The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 6, December 2024
https://doi.org/10.33022/ijcs.v13i6.4456

Resnet-18 With Attention Mechanism-Bidirectional LSTM Hybrid Approach for Music Genre Classification Using Stacking MFCC and Mel-Spectogram Features

Dimas Elang Setyoko[1], Canggih Gelar Setyo Adhi[2], Rizwan Arisandi[3]
dimas.elang@binus.ac.id[1], canggih.gelar@binus.ac.id[2], rizwan.arisandi@binus.ac.id[3]
[1,2,3] Computer Science, School of Computer Science, Binus University, Kampus Semarang, Indonesia

Article Information

Received : 20 Oct 2024
Revised : 16 Dec 2024
Accepted : 30 Dec 2024

Keywords

Music genre classification, resnet-18, bi-lstm, CBAM attention mechanism, mfcc, melspectogram

Abstract

Incorrect Genre Classification is still often found. One of the causes is the selection of inappropriate features. This has an impact on the ability of the classifier model because some methods with a machine learning approach are highly dependent on the features used. Utilization of several features, especially spectral features, can improve the performance of the classifier model. On the other hand, methods with a deep learning approach such as CNN and RNN have been proven to outperform machine learning-based methods. This study proposes a hybrid Resnet18-BiLSTM model with the addition of the Convolutional Block Attention Module (CBAM) attention mechanism to improve the accuracy of music genre classification. Moreover, this study also combines two spectral features, namely mel-spectrogram and MFCC. The results of the experiment using the GTZAN dataset showed that the combination of mel-spectrogram and MFCC and the addition of the CBAM attention mechanism were able to classify music genres with an accuracy rate of 95.60% in validation and 95.30% in testing.

A. Introduction

Music has been an integral part of human life since ancient times, frequently used in various contexts such as religious ceremonies, rituals and entertainment. Prior to the advent of the internet technology, music was generally recorded and stored on physical media such as vinyl, magnetic tape and compact discs. With the advancement of technology, music storage has shifted from specialized physical media to digital formats, making it more accessible.

Currently, several music streaming services allow anyone to listen to music directly through the internet without having to store music files locally. The success of music streaming services is inseparable from the presence of music classification features based on genre. This feature allows users to search for music with a certain genre without having to listen to it first. However, misclassification is still often found. This happens because some music genres have similar characteristics.

Automatic classification of music genres can be done in two stages, feature extraction and classification. Feature extraction is the process of extracting hidden information from raw data [1] that represents each genre. Moreover, feature extraction can also reduce the complexity of data so that it is easier to manage and also eliminate less relevant information without eliminating important information in the data [2].

There are several popular feature extraction techniques used in audio processing. Spectral features such as Short-Term Fourier Transform (STFT), Mel-Spectogram, Mel Frequency Cepstrum Coefficient (MFCC), tempo, and chromagram have been shown to improve the accuracy of music genre classification [3]. Moreover, combining several features at once also has an impact on increasing the accuracy of music genre classification. This is proven by surveys [3] and research [4], [5] that have been conducted where methods with a combination of spectral features are better when compared to methods that use single features.

Apart from feature extraction, the accuracy of music genre classification also depends heavily on the classifier. Several algorithms with Machine Learning (ML) approaches such as K-Nearest Neighbor (KNN) [6], Random Forest [7], Multi-layer Perceptron (MLP) [8], Decision Tree [9], Support Vector Machine [10], Naive Bayes [11] and Deep Learning (DL) approaches such as Convolutional Neural Network (CNN) [12], [13], [14] have been proposed to perform music genre classification. Previous research [8] showed that CNN's capabilities are superior to several other ML-based methods, namely with an accuracy achievement of 91%. Positive results were also shown by other DL methods, namely Recurrent Neural Network (RNN) [15], where the proposed method was able to achieve an accuracy level of up to 84%.

DL approaches such as CNN and RNN have some limitations. CNN architecture is generally better suited to recognizing spatial patterns that are often found in images while RNN is better at recognizing temporal patterns in sequential data [5], [16]. Taking advantage of the advantages and disadvantages of both types of architecture, several previous studies have tried to combine CNN and RNN serially [13], [16] or in parallel [4], [5]. Each of these studies was able to produce a hybrid method that was superior when compared to a single network in terms of accuracy.

Although CNN has proven to be effective in classifying music genres using visually represented features, there are some drawbacks that need to be considered. CNN tends to treat every part of the input in the same way, without considering the relative relevance. Furthermore, unlike image classification, where objects typical of a class can occupy a dominant part of the image, in the context of music genre classification, a particular part of a genre can have a small portion in the audio recording. To overcome these limitations, an attention mechanism should be integrated into the CNN architecture. The attention mechanism can guide the CNN towards relevant important information [17] or focus on local regions of the image, enabling the model to better represent local features [18]. Adding the Convolutional Block Attention Module (CBAM) [19] can improve the performance of Resnet [20]. This is due to its ability to highlight important features on the channel and spatial sides at the same time [19].

This research is inspired by previous research, namely classifying music genres by combining CNN and RNN networks in parallel. Combining the two networks is expected to be able to learn comprehensive features for music genre classification. The Resnet architecture [21] was chosen on the CNN network side because of its advantages in overcoming the vanishing gradient problem. However, due to limited resources, the experiment will be limited to the Resnet-18 variant. On the other hand, the RNN network used to understand temporal patterns is adopted from previous research [22] which has been modified. In this study, two spectral features, namely Mel-Spectogram and MFCC, will be combined and used as input for each network. However, considering that both features can cause information redundancy [3], the CBAM attention mechanism will be added to Resnet-18.

B. Research Method

This research was conducted in several stages. Starting with collecting relevant datasets for music genre classification analysis. The collected datasets are then continued to the preprocessing and augmentation stages. This process aims to clean the dataset and add variety. After completing the preprocessing and augmentation stages, the next step is to perform feature extraction on each trimmed audio data. Feature extraction is important to convert raw audio data in the form of signals into numeric representations that can be easily analyzed by the classification model. After feature extraction is performed, the final step is to build a classification model using the CNN-RNN architecture. The prepared dataset is then trained by the CNN-RNN model and finally, the model is evaluated to measure the performance of the proposed method.

1. Data Collecting

The data used in this study is GTZAN [23] and openly available from the Zenodo Repository, DOI: https://doi.org/10.5281/zenodo.3534000. This dataset consists of 10 categories of music genres such as blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock. Each genre has 100 audio recordings with each audio lasting about 30 seconds with a sampling rate of 22050. The total data available is 1000 audio recordings.

2. Data Processing and Augmentation

Observation shows that most of the data is inconsistent in terms of duration. There are some data below and above 30 seconds. Even though the difference is quite small (a fraction of a millisecond), it can have an impact on the number of frames of the extracted feature later. Therefore, padding and trimming processes are required. All data that has a duration above or below 30 seconds will be padded first to standardize the recording duration. Next, the data will go through a trimming process by dividing each audio recording into 10 segments as shown in figure 1, thus producing new data with a shorter duration of around 3 seconds.

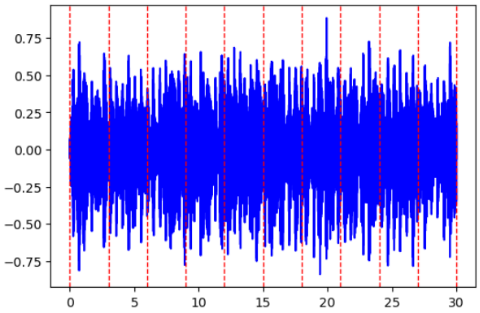

Figure1. Visualization of Trimming Audio

This is an augmentation process that is done to increase data variation. Table 1 shows the difference in data distribution of each genre before and after augmentation.

Table 1. Data Distribution Before and After Preprocessing & Augmentation

Genre	*Before*	*After*
Blues	100	1000
Classical	100	1000
Country	100	1000
Disco	100	1000
Jazz	100	1000
Metal	100	1000
Pop	100	1000
Reggae	100	1000
Rock	100	1000
Hiphop	100	1000
Total	**1000**	**10000**

3. Feature Extraction

Feature extraction using librosa library which is an open source library for audio analysis in Python language. In this study, there are two feature extraction techniques applied, namely Mel-spectogram which is a visual representation of the frequency spectrum of sound over time and MFCC which is a representation of sound signals in the cepstral frequency domain. Details of the parameters used in this study for both feature extraction techniques can be seen in Table 2.

Table 2. MFCC and Mel-Spectogram Parameters Using Librosa Library

Parameter	*Mel-Spectogram*	*MFCC*
Window length	2048	Default
Overlap length	512	Default
FFT length	2048	Default
Num bands	128	-
Mel Coefficient	-	20

Each feature produces two-dimensional data with $F_{mfcc} \in R^{20\times130}$ (128 is the number of mel filters and 130 is the number of frames) and $F_{mfcc} \in R^{20\times130}$ (20 is the number of MFCC coefficients and 130 is the number of frames). These two features are combined to produce new matrix $F_{combined} \in R^{148\times130}$. This data will later be used as input to the RNN architecture. On the other hand, the data used as input to the CNN network is an RGB image. The RGB image is a visual representation of both types of features as shown in figure 3.

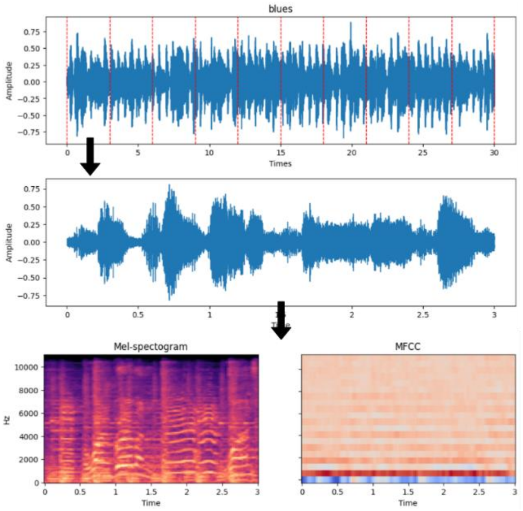

Figure3. Visual Representation of Mel-Spectogram and MFCC Feature

The size of the image generated from both features will be adjusted to 224×224×3 and combined together to produce a new image with a size of 224×224×6. All data created is then randomly selected and grouped into three different data subsets, namely train, validation and test with a proportion of 80% being laith data, 10% validation data and 10% test data.

4. Build CNN-RNN Model

Figure 4 illustrates the DL architecture proposed in this study. It can be observed that the proposed model uses two different networks and is combined in parallel. On the CNN network side, one of the Resnet variants, namely Resnet-18, is utilized to analyze spatial patterns in input images. In the initial layer, the input images $I \in R^{B\times H\times W\times C}$ where H, W, C are the height, width and image channel with values 224, 224 and 6. The input image is first processed through a convolution layer as expressed by equation 1.

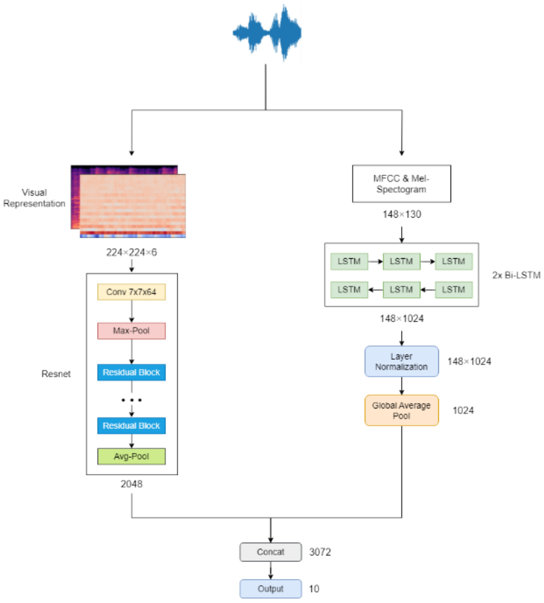

Figure4. Resnet18-BiLSTM architecture

$$X_i = f(W * I + b) \tag{1}$$

W is the parameter weight, b is the bias and the symbol $*$ indicates the convolution operation. The output of this layer is a tensor $X_1 \in R^{B \times H \times W \times C}$, which is then activated using the ReLU function (equation 2) followed by batch normalization (equation 3).

$$f(x) = \max(0, x) \tag{2}$$

$$X_{Bnorm} = \gamma \frac{X_1 - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{3}$$

Where μ and σ^2 are the mean and variance of the batch, γ and β are the trained parameters and ϵ is a constant with a small value to prevent zero division. The first convolution layer is terminated by a max-pooling operation with a size of 3×3. The entire of processes described above produce a tensor $X_1 \in R^{B \times \frac{H}{4} \times \frac{W}{4} \times C}$. The next step is to analyze the tensor on a series of residual blocks. Resnet-18 has four residual blocks formulated by equation 4. Each residual block has two convolution layers followed by batch normalization and ReLU activation. The output of the two convolution layers $F(x)$ will be summed with the identity x which is the original input of the residual block through the skip connection mechanism. However, sometimes the dimensions of x and $F(x)$ are different. Therefore, identity x needs to be transformed using a 1×1 convolution operation to equalize the dimensions.

$$Y_i = F(x) + x \tag{4}$$

After passing through all residual blocks, the tensor is processed by the global average pooling layer and produces a vector $I_{out} \in R^{512}$.
On the other hand, the RNN network receives input from the combination of MFCC and Mel-spectrogram features $F_{combined} \in R^{148\times130}$. Each element contained in the input is calculated using the following LSTM formula.

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \tag{5}$$
$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \tag{6}$$
$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \tag{7}$$
$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \tag{8}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{9}$$
$$h_t = o_t \odot \tanh\,(c_t) \tag{10}$$

Where i, f, g, o, c, h are the input, forget, cell, output gates, cell state, and hidden state at time t. The symbol $\odot$ is the Hadamard product. This study uses two layers of Bidirectional Long Short-Term Memory (Bi-LSTM) with 512 hidden states. Bi-LSTM processes information in two directions, forward and backward. The outputs of forward and backward are combined for each time t, resulting in a vector $F_{out} \in R^{1024}$ and forwarded to the normalization layer. The output of the CNN network $I_{out} \in R^{512}$ and the Bi-LSTM network $F_{out} \in R^{1024}$ are combined to produce a vector $V_{combined} \in R^{1536}$ which will then be fed forward to the Multi-Layer Perceptron (MLP) to map the vector into one of the 10 genre classes. Class determination uses the softmax function (equation 11) and the prediction results are evaluated using cross entropy (equation 12).

$$\hat{y}_i = \frac{\exp\,(z_i)}{\sum_{j=1}^{K} exp^{z_j}} \tag{11}$$
$$H(p, q) = -\sum_i^K p_i \log\,(q_i) \tag{12}$$

Where z_i, K, p_i, q_i are the input vectors, the number of genre classes, the probability of the *i*-th class and the estimated value. Training the network using several hyperparameters and Table 3 shows some of the hyperparameters used.

Table 3. Model hyperparameter

Hyperparameter	***Value***
Optimizer	Adam
Learning rate	0.001
Epoch	30
Batch size	8

5. Build Attention Mechanism

CBAM is modular and can be integrated anywhere in the convolutional layer. The input of CBAM is an intermediate feature map $F_{map} \in R^{H\times W\times C}$ and infers using two type of attention, channel attention map $M_c \in R^{1\times1\times C}$ and spatial attention map $M_s \in R^{H\times W\times 1}$. The whole process is formulated by the following eqation.

$$F' = M_c(F_{map}) \otimes F_{map} \tag{13}$$

$$F'' = M_s(F') \otimes F' \tag{14}$$

However, the placement of CBAM can affect the model performance. In this study, there are three different strategies related to the placement of the CBAM attention mechanism as shown in Figure 5.

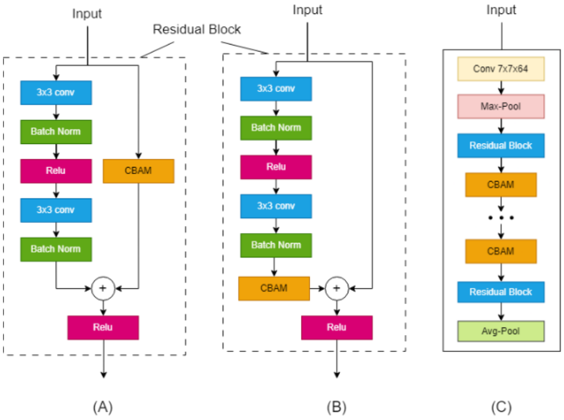

Figure5. Ilustration of different CBAM strategy. (A) CBAM in identity branch. (B) CBAM in last layer of main branch. (C) CBAM between residual block

In Figure 5 (A), CBAM is placed on the identity branch which is expected to maintain or obtain relevant information that may be lost on the main branch. The next strategy is shown in Figure 5 (B), where CBAM is placed on the last layer of the main branch. This strategy is expected to help the model to pay attention to important features that have been extracted. Then, the next strategy is shown in Figure (C) where CBAM is inserted between residual blocks. This aims to help the model in paying attention to features at various levels.

C. Result and Discussion

Experiments were conducted on three different architectures: ResNet-18, Bi-LSTM, and ResNet18-BiLSTM. Each architecture was trained using two different feature extraction strategies, namely single spectral feature and multiple spectral features. Specifically for the ResNet-18 architecture, we add observations on the influence of the CBAM attention mechanism placement strategy as illustrated in Figure 5. The experimental results of each architecture are presented in Table 4, Table 5, and Table 6. Note that the "Val" column in the table shows the best validation accuracy during the training process.

Table 4. Resnet-18 performance with different feature and attention mechanism

Model	***Attention mechanism CBAM***	***Feature***		***Acuracy***		***Parameters***
		Mel-spectogram	***MFCC***	***Val***	***Test***	
Resnet-18	(A)	✓	×	88.90%	88.69%	11.27M
		×	✓	80.10%	78.78%	
		✓	✓	**90.20%**	**89.69%**	11.28M
	(B)	✓	×	93.80%	92.29%	11.27M
		×	✓	88.50%	87.69%	
		✓	✓	**94.10%**	**93.39%**	11.28M
	(C)	✓	×	90.70%	91.59%	11.19M
		×	✓	86.20%	87.39%	
		✓	✓	**91.20%**	**91.78%**	11.20M
	×	✓	×	90.70%	91.09%	11.18M
		×	✓	86.40%	86.99%	
		✓	✓	**91.02%**	**91.59%**	11.19M

In the first part, experiments were conducted on the Resnet-18 architecture. Based on the experimental results presented in table 4, here are some conclusions that can be drawn:

1. Adding CBAM to the Resnet-18 architecture does not significantly increase the number of parameters. The average increase in the number of parameters is only about 0.38%.
2. Different CBAM placement strategies produce varying performance, ranging from 80.10% to 94.10% on the validation side and 78.78% to 93.39% on the testing side. This shows and proves that the position of CBAM in the network can affect the model performance.
3. Adding CBAM to the network does not always improve model performance. It can be observed that the performance of the model implementing the CBAM strategy (A) is unable to outperform the performance of the model without using CBAM.
4. On the input features side, the multiple features strategy tends to improve model performance. However, when comparing the single feature strategy, the melspectogram feature tends to produce the best performing model when compared to the MFCC feature.
5. The best performance was achieved when the model applied both strategies simultaneously (CBAM attention mechanism and multiple features), where the model was able to achieve an accuracy level of up to 94.10% on the validation side and 93.39% on the testing side.

Table 5. Bi-LSTM performance with different feature

Model	***Feature***		***Acuracy***		***Parameters***
	Mel-spectogram	***MFCC***	***Val***	***Test***	
Bi-LSTM	✓	×	94.20%	94.19%	1.32M
	×	✓	94.50%	93.99%	1.21M
	✓	✓	**95.90%**	**94.99%**	1.34M

Similar to the previous experiment, the results presented in table 5 show that the multiple feature strategy is able to improve the performance of the Bi-LSTM model. It is noted that the highest level of accuracy was obtained after applying the multiple features strategy, which was 91.80% on the validation side and 91.18% on the testing side. In addition, when comparing the single feature strategy, the melspectogram feature produces a model with better performance when compared to the model that only uses the MFCC feature. Then in terms of the number of parameters, the model that applies the multiple features strategy has more parameters when compared to the model that applies the single feature strategy, with an average increase rate of around 6.12%.

Table 6. Resnet18-BiLSTM performance with different feature and attention mechanism

Model	***Attention mechanism CBAM***	***Feature***		***Acuracy***		***Parameters***
		Mel-spectogram	***MFCC***	***Val***	***Test***	
Resnet18-BiLSTM	(A)	✓	×	93.60%	94.09%	12.60M
		×	✓	93.00%	94.30%	12.49M
		✓	✓	**94.60%**	**94.99%**	12.63M
	(B)	✓	×	94.70%	94.79%	12.60M
		×	✓	94.50%	93.19%	12.49M
		✓	✓	**95.60%**	**95.30%**	12.63M
	(C)	✓	×	93.50%	93.19%	12.61M
		×	✓	93.10%	93.99%	12.50M
		✓	✓	**95.40%**	**95.20%**	12.64M
	×	✓	×	**94.70%**	93.79%	12.60M
		×	✓	93.00%	93.79%	12.49M
		✓	✓	94.10%	**94.39%**	12.63M

Finally, the discussion of the experimental results on the Resnet18-BiLSTM architecture. It can be seen in Table 6 that combining two different architectures tends to improve model performance. One of the evidences can be observed when the Resnet18-BiLSTM model applies a multiple features strategy and without the addition of the CBAM attention mechanism. The model is able to produce an accuracy of 93.66% in validation and 93.47% in testing. These results are better when compared to the single Resnet-18 and Bi-LSTM architectures. Similar results are also shown in other strategies such as multiple features and the addition of CBAM, where the Resnet18-BiLSTM model is able to outperform the single architecture of Resnet-18 and Bi-LSTM. The best results from all experiments were achieved by the Resnet18-BiLSTM model by implementing the multiple feature strategy and the addition of the CBAM attention mechanism (B) with an accuracy rate of 95.60% on the validation side and 95.30% on the testing side.

D. Conclusion

This study proves that combining two spectral features simultaneously, adding the CBAM attention mechanism to the Resnet-18 architecture and combining two different architectures, namely Resnet-18 and Bi_LSTM, can create a model with very good performance in classifying music genres. This is proven by the results of the experiments conducted, where the proposed method was able to achieve an accuracy of 95.60% on the validation side and 95.30% on the testing side. We realize that the two features used in this study allow for information redundancy because the MFCC and Mel-Spectogram features are part of the STFT, so it is expected to use other feature extraction techniques.

E. Acknowledgement

Dimas Elang Setyoko contributed significantly to the conceptualization of the study, the development of the software, and the drafting of the original manuscript. Additionally, he provided critical supervision to ensure the research's methodological integrity and practical implementation.

Canggih Gelar Setyo Adhi contributed to the methodology, ensuring the research design was robust and scientifically sound. He carefully compiled the data, ensuring its quality and relevance, which is critical to the integrity of the research. His expertise in visualization allowed for the creation of clear and effective representations of the results, enhancing the comprehensibility and impact of the findings. His role was essential in ensuring the data and results were presented in a compelling and insightful manner.

Rizwan Arisandi provided essential contributions in validating the methodology and ensuring the reliability of the research outcomes. He carried out formal analyses, offering in-depth insights into the data and ensuring that interpretations were accurate and well-supported. Furthermore, he reviewed and edited the manuscript, enhancing its clarity, coherence, and overall quality. His critical eye and analytical skills were invaluable in refining the study and ensuring it met the highest academic standards.

F. References

[1] G. Krishna, C. Tran, M. Carnahan, Y. Han, and A. H. Tewfik, “Generating EEG features from Acoustic features,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1100–1104. doi: 10.23919/Eusipco47968.2020.9287498.

[2] Z. Kh. Abdul and A. K. Al-Talabani, “Mel Frequency Cepstral Coefficient and its Applications: A Review,” *IEEE Access*, vol. 10, pp. 122136–122158, 2022, doi: 10.1109/ACCESS.2022.3223444.

[3] W. Seo, S.-H. Cho, P. Teisseyre, and J. Lee, “A Short Survey and Comparison of CNN-Based Music Genre Classification Using Multiple Spectral Features,” *IEEE Access*, vol. 12, pp. 245–257, 2024, doi: 10.1109/ACCESS.2023.3346883.

[4] J. Liu, C. Wang, and L. Zha, “A Middle-Level Learning Feature Interaction Method with Deep Learning for Multi-Feature Music Genre Classification,” *Electronics*, vol. 10, no. 18, p. 2206, Sep. 2021, doi: 10.3390/electronics10182206.

[5] J. Zhang, “Music Genre Classification with ResNet and Bi-GRU Using Visual Spectrograms,” Jul. 20, 2023, *arXiv*: arXiv:2307.10773. Accessed: May 10, 2024. [Online]. Available: http://arxiv.org/abs/2307.10773

[6] Dr. S. Ponlatha, M. B, D. K. A, Kalaiyarasi. M, and Kowshika. V, “Music Genre Classification Using Deep Learning with KNN,” *IJARSCT*, pp. 224–230, Dec. 2021, doi: 10.48175/IJARSCT-2333.

[7] M. A. As Sarofi, I. Irhamah, and A. Mukarromah, "Identifikasi Genre Musik dengan Menggunakan Metode Random Forest," *JSSITS*, vol. 9, no. 1, pp. D79–D86, Jun. 2020, doi: 10.12962/j23373520.v9i1.51311.

[8] A. Ghildiyal, K. Singh, and S. Sharma, "Music Genre Classification using Machine Learning," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India: IEEE, Nov. 2020, pp. 1368–1372. doi: 10.1109/ICECA49313.2020.9297444.

[9] V. Pavan and R. Dhanalakshmi, "Analysis of Audio Data and Prediction of the Genre using Novel Random Forest and Decision Tree," in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2022, pp. 1773–1777. doi: 10.1109/ICIRCA54612.2022.9985019.

[10] A. Yuwono, C. A. Tjiandra, C. Owen, and I. B. K. Manuaba, "Music Genre Classification Using Support Vector Machine Techniques," in *2023 International Conference on Information Management and Technology (ICIMTech)*, 2023, pp. 511–516. doi: 10.1109/ICIMTech59029.2023.10277842.

[11] D. R. Ignatius Moses Setiadi *et al.*, "Comparison of SVM, KNN, and NB Classifier for Genre Music Classification based on Metadata," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2020, pp. 12–16. doi: 10.1109/iSemantic50169.2020.9234199.

[12] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Convolutional Neural Networks Approach for Music Genre Classification," in *2020 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung City, Taiwan: IEEE, Nov. 2020, pp. 399–403. doi: 10.1109/IS3C50286.2020.00109.

[13] N. Srivastava, S. Ruhil, and G. Kaushal, "Music Genre Classification using Convolutional Recurrent Neural Networks," in *2022 IEEE 6th Conference on Information and Communication Technology (CICT)*, 2022, pp. 1–5. doi: 10.1109/CICT56698.2022.9997961.

[14] S. Vishnupriya and K. Meenakshi, "Automatic Music Genre Classification using Convolution Neural Network," in *2018 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore: IEEE, Jan. 2018, pp. 1–4. doi: 10.1109/ICCCI.2018.8441340.

[15] C. Kakarla, V. Eshwarappa, L. Babu Saheer, and M. Maktabdar Oghaz, "Recurrent Neural Networks for&nbsp;Music Genre Classification," in *Artificial Intelligence XXXIX: 42nd SGAI International Conference on Artificial Intelligence, AI 2022, Cambridge, UK, December 13–15, 2022, Proceedings*, Berlin, Heidelberg: Springer-Verlag, 2022, pp. 267–279. doi: 10.1007/978-3-031-21441-7_19.

[16] M. Ashraf *et al.*, "A Hybrid CNN and RNN Variant Model for Music Classification," *Applied Sciences*, vol. 13, no. 3, p. 1476, Jan. 2023, doi: 10.3390/app13031476.

[17] Y. Hou, Q. Kong, and S. Li, "A Comparison of Attention Mechanisms of Convolutional Neural Network in Weakly Labeled Audio Tagging," in *Proceedings of the 6th Conference on Sound and Music Technology (CSMT)*, vol. 568, W. Li, S. Li, X. Shao, and Z. Li, Eds., in Lecture Notes in Electrical

Engineering, vol. 568. , Singapore: Springer Singapore, 2019, pp. 85–96. doi: 10.1007/978-981-13-8707-4_8.

[18] W. Ye, R. Tan, Y. Liu, and C.-C. Chang, "The Comparison of Attention Mechanisms with Different Embedding Modes for Performance Improvement of Fine-Grained Classification," *IEICE Trans. Inf. & Syst.*, vol. E106.D, no. 5, pp. 590–600, May 2023, doi: 10.1587/transinf.2022DLP0006.

[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," Jul. 18, 2018, *arXiv*: arXiv:1807.06521. Accessed: May 12, 2024. [Online]. Available: http://arxiv.org/abs/1807.06521

[20] M.-H. Guo *et al.*, "Attention mechanisms in computer vision: A survey," *Comp. Visual Media*, vol. 8, no. 3, pp. 331–368, Sep. 2022, doi: 10.1007/s41095-022-0271-y.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 10, 2015, *arXiv*: arXiv:1512.03385. Accessed: May 10, 2024. [Online]. Available: http://arxiv.org/abs/1512.03385

[22] N. N. Wijaya, D. R. I. M. Setiadi, and A. R. Muslikh, "Music-Genre Classification using Bidirectional Long Short-Term Memory and Mel-Frequency Cepstral Coefficients," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 243–256, Jan. 2024, doi: 10.62411/jcta.9655.

[23] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002, doi: 10.1109/TSA.2002.800560.