Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
5(1): 94–111.
DOI: 10.3934/aci.2025007
Received: 11 February 2025
Revised: 06 April 2025
Accepted: 07 April 2025
Published: 15 April 2025

Research article

Ethiopian music genre classification using deep learning

Eshete Derib Emiru[1,*] and Estifanos Tadele Bogale[2]

[1] Department of Information Technology, Debre Markos University, Debre Markos, Ethiopia
[2] Department of Library and Information service, Debre Markos University, Debre Markos, Ethiopia

* **Correspondence:** Email: eshete_derb@dmu.edu.et; Tel: +251911567807.

Academic Editor: Pasi Fränti

Abstract: The process of genre classification involves the identification of distinctive stylistic elements and musical characteristics that define a particular genre. It assists in developing a comprehensive understanding of the historical context, cultural influences, and musical evolution of a particular genre. This study was conducted to resolve the challenges of classifying Ethiopian music genres according to their melodic structures using deep learning techniques. The main objective was to develop a deep learning model for effective audio classification into six genres classes of Ethiopian music: Ancihoye Lene, Ambassel Major, Ambassel Minor, Bati, Tizita Major, and Tizita Minor. To achieve this, we first prepared a dataset consisting of 3952 audio recordings, which includes 533 tracks from Ethiopian Orthodox church music and 3419 samples of secular Ethiopian music. A total of 46 unique features, namely chroma short-time Fourier transform (STFT), root mean square error (RMSE), spectral centroid, spectral bandwidth, roll-off, zero crossing rate, and mel frequency cepstral coefficient (MFCC) 1 up to MFCC40, were extracted both at middle-level and low-level audio features from each sample, focusing on aspects suggested by Ethiopian music experts and preliminary experiments that highlighted the importance of tonality features. A 30-second segment of audio recordings was selected for feature extraction, resulting in datasets formatted in both CSV and JSON for further processing. We proposed deep learning algorithms namely convolutional neural networks (CNN), recurrent neural networks (RNN), a parallel RNN–CNN architecture, and long short-term memory (LSTM) networks for our classification by developing models. Our experiments revealed that the LSTM model achieved the best performance, reaching a classification accuracy of 97% using 40 MFCC features extracted from audio datasets.

Keywords: music genre classification; audio feature; LSTM; CNN; RNN

1. Introduction

Ethiopian music has had a significant social and cultural impact on Ethiopians and has been around for a long time. The music is mostly based on the pentatonic scale and modal tone scheme. Vocal music is equally popular as instrumental music. It is common to listen to vocal music without accompaniment, such as without drums, as well as instrumental music with string instruments and percussion, like kirar. Four basic genres (Tezeta, Bati, Ambassel, and Anchihoy Lene) of the fundamental modal system known as Qenet are used in the music of the Ethiopian Highlands. Three additional modes are variations on these, namely Tezeta Minor, Ambasel Minor, and Bati Minor. Some songs take the name of their Qenet, such as Tizita, a song of reminiscence.

Music is an important part of everyday life. It exists in various shapes and styles all over the world. Because everyone has different tastes in music, music should be classified and recommended to listeners. It is an important research topic, as there are many applications in auditory apps, and other platforms. The creation and sharing of music files across various media have increased significantly. As a result, indexing, browsing, and retrieving music files from a specific music genre has become difficult and time-consuming [1].

Dozens of digital music classification techniques have been introduced [2]; however, most of them have been developed and tested only on well-known Western musical data. In Ethiopia, music classification is still performed by individual music experts for archiving or related purposes. Although Ethiopian music is now available in digital form, classification remains slow and insufficient. Consequently, despite the contributions of the composer Saint Yared, who developed various musical scales (zema) in Ethiopia during the 6th century [3], Ethiopian music is not well known worldwide. Ethiopian music is based on several types of scales, with four pentatonic scales (genres) being particularly important [4,5]. These major genres include Anchihoye Lene, Ambassel Major, Ambassel Minor, Bati Major, Bati Minor, Tizita Major, and Tizita Minor. Since every piece of music written in these genres has its own characteristic styles, genre classification is closely related to or synonymous with the genre classification of music from other countries. The main challenge in classifying Ethiopian music genres is the lack of training data.

Recent studies have explored the classification of melodic scales of a music genre, using machine learning and deep learning techniques, focusing on features such as the arrangement and position of notes. Kızrak employed deep belief networks (DBNs) to classify Turkish Makams [2]. Traditional machine learning methods like support vector machines (SVM) are also used for music classification. Nowadays, deep learning algorithms are used for classification tasks. We selected convolutional neural networks (CNNs), recurrent neural networks (RNN) with long short-term memory (LSTM) units, and a hybrid model combining CNN and LSTM (CNN-LSTM) for Ethiopian music genre classification based on insights from the existing literature. Since these deep learning algorithms need a huge amount of training data, we created a new dataset containing data from the six main genres. The objective of the study was to develop an efficient Ethiopian music genre classification model using CSV, JSON and audio datasets, and audio features such as chroma staff, root mean square error (RMSE), spectral centroid, spectral bandwidth, roll-off, zero crossing rate, mel frequency cepstral coefficients (MFCCs), and Tonnetz features by taking the mean, min, and max features of each feature. Our study makes the following contributions:

- A dataset of 3952 audio recordings has been prepared, including 3419 samples of secular Ethiopian music and 533 tracks of Ethiopian Orthodox church music. This rich resource provides training and

validation for deep learning models for Ethiopian music classification, and it is available online for further research [6].

- The researchers extracted 46 unique features from the audio recordings, including chroma short-time Fourier transform (STFT), RMSE, spectral centroid, spectral bandwidth, roll-off, zero crossing rate, and MFCC1 to MFCC40. These features encompass both middle-level and low-level audio characteristics, specifically emphasizing aspects identified by Ethiopian music experts and preliminary experiments that underscore the significance of tonality features. This tailored approach represents a novel application of these audio features in the context of Ethiopian music
- To address the class imbalance issue in the dataset, the researchers employed various data augmentation techniques, including sample rate adjustment, room impulse response (RIR) analysis, 8-bit mu-law format and Ogg Vorbis compression, signal-to-noise ratio (SNR) assessment, and incorporation of GSM files.
- We experimented with four deep learning algorithms for audio classification, namely CNN, RNN, parallel RNN-CNN, and LSTM, allowing for a comparison of these architectures and the identification of the most effective approach for Ethiopian music classification.

2. Music generic classification

We reviewed the related literature and adapted it for our study by identifying gaps in their research. Many studies have been performed with a limited set of features, as indicated in Table 1. In the field of music genre classification, various models have been developed using machine learning and deep learning algorithms. The research gap indicates a need for more diverse audio features. It also calls for a larger and more balanced dataset. Current studies use limited records from each genre. This limitation results in low accuracy in classification. It hinders the exploration of potential benefits from adding more audio feature sets. All these research works are motives to classify the Ethiopian music genre using deep learning algorithms, large datasets, the six generic classes extended by classifying them into majors and minors, more feature representations of audio data, and various datasets, namely JSON and CSV, which are extracted from the original audio dataset and evaluated with different metrics.

Table 1. Literature review summary.

Article	Algorithm	Features	Best accuracy	Research gap
[2]	Deep Belief Networks (DBN)	MFCC	Classification precision: 93.10%	Other audio features also needed to be included and experimented with them.
[7]	CNN+Bi-GRU, CNN-LSTM, RNN-CNN	MFCC, Mel-spectrogram	CNN+Bi-GRU: 89.30%	Mel-spectrogram and MFCC and used 100 records from one genre. Adding more records from each genre and extract more features required.
[8]	RNN-CNN, KCNN-SVM	STFT, FFT Mel-spectrogram	RNN-CNN: 90.2%	Train on audio dataset other dataset formats were not included in the experiments.
[9]	CNN	MFCC	CNN: 69%	MFCC and less records from each genre and gained a low accuracy of 69%. Other audio features were not included in the experiments.
[10]	CNN	-Spectrogram Image	CNN: 62%	995 songs for 27 genres are a small data set, with 39 songs per genre and scored low accuracy.

Continued on next page

Article	Algorithm	Features	Best accuracy	Research gap
[11]	Random Forest, Logistic Regression, KNN, SVM	Mid and low label features	SVM: 80%	The research extracted 29 features from the audio, which is good, but experimented on a small data set.
[12]	CNN, RNN	FFT, MFCC, Mel-Spectrogram	CNN: 59%	The research used the GTZAN dataset and gained low accuracy result.
[13]	SVM	Spectrogram	SVM: 67.2%	The research used the Latin Music Dataset but only extracted one feature and experimented on one.
[14]	SVM	Spectral Roll-off	SVM: 83.3%	The research utilizes a dataset containing 45 songs from each genre and extracts two features from a relatively small CSV dataset.
[3]	AlexNet, ResNet50, VGG16 and LSTM	Filterbank, Mel-spectrogram, chroma, or MFCC	CNN: 95%	The research used 600 sample recordings of Orthodox Tewahedo chants, traditional Azmari songs and contemporary Ethiopian secular music which used small, recorded dataset.
[4]	SVM, ANN and Decision tree	36 tonal and 9 timbre features extracted using MAHLAB MIR toolbox	SVM: 86.96%	A total of 710 sample records (310 samples from the Ethiopian Orthodox church songs and 400 secular samples records) are used but it is small recorded dataset.
[8]	CNN, Bi-RNN	STFT	CNN: 92.0%	The research only included three audio features and other audio features were not included.

3. Ethiopian music genre classification

3.1. Data collection

The dataset consists of Tizita (ትዝታ) Major and Minor, Batti (ባቲ), Ambassel (አምባሰል) Major and Minor, and Anchihoye Lene (አንቺ ሆዬ ለኔ). The data needed rearrangement, as well as preprocessing, which means representing the audio files as a visual form or spectrogram, CSV, or JSON, and taking features in acoustic form from the long-recorded file, which must be segmented with equal size to have a uniform time interval of 30 seconds. Finally, the segmented audio files were changed into a visual representation, which is a spectrogram, CSV, or JSON. The data feed for different algorithms were experimented on. The number of audio data files is 814 for Bati, 768 for Anchihoye Lene, 707 for Tizita Major, 671 for Ambassel Major, 515 for Tizita Minor, and 477 for Ambassel Minor, with a total of 3952. The description of each genre's audio record dataset with the data source and number of records is given in Table 2.

The dataset contained a variety of audio files, including Begena (በገና) songs from the Ethiopian Orthodox church, modern songs with various instrument compositions, Begena songs, Kirar songs, and more. We fed these various musical arrangements into our model to boost its performance. Audio data were collected in different audio formats, and each audio file was converted to the same audio file extension, Waveform (WAV). The reason we chose the WAV format was because it is an uncompressed and lossless audio file format. It was developed by Microsoft and IBM as a tool for the Windows operating system. Because it is an uncompressed format, no information is lost from the

original audio, resulting in high-quality audio input. However, the WAV file format can be slightly larger due to the lack of compression.

Table 2. Description of each genre dataset.

No	Genre	Number of audio records	Duration of the audio in seconds	Data source and number of collections
1	Tizita Major	707	21,210 s	28 songs from the Ethiopian Orthodox church and 679 Ethiopian secular songs.
2	Tizita Minor	515	15.450 s	21 songs from the Ethiopian Orthodox church and 494 secular Ethiopian songs.
3	Ambassel Major	671	20,130 s	290 songs from the Ethiopian Orthodox church and 381 Ethiopian secular songs
4	Ambassel Minor	477	14,310 s	40 songs from the Ethiopian Orthodox church and 437 Ethiopian secular songs
5	Bati	814	24,420 s	814 Ethiopian secular songs
6	Anchihoye Lene	768	23,040 s	154 songs from the Ethiopian Orthodox church and 614 Ethiopian secular songs

3.2. Audio data preprocessing

Preprocessing is a crucial step in the data preparation process since it ensures the model's performance and accuracy. In order to remove unwanted noises, silent portions, and other irrelevant song signal details, we need to clean the audio signals. This also focuses on audio file segmentation with the same amount of time, which enables us to correctly extract the features. Thus, noise reduction and audio file segmentation are included in the preprocessing of audio data.

Converting the data into WAV format: Waveform (WAV) is an uncompressed and lossless audio file format. It was developed by Microsoft and IBM as a tool for the Windows operating system. Because it is an uncompressed format, no information is lost from the original audio, resulting in high-quality audio inputs.

Removing unwanted sounds: This involves taking out sounds from the record that are not needed, such as label ads, conversations, animal noises, dramatic sounds, and rap parts in the songs.

Silence removal: To effectively extract features, the silent portions of the audio signals below 0.04 that are present at the start, end, and any middle point are eliminated from the record throughout this process.

3.3. Audio features

Audio features are categorized into low, medium and high levels for music classification [5]. These features are discussed one by one below.

Low-level features: Several signal processing techniques are used to extract features from the different attributes of the audio signal. This process involves transforming raw audio input into a numerical dataset that contains instances and features. These characteristics are known as instantaneous features. They are derived from a short frame (or time block) of the audio signal, generating a value for each frame. There are two ways to segment frames. First, fixed length segmentation is when a frame length of 10–1000 ms is used. Second, beat synchronous segmentation

frames align with musical beats' boundaries. This is often utilized in applications such as beat tracking. They can also be called short-term or low-level features. Peeters [15] proposed the following classification for low-level features: temporal shape. Features are estimated from the waveform of the signal. Examples include attack time and effective duration. Temporal features are derived from the signal's statistical qualities. For example, they include autocorrelation and the zero-crossing rate. Energy features refer to the signal energy content. This includes global, harmonic, and noise energy. Spectral shape features are derived from the STFT. They include spectral centroid, roll-off, kurtosis, spectrogram, mel spectrogram, and MFCCs. Lastly, harmonic features are generated from sinusoidal signal modeling. Examples include harmonic noise ratio and harmonic derivation.

Mid-level features: This is another classification of features provided in [16]. The features are classified into three levels: low, mid, and high. A wide range of mid-level properties have been proposed, including the instrogram, timbregram, and chromagram. These try to link low-level features to high-level features.

High-level features: High-level features are representations of music that are not produced by the audio content of the signal. These can also be described as symbolic representations of music such as a score or sheet music with the notes arranged in staff notation. This encompasses musical aspects such as the melody, chords, harmony, key, and time signature. As a result, it is a popular method of representing music. Kirar (ክራር) or Begena (በገና) tabs, which are another common high-level representation, organizes notes in terms of Kirar or Begena frets. Another high-level representation is the musical instrument digital interface (MIDI) format. Its main characteristics define audio information, while high-level features indicate musical aspects such as melody, harmony, chords, and instruments. Mid-level features seek to bridge the gap between these two sets of features. Features are classified as low-level, mid-level, or high-level. Low-level features are extracted from the audio content, and high-level features are symbolic representations of music.

For this study, both mid-level and low-level audio features are used. Mid-level features consist of pitch, beat-related descriptors, note onsets, fluctuation patterns, and MFCCs, which are a compilation of low-level features. Low-level features are statistical attributes taken from audio files. These make sense to machines but not to people. They include the amplitude envelope, energy, spectral centroid, spectral flux, zero crossing rate, and others.

3.4. Class imbalance problem

In this research, we encountered a class imbalance problem. Certain classes were far more prevalent than others, which it usually leads to the class imbalance issue. Standard classifiers in these situations often become overloaded with the big classes and overlook the smaller ones. Classifiers' performance drastically decreases when exposed to imbalanced datasets, where the proportion of negative instances greatly exceeds that of positive examples. A number of approaches, including raising the penalty for incorrectly identifying the positive class in relation to the negative class, oversampling the majority class, and undersampling the minority class, have been put forth to address issues with class imbalance.

In this study, we encountered a class imbalance issue in the dataset, where a particular genre class had fewer records than others, as shown Figure 1. To address this, we employed various data augmentation techniques such as adjusting the sampling rate, the frequency of records, the room impulse response, the 8-bit mu-law formula, and the signal-to-noise ratio and incorporated GSM files. Ambassel Minor is a genre that musicians do not use frequently, so collecting records of it is difficult,

and there is a class imbalance in our dataset as well. Specifically, there are less data in this genre than in the other genres.

Figure 1 indicates that the Ambassel Minor genre is smaller than the others. To address this issue, we utilized an augmentation approach to enhance the audio by incorporating various effects. One method involved adjusting the sample rate, which determines the number of waveform samples taken per second [17]. This adjustment affects the range of frequencies that can be recorded in digital audio, and manipulating the sample rate is possible with most digital audio workstations. We also explored the room impulse response (RIR) to capture and analyze the acoustics of different environments [18], including their echoes and reverberations, to further augment our dataset. Additionally, we implemented the 8-bit mu-law format [19], a lossy single-channel audio compression standard, and Ogg Vorbis for its efficient quality compression across a range of frequencies [20], both of which contributed to data enhancement.

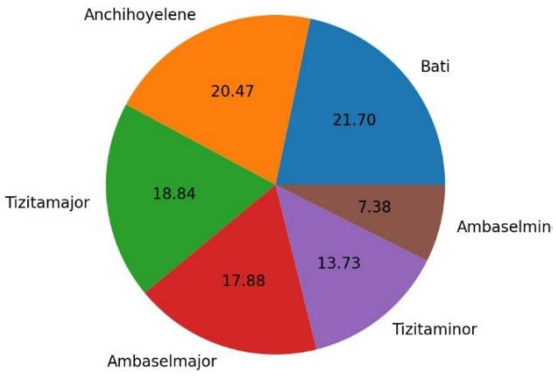

Figure 1. Dataset before augmentation.

Furthermore, we assessed the signal-to-noise ratio (SNR) by introducing various levels of noise to evaluate its impact on audio clarity [21]. Finally, we incorporated GSM files [22], which utilize a constant bitrate compression algorithm suitable for mobile applications, to round out our audio augmentation techniques.

We were able to add 200 additional audios to the Ambasel Minor genre, and we were able to address the issue of class imbalance in the data set by using the augmentation approaches mentioned above, which resulted in an increase in the Ambassel Minor genre from 7.38% to 12.07%, as shown in Figure 2.

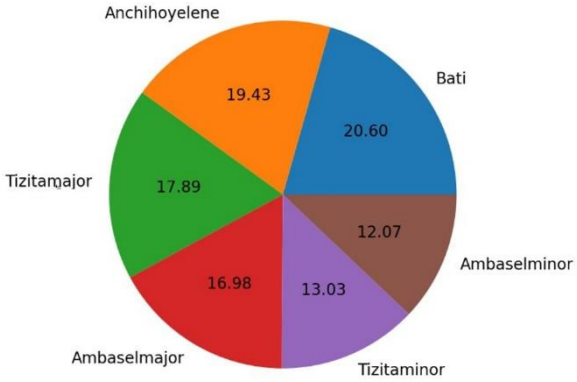

Figure 2. Audio dataset after augmentation.

3.5. Experimental setup, model development, and results

For the experimental setup of Ethiopian music genre classification using deep learning with PyTorch on Google Colab, we initiated the process by preparing our dataset. The dataset consists of various Ethiopian music genres, comprising 3952 audio files. During dataset preparation, we extracted features directly from audio data using Python to generate CSV and mel spectrogram files. The Python code we implemented captured a variety of features, including chroma STFT [23], RMSE [24], spectral centroid [25], spectral bandwidth, roll-off [26], zero crossing rate, and MFCCs [27] from MFCC1 to MFCC40, which were then saved as a CSV file. We conducted experiments with various deep learning algorithms and different types of datasets to discover more effective and efficient research methods. The algorithms utilized for model development included LSTM, CNN, and parallel RNN-CNN, along with tools such as Librosa and TensorFlow/Keras. Throughout the experiments, we extracted features from a dataset of 3952 audio files, resulting in a total of 46 audio features being employed, including chroma STFT, RMSE, spectral centroid, spectral bandwidth, roll-off, zero crossing rate, and MFCCs up to MFCC40. In this experiment, we split the audio dataset to test, validate, and training data; 70% of the data were for training, 10% were for validation, and 20% were for testing across all proposed deep learning algorithms.

For model development, we leveraged PyTorch's libraries to build various architectures, including LSTM, CNN, and hybrid RNN–CNN models [28]. We also implemented techniques such as Adam optimization and categorical cross-entropy loss for training the models. The training process involved setting hyperparameters like the learning rate, batch size, and the number of epochs. To enhance the models' generalization, we applied data augmentation strategies. During training, we monitored performance metrics such as accuracy and F1-score on the validation set. We incorporated early stopping and model checkpointing to prevent overfitting.

3.5.1. CNN model

The proposed CNN model for our study had a multi-layer architecture designed to efficiently extract and classify audio features. The model begins with an input layer that accepts preprocessed audio spectrograms generated from raw audio files of various Ethiopian music genres. The first layer consists of multiple convolutional layers with varying kernel sizes (5 × 5) to capture local patterns in the audio features. These layers are followed by batch normalization, which stabilizes the learning process and allows for the construction of deeper architectures. Interspersed between the convolutional layers are ReLU activation functions, which introduce nonlinearity, and max-pooling layers, which reduce dimensionality while enhancing feature extraction and preserving critical information. To counteract overfitting and promote generalization across unseen music genres, the model incorporates several stacked dropout layers.

After the feature extraction process, the architecture transitions into a fully connected layer that leads to a softmax output layer, producing probabilities for each genre class. The design of this CNN model aims to deliver robust performance in recognizing the diverse rhythmic and melodic characteristics intrinsic to Ethiopian music. To evaluate its effectiveness, we utilized standard metrics such as accuracy and F1-score. The CNN model processes the training data, which consists of 46 audio features. At the end of each epoch, we evaluated the performance on the validation set to measure how well the model generalizes to unseen data. After assessing the performance of the training and

validation datasets, we tested the network using the test data following the completion of training. The accuracy and loss graphs for the music genre classification generated by the CNN algorithm and its confusion matrix are shown in Figures 3 and 4, respectively. Additionally, the performance of the CNN algorithm across various metrics is summarized in Table 3.

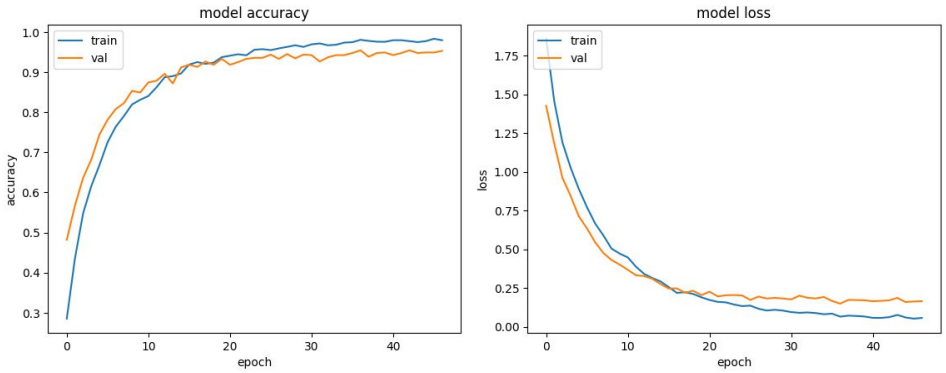

Figure 3. CNN algorithm's accuracy (left) and training loss (right).

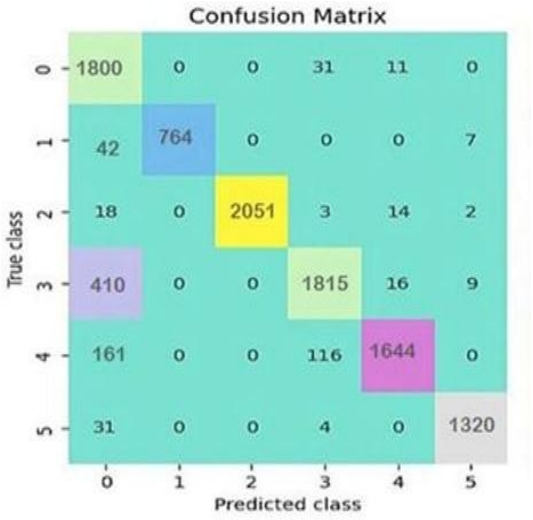

Figure 4. CNN model's confusion matrix.

Table 3. CNN algorithm's performance in each metric.

Genere	Precision	Recall	F1-score	Support
Ambassel Minor	0.99	0.68	0.81	451
Ambassel Major	1.00	0.79	0.88	170
Anchihoye Lene	0.96	0.76	0.85	529
Bati	0.51	0.99	0.67	535
Tizta Major	0.84	0.68	0.75	472
Tizita Minor	0.97	0.62	0.76	356

The performance results of the CNN algorithm, as presented in Table 3, reveal varying metrics of precision, recall, and F1-score across different musical genres. Ambassel Minor achieved a high precision of 0.99 but had a lower recall of 0.68, indicative of the model's ability to correctly identify

positive instances while missing a significant number of actual occurrences. Ambassel Major demonstrated a perfect precision of 1.00 coupled with a solid recall of 0.79, reflecting the model's reliability in identifying true positives. In contrast, the Bati genre exhibited a notable disparity with a low precision of 0.51 yet an impressive recall of 0.99, suggesting that while the model captures almost all actual instances, it also mislabels many as positive. Other genres, such as Anchihoye Lene, Tizita Minor, and Tizita Major, displayed balanced performances with varied scores, emphasizing the algorithm's strengths and weaknesses across the diverse dataset. Overall, the results highlight the need for further optimization and tuning of the CNN model to improve its consistency and efficacy in genre classification.

3.5.2. LSTM model

The LSTM model architecture for our study is specifically designed to effectively capture the temporal dependencies present in musical sequences. The input layer accepts preprocessed audio features, such as MFCCs, which are transformed into a suitable sequence format for time series analysis. The architecture begins with an embedding layer that converts the input features into a dense representation, facilitating a more effective learning process [29].

Following the embedding layer, the model comprises stacked long short-term memory (LSTM) layers, which are tailored to learn long-range dependencies within the musical data. Each LSTM layer incorporates dropout regularization to mitigate overfitting and enhance generalization. The output from the last LSTM layer is flattened and then passed into one or more fully connected (dense) layers equipped with ReLU activation functions. This culminates in a softmax output layer that predicts the probability distribution across multiple Ethiopian music genres. This architecture aims to leverage the sequential nature of music for effective genre classification.

To evaluate the model's performance, we utilized metrics such as accuracy and confusion matrices across a diverse dataset of Ethiopian musical compositions. The LSTM model processes the training data consisting of 46 audio features extracted directly from the audio data. At the end of each epoch, we assessed the performance of the validation data to determine how well the model generalizes to unseen data. We then evaluated the model's performance using test data after analyzing the results from the training and validation datasets. The accuracy and loss graphs for generic music classification generated by the LSTM algorithm and its confusion matrix are presented in Figures 5 and 6, respectively. Additionally, the performance of the LSTM algorithm across various metrics is summarized in Table 4.

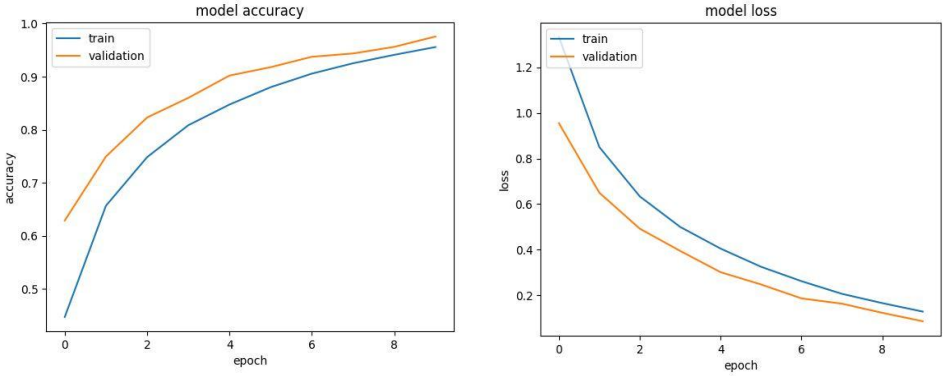

Figure 5. LSTM algorithm model's accuracy (left) and loss (right).

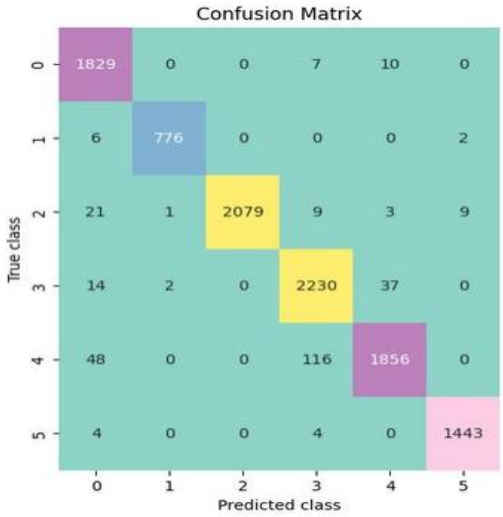

Figure 6. LSTM algorithm's confusion matrix.

Table 4. LSTM algorithm's performance in each metric.

Genere	Precision	Recall	F1-score	Support
Ambassel Minor	0.95	0.76	0.85	55
Ambassel Major	0.96	0.62	0.75	133
Anchihoye Lene	0.93	0.74	0.82	152
Bati	0.47	0.99	0.64	161
Tizta Major	0.80	0.56	0.66	140
Tizita Minor	0.95	0.56	0.70	102

The performance results of the LSTM algorithm, as presented in Table 4, demonstrate a distinct trend compared with those of the CNN model. Notably, Bati remains a challenge, with a low precision of 0.47 despite its exceptionally high recall of 0.99, signifying that the LSTM model struggles to correctly identify positive instances while capturing almost all actual instances. Conversely, Ambassel Minor and Tizita Minor show strong performance in all three metrics, with precisions above 0.95 and F1-scores of 0.70–0.85, indicating a balanced ability to identify true positives and avoid misclassifications. However, Ambassel Major and Anchihoye Lene display relatively lower precision and recall rates, while Tizta Major has a low recall, suggesting areas where the LSTM model requires improvement. In comparison with the CNN results, the LSTM model demonstrates a somewhat different set of strengths and weaknesses across the dataset, pointing to the potential benefits of exploring and combining these models for improved overall performance.

3.5.3. RNN model

The RNN model architecture for our study is designed to capture temporal patterns in audio data by processing sequences systematically. The input layer accepts the input data, formatted as sequences. For classification tasks such as audio classification, the input could be features like MFCCs or raw audio waveforms [30]. Each input sequence consists of time-ordered feature vectors that capture relevant information about the data over time. One or more recurrent layers form the core of the RNN architecture. These layers are designed to capture temporal dependencies within the input sequences.

Each recurrent layer processes the input sequences recursively, maintaining a hidden state that captures information about the history of the input up to the current time step. To prevent overfitting, dropout layers are often applied between recurrent layers. Dropout randomly deactivates a fraction of the nodes during training, promoting the model's robustness and generalization.

After the recurrent layers, the output is typically fed into one or more dense (fully connected) layers. This layer applies a linear transformation followed by a nonlinear activation function (ReLU) to transform the learned features into a form that is suitable for genre classification.

The final layer is a softmax output layer, which converts the output of the previous dense layer into probabilities for each class. Each neuron in this layer corresponds to a class label and outputs a probability indicating the model's confidence in each class. The model is trained using a loss function suitable for classification tasks, such as categorical cross-entropy, which measures the difference between the predicted probabilities and the actual labels. An optimization algorithm (Adam) is employed to update the model's weights during training, typically using backpropagation through time (BPTT) to handle the recurrent nature of the architecture.

Finally, we evaluated the validation data performance at the end of each epoch to determine the extent to which the training set model may be used to extensively generalize the unknown data. After identifying the training and validation data's performance, we utilized test data to assess the algorithm's post-training performance. The accuracy and loss graphs of music genre classification generated using the RNN algorithm and its confusion matrix are shown in Figures 7 and 8, respectively. The music genre classification performance of the RNN algorithm in each metric are also shown in Table 5.

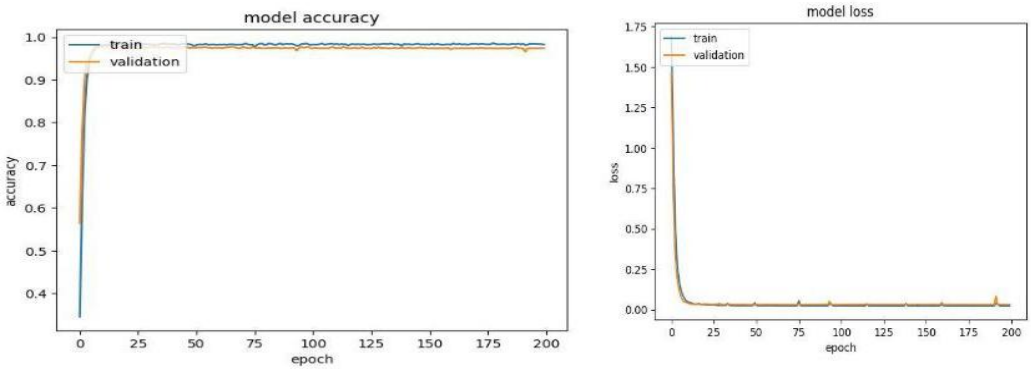

Figure 7. RNN algorithm model's accuracy (left) and loss (right).

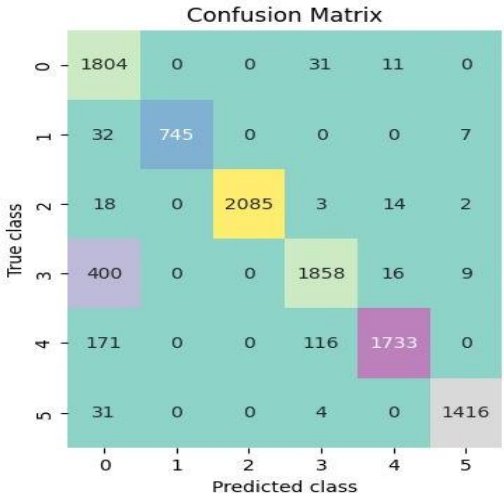

Figure 8. RNN confusion matrix.

Table 5. RNN algorithm's performance in each metrices.

Genere	Precision	Recall	F1-score	Support
Ambassel Minor	1.00	0.81	0.89	194
Ambassel Major	0.99	0.68	0.80	470
Anchihoye Lene	0.97	0.75	0.85	537
Bati	0.52	0.99	0.68	570
Tizta Major	0.85	0.71	0.78	495
Tizita Minor	0.99	0.64	0.78	360

The performance of the RNN algorithm, as outlined in Table 5, reveals a commendable overall effectiveness in genre classification across multiple metrics. Ambassel Minor stands out with a perfect precision of 1.00 and an F1-score of 0.89, indicating its robustness in identifying true positive cases while maintaining a reasonable recall of 0.81. Similarly, Anchihoye Lene displays a strong performance with a precision of 0.97 and an F1-score of 0.85, though its recall of 0.75 suggests some missed instances. While Ambassel Major achieves nearly perfect precision at 0.99, its recall of 0.68 highlights its tendency to overlook some true positives, affecting the overall F1-score (0.80). The Bati genre again shows a critical imbalance, with low precision (0.52) yet high recall (0.99), indicating a propensity to identify most instances accurately but with much misclassification. Overall, the RNN algorithm demonstrates effective classification capabilities for most genres, with particular attention necessary for models with low precision and recall discrepancies, showcasing potential areas for enhancement in future iterations.

3.5.4. RNN-CNN model

The RNN-CNN model architecture for our study combines the strengths of both recurrent and convolutional neural networks to effectively analyze and classify music audio data [31]. The model begins with an input layer that receives preprocessed audio features, such as spectrograms or MFCCs, formatted for sequential processing.

A convolutional layer follows, applying multiple filters in parallel to capture local patterns from the input features, enhanced by batch normalization and max-pooling layers to reduce dimensionality and improve feature extraction. The output of the convolutional layers is then reshaped and passed into a recurrent layer using long short-term memory (LSTM), which captures the temporal relationships and dependencies within the sequential data.

To further mitigate overfitting, dropout layers are interspersed throughout the architecture. Finally, the features processed by the recurrent layer are flattened and fed into one or more fully connected layers with ReLU activation, culminating in a softmax output layer that assigns probabilities to each Ethiopian music genre. This RNN-CNN hybrid model is designed to effectively leverage both spatial feature extraction and temporal sequence modeling, aiming to achieve high classification performance evaluated through metrics like accuracy and F1-score.

For this experiment, we separated the JSON dataset into the test, validation, and training components. These training data are then transformed by parallel RNN-CNN. After each epoch, we assessed the validation data's performance to ascertain the degree to which the training set model may be applied and broadly generalized to the unknown data. We used test data to evaluate the algorithm's post-training performance after identifying the performance on training and validation data. After training, we assessed the parallel RNN-CNN model performance using the F1-score, recall, and

precision. The findings are displayed in Figure 9. The music genre classification performance of the RNN-CNN algorithm in each metric are also shown in Table 6.

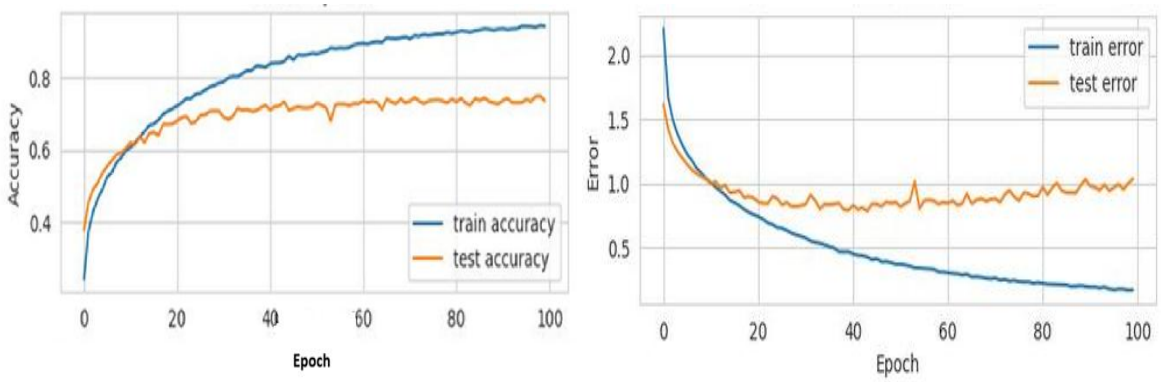

Figure 9. RNN-CNN algorithm model's accuracy (left) and loss (right).

Table 6. RNN-CNN algorithm's performance in each metric.

Genere	Precision	Recall	F1-score	Support
Ambassel Minor	0.00	0.00	0.00	55
Ambassel Major	0.00	0.00	0.00	133
Anchihoye Lene	0.93	0.78	0.85	152
Bati	0.49	0.99	0.66	161
Tizta Major	0.84	0.61	0.71	140
Tizita Minor	0.97	0.61	0.74	102

The results from Table 6 indicate the concerning performance of the RNN-CNN hybrid algorithm, particularly for the Ambassel Minor and Ambassel Major genres, which received a precision and recall of 0.00, signaling an inability to accurately predict or identify any instances within these categories despite their support sizes. Conversely, the algorithm performs reasonably well with Anchihoye Lene, achieving a precision of 0.93 and a recall of 0.78, and yielding an impressive F1-score of 0.85. The genre Bati exhibits a similar pattern observed in prior analyses, with a recall of 0.99 but a lower precision of 0.49, reflecting a tendency to identify many correct instances while also producing a significant number of false positives. Furthermore, Tizita Minor and Tizta Major show reasonable precision and weaker recall, highlighting areas for improvement. Overall, while there are some successful classifications, the hybrid model's failure to address key genres significantly hampers its effectiveness, suggesting a need for further refinement and tuning to enhance performance across the entire dataset.

3.5.5. Summary of the results

The experimental results of a genre classification model for Ethiopian music using various deep learning algorithms, including LSTM, CNN, parallel CNN-RNN, and RNN, are summarized in terms of classification accuracy. The experiments expected to identify a more accurate and efficient model through different feature extraction methods and datasets. The findings, summarized in Table 7, indicate that directly extracting features from the audio dataset outperformed using pre-extracted features from JSON and CSV datasets, with the LSTM model achieving the highest accuracy of 97%, surpassing the performance of the other algorithms tested. Direct extraction from the audio dataset

captures raw audio characteristics without the potential loss of information that can occur during quantization or data transformation in pre-extracted features. This approach retains the full spectrum of nuances in the audio signal, enabling the model to learn more effectively. Moreover, direct extraction allows for tailored feature learning that is specific to the classification task, resulting in improved performance and generalization.

Table 7. Experimental results of the proposed deep learning algorithms.

Algorithm	Dataset	Best accuracy
CNN	CSV	93%
LSTM	WAV format	97%
RNN	WAV format	94%
RNN-CNN	JSON dataset	74%

The results from Table 7 showcase varying degrees of success across different algorithms and datasets. The mid-level and low-level audio features such as chroma STFT, RMSE, spectral centroid, spectral bandwidth, roll-off, zero crossing rate, and MFCC1 to MFCC40 were used in our proposed deep learning algorithms, and the best results with their corresponding datasets are reported as shown in Table 7. The CNN model, applied to the CSV dataset, achieved a best accuracy of 93%. In contrast, the LSTM model, tested on the WAV format audio dataset, surprisingly outperformed other models with a best accuracy of 97%, indicating its robustness in handling sequential data. The RNN model also performed well on the WAV format dataset with a best accuracy of 94%. However, the RNN-CNN model, applied to the JSON dataset, yielded a relatively lower best accuracy of 74%, suggesting its potential for improvement. Overall, these results highlight the importance of selecting the right algorithm and dataset for optimal performance in audio classification tasks and demonstrate the significance of exploring different combinations of features to achieve high accuracy.

The findings indicate the significant presence of genre confusion within Ethiopian music, largely due to the overlapping acoustic and structural properties of many genres. These similarities often manifest as comparable rhythmic patterns, melodic structures, and instrumental arrangements, which can lead to higher misclassification rates in genre classification tasks. A more thorough investigation into these shared characteristics is essential, as it will illuminate the challenges encountered by classification models. Additionally, understanding these ambiguities may provide insights into potential strategies for enhancing accuracy and improving models' performance

4. Conclusion and recommendations

We collected a dataset of 3952 audio files, including 533 tracks of Ethiopian Orthodox church music and 3419 samples of secular music. The dataset was preprocessed to extract various features using Python, Librosa, and TensorFlow/Keras. Key features extracted from the audio signals included chroma STFT, RMSE, spectral centroid, and MFCCs, which were saved as CSV files. To identify the most effective model for Ethiopian music classification, experiments were conducted using different deep learning algorithms, including LSTM, CNN, parallel RNN-CNN, and RNN.

This study emphasizes the significance of exploring features and algorithms to achieve high accuracy in audio classification tasks. Notably, mid-level and low-level audio features, such as chroma STFT, RMSE, and MFCCs, were crucial to the performance of the proposed models. The results

revealed that directly extracting features from the audio dataset outperformed the use of pre-extracted features from the JSON and CSV formats.

Our findings highlight the effectiveness of using a deep learning model for categorizing Ethiopian music into six genre classes: Ancihoye Lene, Ambassel Major, Ambassel Minor, Bati, Tizita Major, and Tizita Minor. A total of 46 features, extracted at both mid-level and low-level, were centered on aspects suggested by Ethiopian music experts, emphasizing the importance of tonality. A 30-second segment of audio recordings was selected for feature extraction, resulting in datasets formatted in both CSV and JSON for further processing. The LSTM model achieved the highest accuracy of 97% when tested on the WAV format audio dataset, surpassing the performance of the other algorithms. Our experiments confirmed that it provided the best performance, utilizing 40 MFCC features extracted from the audio datasets.

For future work, our research highlights the effectiveness of low-level and mid-level musical features for numerically identifying different music genres; therefore, we recommend their application in various audio-related tasks, including speech classification and speaker detection. We also found that WAV audio formats excel in representing audio features due to their low compression rates and preservation of the original audio data, making them the preferred choice for any audio-related tasks.

Use of Generative AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

We would like to thank Mr. Haileyesus Asamenw and Mr. Ermiyas Yaiyu, both music experts, for their invaluable cooperation in identifying and labeling the dataset. Without their assistance, it would have been impossible to complete this task. We would also like to express our deepest gratitude to Yodahe Ethiopian Orthodox Church Spiritual Music School for their help in labeling the dataset and evaluating the system once it was completed. Additionally, we would like to thank Yonas Tilahun for providing use with a vast collection of Ethiopian music.

Conflict of interest

There is no conflict of interest. The audio data supporting the findings of this study are available upon request from the corresponding author, as well as on the online GitHub repository [6].

Reference

1. E. Abate, Ethiopian Kiñit (scales): analysis of the formation and structure of the Ethiopian scale system, *Proceedings of the 16th International Conference of Ethiopian Studies*, 2009, 1213–2124.
2. M. Sağun, B. Bolat, Classification of classic Turkish music makams by using deep belief networks, *Proceedings of International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2016, 1–6. https://doi.org/10.1109/INISTA.2016.7571850

3. E. Retta, R. Sutcliffe, E. Almekhlafi, Y. Enku, E. Alemu, T. Gemechu, et al., Kiñit classification in Ethiopian chants, Azmaris and modern music: a new dataset and CNN benchmark, *PLoS ONE*, **18** (2023), e0284560. https://doi.org/10.1371/journal.pone.0284560
4. F. Terefe, Pentatonic scale (kiñit) characteristics for Ethiopian music genre classification, Ph. D Thesis, Bahir Dar University, 2019.
5. A. Ramaseshan, Application of multiway methods for dimensinality reduction to music, Ph. D Thesis, Aalto University, 2013.
6. *Eshete Derb, Estifanos Tadele, Ethiopian-music-genre*, GitHub Inc., 2025. Available from: https://github.com/EsheteDerbAndEstifanosTadele/Ethiopian-Music-genre/blob/main/Ethiopian%20music%20genres.csv.
7. M. Ashraf, F. Abid, I. Din, J. Rasheed, M. Yesiltepe, S. Yeo, A hybrid CNN and RNN variant model for music classification, *Appl. Sci.*, **13** (2023), 1476. https://doi.org/10.3390/app13031476
8. L. Feng, S. Liu, J. Yao, Music genre classification with paralleling recurrent convolutional neural network, arXiv: 1712.08370. https://doi.org/10.48550/arXiv.1712.08370
9. L. Li, Audio musical genre classification using convolutional neural networks and pitch and tempo transformations, Ph. D Thesis, City University of Hong Kong, 2010.
10. N. Pelchat, C. Gelowitz, Neural network music genre classification, university of regina, *Can. J. Elect. Com.*, **43** (2020), 170–173. https://doi.org/10.1109/CJECE.2020.2970144
11. J. Tulisalmi-Eskola, Automatic music genre classification-supervised learning approach, Ph. D Thesis, Metropolia University of Applied Sciences, 2022.
12. J. Yang, Music genre classification with neural networks: an examination of several impactful variables, Ph. D Thesis, Trinity University, 2018.
13. Y. Costa, L. Oliveira, A. Koericb, F. Gouyon, Music genre recognition using spectrograms, *Proceedings of 18th International Conference on Systems, Signals and Image Processing*, 2011, 1–4.
14. B. Ismanto, T. Kusuma, D. Anggraini, Indonesian music classification on folk and dangdut genre based on rolloff spectral feature using support vector machine (SVM) algorithm, *IJCCS*, **15** (2021), 11–20. https://doi.org/10.22146/ijccs.54646
15. G. Peeters, A large set of audio features for sound description (similarity and classification) in the CUIDADO project, *CUIDADO 1st Project Report*, **54** (2004), 1–25.
16. Z. Raś, A. Wieczorkowska, *Advances in music information retrieval*, Berlin: Springer-Verlag, 2010. https://doi.org/10.1007/978-3-642-11674-2
17. R. Devi, D. Pugazhenthi, Ideal sampling rate to reduce distortion in audio steganography, *Procedia Computer Science*, **85** (2016), 418–424. https://doi.org/10.1016/j.procs.2016.05.185
18. S. Mehrotra, W. Chen, Z. Zhang, Interpolation of combined head and room impulse response for audio spatialization, *Proceedings of IEEE 13th International Workshop on Multimedia Signal Processing*, 2011, 1–6. https://doi.org/10.1109/MMSP.2011.6093794
19. R. Mohammad, M. Kumar, Audio compression using multiple transformation techniques, *International Journal of Computer Applications*, **86** (2014), 13. https://doi.org/10.5120/15043-3405
20. A. Carlacci, Ogg vorbis and MP3 audio stream characterization, Ph. D Thesis, University of Alberta, 2002.

21. C. Kim, R. Stern, Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis, *Proceedings of 9th Annual Conference of the International Speech Communication Association*, 2008, 2598–2601. https://doi.org/10.21437/Interspeech.2008-644
22. F. Abro, F. Rauf, B. Chowdhry, M. Rajarajan, Towards security of GSM voice communication, *Wireless Pers. Commun.*, **108** (2019), 1933–1955. https://doi.org/10.1007/s11277-019-06502-y
23. M. Muller, F. Kurth, M. Clausen, Chroma-based statistical audio features for audio matching, *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, 275–278. https://doi.org/10.1109/ASPAA.2005.1540223
24. C. Creusere, K. Kallakuri, R. Vanam, An objective metric of human subjective audio quality optimized for a wide range of audio fidelities, *IEEE Trans. Audio Speech*, **16** (2008), 129–136. https://doi.org/10.1109/TASL.2007.907571
25. J. Seo, M. Jin, S. Lee, D. Jang, S. Lee, C. Yoo, Audio fingerprinting based on normalized spectral subband centroids, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, 213–216. https://doi.org/10.1109/ICASSP.2005.1415684
26. V. Nandedkar, Audio retrieval using multiple feature vectors, *IJEEE*, **1** (2011), 1–5.
27. C. Ittichaichareon, S. Suksri, T. Yingthawornsuk, *Proceedings of International Conference on Computer Graphics, Simulation and Modeling*, 2012, 135–138.
28. M. Gheisari, F. Ebrahimzadeh, M. Rahimi, M. Moazzamigodarzi, Y. Liu, P. Pramanik, et al., Deep learning: applications, architectures, models, tools, and frameworks: a comprehensive survey, *CAAI Trans. Intell. Techno.*, **8** (2023), 581–606. https://doi.org/10.1049/cit2.12180
29. M. Anam, S. Defit, H. Haviluddin, L. Efrizoni, M. Firdaus, Early stopping on CNN-LSTM development to improve classification performance, *Journal of Applied Data Sciences*, **5** (2024), 1175–1188. https://doi.org/10.47738/jads.v5i3.312
30. Z. Abdul, A. Al-Talabani, Mel frequency cepstral coefficient and its applications: a review, *IEEE Access*, **10** (2022), 122136–122158. https://doi.org/10.1109/ACCESS.2022.3223444
31. K. Rezaul, M. Jewel, M. Islam, K. Siddiquee, N. Barua, M. Rahman, et al., Enhancing audio classification through MFCC feature extraction and data augmentation with CNN and RNN models, *Int. J. Adv. Comput. Sci.*, **15** (2024), 37–53. https://doi.org/10.14569/ijacsa.2024.0150704

AIMS Press