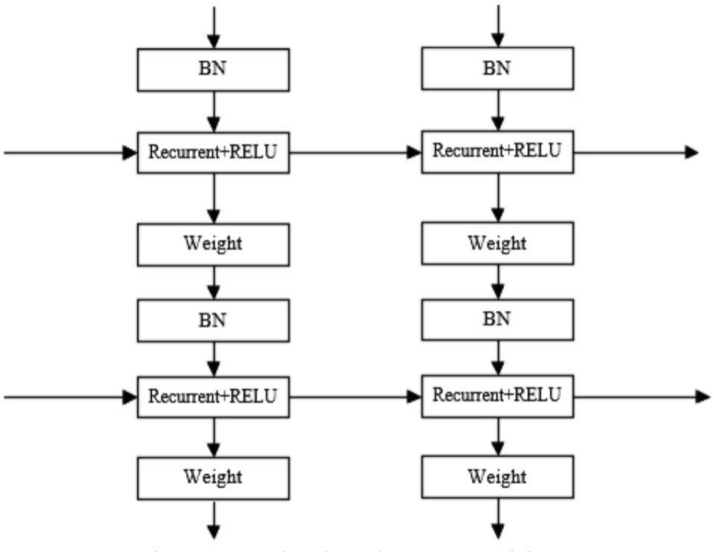

Fig.5. Deep IndRNN architecture, sourced from [2]

5. CRNN in Time and Frequency Dimensions (CRNN-TF) [3]

Wang et al. used CRNN in Time and Frequency dimensions or better known as CRNN-TF, to classify the music genre. The study used a CNN with different kernels and pool strides so that it outputs an activation map of mid-level frequency. After that, to convert the activation map into eight sequences, a novel multi-directional scanning strategy was employed. Then one grid LSTM-RNN was fed each of the obtained sequences. By merging the output of all the LSTM-RNN blocks, a high-level feature vector of the input music signal. Hence the architecture captured spatial dependencies in Time and frequency dimensions of the music signal in multiple-dimensions.

5.1. Construction Of The Used CRNN-TF

As Mel-spectrogram represents the short-term power spectrum of an audio signal and can capture low-level details along multiple dimensions, the preprocessor transformed the music signal into a log-amplitude Mel-spectrogram (LAMS), which is a scaled Mel-spectrogram. The pre-processing followed [30]. To obtain tracks of equal length, trimming the music tracks to 29 seconds was done, which was the input to the architecture. The output of the pre-processing was a 96×1360 matrix of Mel-spectrogram. Each row of the matrix corresponded to a Mel-frequency scale, while the column to a Mel-frequency time frame.

The architecture used 4-layer CNN, where each layer was of a set of 3×3 filters and 2×4 pooling strides. The first layer contained 64 filters, while the remaining layers had 128 filters. The configuration produced an activation map of $(6,6,128)$. Because of the different convolution and pooling operators, the activation map contained 6 Mel-frequencies.

RNN structures, namely a network of multi-directional RNN and a Grid LSTM block, were used in the CRNN-TF. The activation map obtained using the CNN was fed to a networked multi-directional RNN to convert the activation map to eight sequences, each of which was generated by one scanning method (top-down/bottom-up, left-to-right/right-to-left, row-wise/column-wise). The three variables had eight combinations that produced eight sequences. Then standard one Grid LSTM network [44] was fed each of the obtained sequences. A Grid LSTM block had 32 grid cells, and a grid cell was of two LSTM cells. The LSTM cells model spatial dependencies in time and depth dimensions. Each block outputs a 32-dimensional vector.

Table 2. Parameters of transformation used in the pre-processing of CRNN-TF

Name of Parameter	Parameter Value
Down Sampling Rate	12 KHz
Hop Size	256
FFT	512 - point
Number of Mel-bins	96

The RNN blocks outputted feature vectors and inputted to the fully-connected layer, where the concatenation of them formed a 256-dimensional feature vector and fed to a standard fully connected layer consisting of linear transformation and a softmax layer. Each neuron outputs a probable music genre.

The objective function of CRNN-TF was,

$$\begin{aligned}\mathcal{L}(\theta_*) = & \\ & -\frac{1}{|\mathcal{A}|}\sum_{X,\omega} P(\omega|X;\theta_*)\log\left(P(\omega|X;\theta_*)\right) \\ & +\lambda_1\sum_{L,K}\left(\|W_k^{(L)}\|^2 + \|b_k^{(L)}\|^2\right) \\ & +\lambda_2\left(\|W_o\|^2 + \|b_o\|^2\right) \\ & +\lambda_3\sum_{\xi,\psi,\varphi}\left(\|W_\xi^{(\psi,l,\varphi)}\|^2 + \|U_\xi^{(\psi,l,\varphi)}\|^2 + \|b_\xi^{(\psi,l,\varphi)}\|^2\right),\end{aligned} \tag{10}$$

where X was the input of the signal, ω was the music label (genre), θ_* was the set of parameters of the architecture which included weights $W_k^{(L)}$ and bias terms $b_k^{(L)}$ of the CNN network, weights $W_\xi^{(\psi,l,\varphi)}$, $U_\xi^{(\psi,l,\varphi)}$ and bias terms $b_\xi^{(\psi,l,\varphi)}$ of the RNN network, and weights W_o and bias b_o of the fully-connected layer. More details on the parameters of CNN and RNN are stated in [16] and [44], respectively. λ_1, λ_2, and λ_3 were hyper-parameters that balanced the weight decay of different components in the network. Since genre classification is a multi-class classification problem, softmax activation along with categorical cross-entropy served as a loss function. The dropout technique was further employed at the fully connected layer to reduce over-fitting,

$$g = W_o\left(y \odot q\right) + b_o, \tag{11}$$

where q was a masking vector applied on concatenated feature vector y, $\odot$ was the Hadamard product, W_o was the linear transformation function, and b_o was the bias term.

5.2. Performed Experiment

Other deep learning architectures, Fully Convolutional Neural Network FCN [16], Timbre CNN [45], End-to-end [46], and CRNN [30], were compared to CRNN-TF for similar music genre classification. Using the medium-sized Free Music Archive dataset [47], Wang et al. trained and tested the architecture. The split of the dataset following the method described in [47] resulted in the obtainment of $19{,}922$ training tracks, $2{,}505$ validation tracks, and $2{,}573$ testing tracks. Back-propagation trained the CRNN-TF and used a batch sample size of 32, an initial learning rate of 0.001 with a 0.9 decay rate after each epoch, and an ELU activation function with $\alpha = 1.0$. The ADAM optimizer trained the architecture, and after every convolution layer, applied BN on the CNN. For the training of fully-connected layers, the dropout rate was 5, and the hyper-parameters, λ_1, λ_2, and λ_3 was 10^{-6}. AUC score, recall, precision, f1 score, and accuracy (representing fraction of misclassified genres) were the parameters for evaluation. The performance was predicted first by considering each genre as a binary label, then average performance over all the genres was evaluated.

6. Outcome Analysis and Discussion

The aim of Liu et al.'s BBNN architecture was to handle the multi-scale of audio feature and use the low level along with high-level information of Mel-spectrogram to achieve higher music genre classification accuracy [1]. The BBNN, equipped with a novel BM module consisting of inception blocks, helped the architecture to handle multi-scale of audio features. The BM was connected to form a dense network, which aided the architecture to transmit low-level information to the higher layer. Following the architectural setup and experiment as discussed in section 2, Liu et al.'s BBNN achieved a remarkable classification accuracy of 93.9%, 96.7%, and 97.2% on the GTZAN, Ballroom, and Extended Ballroom datasets and thus it is evident the use of low-level information and the ability to manipulate the time-frequency scale of audio features contributes to greater classification accuracy. On the GTZAN dataset, Liu et al.'s BBNN achieved a classification accuracy of $93.9/\%$ and average precision of 93.7%. As stated in the study, the BBNN struggled to distinguish Rock from Country and Metal, while other genres had been more or less correctly classified, Fig. 6. The study mentioned, the genres share similar frequency information, which has confused. Although the BBNN achieved a remarkable classification accuracy of 96.7% and average precision of 97.2% on the Ballroom dataset, it was relatively confused between Rumba and Slow Waltz, Fig. 6, because the genre boundaries of these two genres are not clear. The potential of BBNN shines through the accuracy (97.2%) it achieved on the Extended Ballroom dataset and that too without requiring any pre-training on this larger dataset. The collected confusion matrix

shows, Fig. 6, BBNN severely confused Rumba and Slow Waltz with Waltz. The study stated that the three genres contain similar patterns [48] for which they were difficult to classify. Furthermore, in the Extended Ballroom dataset, samples of the Wcswing genre was significantly small. For this reason, the learning opportunities of BBNN got extremely limited, which resulted in the relatively worse classification of this genre.

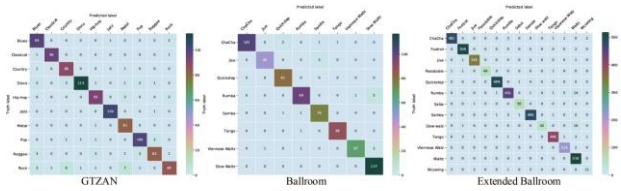

Fig.6. Confusion Matrices representing accuracies of training, validating, and Testing of each fold on the datasets, sourced from [1]

Wu et al. used IndRNN's superior ability to learn long term dependencies for music genre classification because such capability helps to process music signals that depend on multi-scale features [2]. 5-layer IndRNN architecture was applied on the GTZAN dataset for music genre classification and achieved 96% accuracy with a training time of 23 seconds per iteration following the architectural setup and experiment as discussed in section 4. Although the architecture did not outperform LSTM in terms of accuracy (97%), it did outperform LSTM (0.68 seconds per iteration) in training time. The collected figure, Fig. 7, shows that only after 75 epochs, the IndRNN converged to very high accuracy. Wu et al. compared the IndRNN to RNN and LSTM, from which to us the comparison between LSTM and IndRNN seems more significant. The accuracy of LSTM was higher at 97%, but the training time of each iteration was high 0.68 seconds. However, the IndRNN did not lack behind in terms of accuracy and scored an accuracy of 96%. But the architecture shined more in training time as it was only 23 seconds per iteration, which was almost a third of that of LSTM.

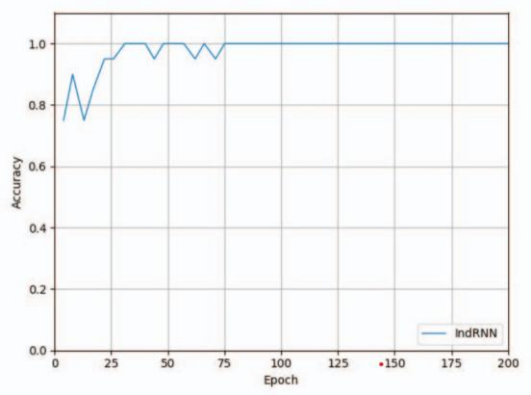

Fig.7. AUC-ROC score of the used IndRNN, sourced from [2]

Liu et al.'s BBNN captured spatial dependencies of music signal in time-frequency dimensions [1], whereas Wu et al.'s IndRNN captured spatial dependencies of music signal in time dimension only [2]. Wang et al. improved existing CRNN architecture [30] by enabling the RNN layer to capture spatial dependency of the audio signal in both time and frequency dimensions, just as BBNN, for MGC tasks. He achieved this by implementing the architecture CRNN-TF, which outputted activation map of mid-level time-frequency representation [3]. Following the architectural setup and experiment, as discussed in section 5, Wang et al.'s CRNN-TF showed promise in MGC tasks scoring higher AUC, Recall, F1 Score, and Accuracy corresponding to 0.910, 0.435, 0.423, and 0.647, which was higher than existing CRNN architecture [30]. According to Wang et al., several genres were better clustered in CRNN-TF feature space compared to that of CRNN [30], and this is the reason for the better performance of CRNN-TF compared to CRNN. The study further stated that FCN [16] achieved the highest precision among all the genres and a probable reason for this was the more sophisticated CNN architecture with more layers, kernels, and a multiscale strategy.

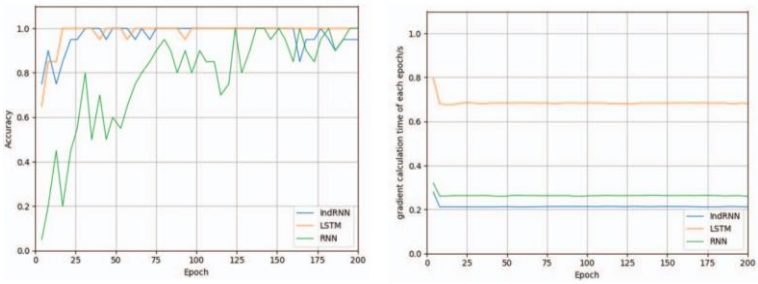

Fig.8. Comparison of classification accuracy and time of each iteration of IndRNN, LSTM and RNN, sourced from [2]

To us, other than the staggering classification accuracy of BBNN, another significant achievement of the BBNN [1] was the reduction of the need for data-augmentation with the help of its compact parameters, which is often a requirement in traditional CNN. MIR tasks involve training of classifiers from few labeled data are often a challenge, which the BBNN architecture addressed and solved to a certain extent that it eliminates the need for pre-training on bigger datasets such as the Extended Ballroom dataset. However, there is still room for improvement as the architecture struggles to classify genres that are quite similar. We find IndRNN [2] to be a promising architecture for MGC tasks as the trade-off between accuracy and training time is very little. Furthermore, [43] showed that increasing the number of layers of the architecture results in better performance in Cross-Subject (CS) and Cross-View (CV). Being inspired by the work of Li et al. [43], we propose increasing the number of layers of the used IndRNN to obtain performance improvement of the architecture for MGC tasks. Wang et al. found that the precision of CRNN-TF was not as that of FCN [16] and believed that replacing the CNN layer of the architecture with a more sophisticated one such as that of [16] can improve the performance [3]. In addition to this, we propose replacing the LSTM layer of the architecture with structures that can work on a long time scale and learn long-term relationship better, such as IndRNN [43], and hence improve the performance because LSTM struggles to work in the long time scale as in music clips. Furthermore, we observe the comparatively new CRNN architecture precisely the CRNN-TF lacks behind in comparison to the more established CNN and RNN architectures, that is, the BBNN and IndRNN. Wang et al.'s CRNN-TF scored an AUC score of 0.910 , which surpassed the previous state of the art CRNN architecture [30], despite the achievement, the overall performance of the architecture is significantly behind that of BBNN and IndRNN, which uses low-level information in the decision-making layer and learn long term dependencies respectively to achieve superior classification accuracy.

Table 3. Performance of CRNN-TF compared to other deep learning architectures, sourced from [3]

Method	AUC	Recall	F1-Score	Accuracy
FCN	0.907	0.430	0.403	0.639
Timbre CNN	0.891	0.364	0.350	0.617
End-to-end	0.891	0.384	0.345	0.614
CRNN	0.903	0.407	0.402	0.634
CRNN-TF	**0.910**	**0.435**	**0.423**	**0.647**

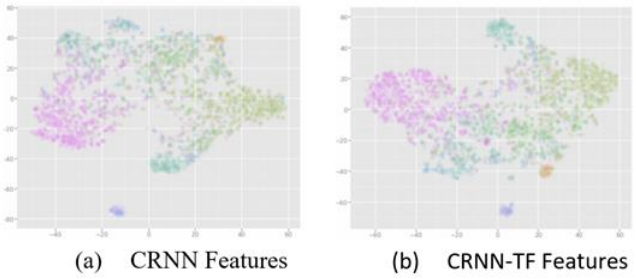

Fig.9. Distribution of FMA Testing Data in CRNN (a) and CRNN-TF (b) feature spaces, sourced from [3]

In this study we analyzed and discussed three improved architectures of CNN, RNN, and CRNN of which the architectures of CNN and CRNN-TF (BBNN and CRNN-TF) achieved the highest classification accuracy for music genre classification among the CNN and CRNN architectures, while IndRNN achieved remarkable classification accuracy of 96% , which despite not being the highest among the RNN architectures is still of significance as the architecture achieved the classification accuracy with much efficiency (23 seconds per iteration), while keeping the classification accuracy trade-off to only 1% in comparison to LSTM-RNN. The study holds discussion of the vital architectures of CNN, RNN, and CRNN for music genre classification and would allow future researchers to get

information and understanding of the highest music genre classification accuracy achieving architectural variants of the three deep learning models, CNN, RNN, and CRNN from one study.

7. Conclusion

To the best of our knowledge, no previous work performed a comparative analysis of the three deep learning algorithms, CNN, RNN, and CRNN. In this study, we reviewed and discussed three improved deep learning architectures of CNN, RNN, and CRNN, namely BBNN, IndRNN, and CRNN-TF, respectively, all of which classified music genres. All of the three architectures handled multi-scale features of audio signals in their unique ways and achieved remarkable results. BBNN focused on the transmission of both low-level and high-level information and extracting information in the time-frequency of the audio signal. Furthermore, BBNN used compact parameters, which reduced the need for data-augmentation. IndRNN focused on learning long-term dependencies, while CRNN-TF focused on the extraction of spatial dependencies on both time and frequency dimensions for MGC. Liu et al.'s BBNN had outstanding performance but struggled to distinguish similar genres. Wu et al.'s IndRNN could not outperform LSTM-RNN in terms of classification accuracy but demonstrated remarkable efficiency by reducing the training time to a third of that of LSTM, also keeping the trade-off in terms of accuracy to only 1% . Wang et al.'s CRNN-TF outperformed existing state of the art CRNN architecture but lacked much behind in terms of accuracy when compared to the other two deep learning architectures, BBNN and IndRNN. We also proposed that increasing the layers of IndRNN can yield better performance, and the LSTM layer of CRNN-TF can be replaced with IndRNN to improve the performance of the architecture. Due to time constraints, we could not implement the architectures, nor could implement our proposed changes to the architectures, but in the future, we will be focusing on implementing the architectures and improving the RNN and CRNN architecture according to our proposal.

References

[1] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification, 2019.

[2] W. Wu, F. Han, G. Song, and Z. Wang. Music genre classification using independent recurrent neural network. In 2018 Chinese Automation Congress (CAC), pages 192–195, 2018.

[3] Z. Wang, S. Muknahallipatna, M. Fan, A. Okray, and C. Lan. Music classification using an improved crnn with multi-directional spatial dependencies in both time and frequency dimensions. In 2019 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2019.

[4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. Proceedings of the IEEE, 96(4):668–696, 2008.

[5] P. Mermelstein. Distance measures for speech recognition - psychological and instrumental. Pattern Recognition and Artificial Intelligence (C. H. Chen, ed.), pages 374–388, 1976.

[6] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In Proceedings of 12th International Conference on Pattern Recognition, volume 1, pages 582–585 vol.1, 1994.

[7] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5):293–302, 2002.

[8] Y. M. G. Costa, L. S. Oliveira, A. L. Koericb, and F. Gouyon. Music genre recognition using spectrograms. In 2011 18th International Conference on Systems, Signals and Image Processing, pages 1–4, 2011.

[9] Volodymyr Mnih, Nicolas Heess, Alex Graves, and koray kavukcuoglu. Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 2204–2212. Curran Associates, Inc., 2014.

[10] Jimmy Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. CoRR, abs/1412.7755, 2015.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 91–99. Curran Associates, Inc., 2015.

[12] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, page 2048–2057. JMLR.org, 2015.

[13] Nianyin Zeng, Hong Zhang, Baoye Song, Weibo Liu, Yurong Li, and Abdullah M. Dobaie. Facial expression recognition via learning deep sparse autoencoders. Neurocomput., 273(C):643–649, January 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[15] P. Chiliguano and G. Fazekas. Hybrid music recommender using content-based and social information. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2618– 2622, 2016.

[16] Keunwoo Choi, György Fazekas, and Mark B. Sandler. Automatic tagging using deep convolutional neural networks. CoRR, abs/1606.00298, 2016.

[17] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. CoRR, abs/1207.0580, 2012.

[18] Tom L. H. Li, Antoni B. Chan, and Andy H. W. Chun. Automatic musical pattern feature extraction using convolutional neural network. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, pages 546–550, 2010. International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010 ; Conference date: 17-03-2010 Through 19-03-2010.
[19] Thomas Lidy. Parallel convolutional neural networks for music genre and mood classification. 2016.
[20] Christine Senac, Thomas Pellegrini, Florian Mouret, and Julien Pinquier. Music feature maps with convolutional neural networks for music genre classification. In Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, CBMI '17, New York, NY, USA, 2017. Association for Computing Machinery.
[21] Hareesh Bahuleyan. Music genre classification using machine learning techniques. CoRR, abs/1804.01149, 2018.
[22] Hansi Yang and Wei-Qiang Zhang. Music Genre Classification Using Duplicated Convolutional Layers in Neural Networks. In Proc. Interspeech 2019, pages 3382–3386, 2019.
[23] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 28(10):2222–2232, 2017.
[24] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.
[25] Sepp Hochreiter and Jürgen Schmidhuber. Long shortterm memory. Neural Computation, 9(8):1735–1780, 1997.
[26] Scott Zhang, Huaping Gu, and Rongbin Li. Music genre classification: Near-realtime vs sequential approach. 2019.
[27] Jia Dai, Shan Liang, Wei Xue, Chongjia Ni, and Wenju Liu. Long short-term memory recurrent neural network based segment features for music genre classification. In 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), pages 1–5. IEEE, 2016.
[28] Jan Jakubik. Evaluation of gated recurrent neural networks in music classification tasks. In International Conference on Information Systems Architecture and Technology, pages 27–37. Springer, 2017.
[29] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing, pages 1422–1432, 2015.
[30] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2392–2396. IEEE, 2017.
[31] Sharaj Panwar, Arun Das, Mehdi Roopaei, and Paul Rad. A deep learning approach for mapping music genres. In 2017 12th System of Systems Engineering Conference (SoSE), pages 1–5. IEEE, 2017.
[32] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. IEEE Signal Processing Magazine, 23(2):133– 141, 2006.
[33] Déborа C Corrêa and Francisco Ap Rodrigues. A survey on symbolic data-based music genre classification. Expert Systems with Applications, 60:190–210, 2016.
[34] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. IEEE transactions on multimedia, 13(2):303–319, 2010.
[35] Snigdha Chillara, AS Kavitha, Shwetha A Neginhal, Shreya Haldia, and KS Vidyullatha. Music genre classification using machine learning algorithms: A comparison. 2019.
[36] D Pradeep Kumar, BJ Sowmya, KG Srinivasa, et al. A comparative study of classifiers for music genre classification based on feature extractors. In 2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pages 190–194. IEEE, 2016.
[37] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing, 10(5):293–302, 2002.
[38] Pedro Cano, Emilia Gómez Gutiérrez, Fabien Gouyon, Herrera Boyer, Markus Koppenberger, Bee Suan Ong, Xavier Serra, Sebastian Streich, Nicolas Wack, et al. Ismir 2004 audio description contest. 2006.
[39] Ugo Marchand and Geoffroy Peeters. The extended ballroom dataset. 2016.
[40] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference, volume 8, pages 18–25, 2015.
[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[42] Joakim Andén and Stéphane Mallat. Deep scattering spectrum. IEEE Transactions on Signal Processing, 62(16):4114–4128, 2014.
[43] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5457–5466, 2018.
[44] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. arXiv preprint arXiv:1507.01526, 2015.
[45] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In 2017 25th European Signal Processing Conference (EUSIPCO), pages 2744–2748. IEEE, 2017.
[46] Sander Dieleman and Benjamin Schrauwen. End-toend learning for music audio. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6964–6968. IEEE, 2014.
[47] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. arXiv preprint arXiv:1612.01840, 2016.
[48] Athanasios Lykartsis and Alexander Lerch. Beat histogram features for rhythm-based musical genre classification using multiple novelty functions. 10.14279/depositonce-9530, 2015.

Authors' Profiles

Quazi Ghulam Rafi received his B.Sc. in Computer Science and Engineering from the American International University - Bangladesh, Dhaka, Bangladesh in 2020. He is currently working as a Front-end Developer at Battery Low Interactive Limited, Dhaka, Bangladesh. Mr. Rafi was twice awarded the Dean's List of Honors from the Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh. He is also a Microsoft Certified: Azure Developer Associate.

Mohammed Noman was born in Riyadh, Saudi Arabia on the 28th of June 1997. He is studying B.Sc. in Computer Science and Software Engineering at the American International University-Bangladesh, Dhaka, Bangladesh. In addition, he is working as a Teaching Assistant at the American International University-Bangladesh, Dhaka, Bangladesh.

Sadia Zahin Prodhan was born in Dhaka, Bangladesh on the 13th of October 1998. She received her B.Sc. in Computer Science and Engineering from the American International University-Bangladesh, Dhaka, Bangladesh in 2020. She worked as a Teaching Assistant at the American International University-Bangladesh, Dhaka, Bangladesh.

Sabrina Alam Mohona was born in Rajshahi, Bangladesh in the year 1998. She is studying B.Sc. in Computer Science and Engineering at the American International University-Bangladesh, Dhaka, Bangladesh. In addition, he is working as a Teaching Assistant at the American International University-Bangladesh, Dhaka, Bangladesh.

Dr Dip Nandi has completed his PhD in Computer Science from the School of Computer Science and Information Technology at the RMIT University, Melbourne, Australia. His research interest includes Software Engineering, Management Information Systems, E-learning etc. He is currently working as an Associate Professor, Department of Computer Science and Director, Faculty of Science and Technology in the American International University-Bangladesh, Dhaka, Bangladesh.

How to cite this paper: Quazi Ghulam Rafi, Mohammed Noman, Sadia Zahin Prodhan, Sabrina Alam, Dip Nandi, "Comparative Analysis of Three Improved Deep Learning Architectures for Music Genre Classification", International Journal of Information Technology and Computer Science(IJITCS), Vol.13, No.2, pp.1-14, 2021. DOI: 10.5815/ijitcs.2021.02.01

