

Music genre classification using Inception-ResNet architecture

Fauzan Valdera, Ajib Setyo Arifin

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia

Article Info

Article history:

Received Mar 28, 2024

Revised Jun 10, 2025

Accepted Jul 10, 2025

Keywords:

Classification

Convolutional neural networks

Genre

Inception-ResNet

Music

ABSTRACT

Music genres help categorize music but lack strict boundaries, emerging from interactions among public, marketing, history, and culture. With Spotify hosting over 80 million tracks, organizing digital music is challenging due to the sheer volume and diversity. Automating music genre classification aids in managing this vast array and attracting customers. Recently, convolutional neural networks (CNNs) have been used for their ability to extract hierarchical features from images, applicable to music through spectrograms. This study introduces the Inception-ResNet architecture for music genre classification, significantly improving performance with 94.10% accuracy, precision of 94.19%, recall of 94.10%, F1-score of 94.08%, and 149,418 parameters on the GTZAN dataset, showcasing its potential in efficiently managing and categorizing large music databases.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ajib Setyo Arifin

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia

Depok 16424, Indonesia

Email: ajib.sa@ui.ac.id

1. INTRODUCTION

Music genre is a label used by humans to categorize and describe the characteristics of a music. Music genres do not have strict definitions and boundaries as they emerge through complex interactions between the audience, marketing, history, and culture [1]. Observations on music genres have led some researchers to propose new classification definitions purely for the purpose of information retrieval from music [2]. However, with the existing music genres, it is clear that certain genres have characteristics typically associated with instrumentation, rhythmic structure, and musical content.

The process of extracting information from music is becoming increasingly important in organizing and managing the vast amount of digitally available music files on the internet. However, this process has become nearly impossible to be done manually by humans due to the continuously increasing and diverse number of digital music. Therefore, automated music genre classification has become one of the services that will assist music distribution vendors in organizing the multitude of music files and leveraging music information to attract customers.

During the past decade, there has been a surge in the use of convolutional neural network (CNN) architectures, which have achieved satisfactory performance in the field of image recognition [3]. CNNs can effectively extract information from an image due to their hierarchical structure [4]. Low-level features, such as basic textures, are built into high-level semantic information through CNN layers [5]. The specific capabilities of CNNs can assist in tasks such as music classification by leveraging information from spectrograms, which contain texture information from music signals.

Liu *et al.* [6] proposed an architecture called bottom-up broadcast neural network (BBNN), which adopts a relatively wide and shallow structure. The main idea behind the BBNN architecture is to develop effective blocks and different block-to-block connections to exploit and preserve low-level information to

higher layers. The architecture is designed in such a way that spectrogram information at the lower level can participate in decision-making layers throughout the network. Therefore, BBNN is equipped with a broadcast module (BM) consisting of InceptionV1 blocks and dense connectivity. They reported an accuracy of 93%, indicating an improvement over previous CNN architectures. However, there are several drawbacks to the BBNN, including the use of InceptionV1 blocks in the BM. InceptionV1 has a high computational cost due to the use of large filters, specifically a 5×5 filter. BBNN adopts a relatively shallow structure [6]. This can limit its capacity to capture complex features and representations, especially for tasks that require depth and higher-level hierarchical information processing, such as music classification. The use of max-pooling with a large window size, specifically (4, 1), in shallow layers. Down-sampling using max-pooling with a large window size can drastically reduce the input dimensions, resulting in lost information and potential accuracy degradation [7].

In 2016, Szegedy *et al.* [8], Inception-v4 and Inception-ResNet, combining Inception modules with residual connections to improve deep learning efficiency. Inception-v4 refines the original Inception architecture, while Inception-ResNet integrates residual connections to enhance gradient flow and training speed. Experiments show that Inception-ResNet trains faster and achieves comparable or better accuracy than traditional Inception networks. The study highlights how residual connections mitigate the vanishing gradient problem, leading to more stable learning. Overall, the research demonstrates that combining Inception modules with residual learning results in highly accurate and computationally efficient deep networks. This research aims to improve the accuracy performance and reduce the computational complexity of the BBNN architecture. The researchers propose music genre classification using the Inception-ResNet architecture with input in the form of mel-spectrograms of audio signals.

2. LITERATURE REVIEW

Over recent years, the classification of music genres through visual representations like spectrograms short-time Fourier transform (STFT) and mel-spectrogram, mel-frequency cepstral coefficients (MFCC) has seen significant advancements. These visual methods leverage traditional texture descriptors from computer vision such as local phase quantization, local binary patterns, and Gabor filters to encapsulate the spectrograms' content, which resembles temporal energy distribution changes across frequency bins. Despite the traditional classification techniques, including support vector machine (SVM) and Gaussian mixture models (GMM), outperforming human accuracy (70%) on various music datasets, they are still heavily reliant on feature engineering [9].

Deep neural networks have significantly reduced the reliance on task-specific prior knowledge, achieving notable successes in computer vision [10], [11] and inspiring applications in music genre classification [12], [13]. Pioneering work by Lee *et al.* [14], a deep learning framework for audio classification was introduced, employing a convolutional deep belief network to learn from spectrograms, inspiring further research in using deep learning for audio recognition. Previous researchers in [15], [16], innovated by stacking hidden layers and employing different activation functions and classifiers, achieving up to 84% accuracy on the GTZAN dataset [17]. Despite these advancements, challenges remain in feature learning without classifier supervision, impacting the prediction capabilities of the models and maintaining a two-stage process in the framework.

Recent advancements in music genre classification have seen the integration of feature learning and classification into a single stage, primarily using CNN-based methods. Jakubik [18] introduced recurrent neural network (RNN) architectures, specifically long short-term memory (LSTM), and gated recurrent unit (GRU), from the image domain to music analysis, achieving remarkable accuracies of 91 and 92% on the GTZAN dataset, showcasing their efficacy. The NNet2 model introduced a novel CNN architecture with shortcut connections to all layers, enhancing learning capacity through a combination of max and average pooling, and achieved an 87% accuracy on GTZAN [19]. Additionally, to address the varying significance of different temporal segments in music, the studies in [3], [20], [21] incorporated an attention mechanism with a bidirectional RNN, a technique further refined by integrating stacking attention modules in subsequent work, emphasizing the evolving focus on nuanced temporal analysis in music content. The most recent work was conducted by Liu *et al.* [6] introduced the BBNN, a model featuring a wide and shallow architecture designed to efficiently utilize low-level spectrogram information across its decision-making layers. This network incorporates a BM with InceptionV1 blocks and dense connections, aiming to enhance the flow of information from lower to higher layers. Despite achieving a 93% accuracy, surpassing many conventional CNN architectures, the BBNN faces challenges such as the computationally expensive use of InceptionV1 blocks, its shallow structure which might limit the extraction of complex features necessary for music classification, and the use of max-pooling with large window sizes that may lead to significant information loss and potential reductions in accuracy. To overcome these drawbacks, in this article an Inception-ResNet module is proposed.

3. PROPOSED ARCHITECTURE

We propose the use of Inception-ResNet blocks to replace the InceptionV1 blocks and make modifications to the shallow layers in the BBNN model [6]. The Inception-ResNet block is designed using the TensorFlow library and consists of a total of 89 layers, including several components such as the stem module (5 layers), reduction module (23 layers), Inception-ResNet module (54 layers), and the fully connected module (7 layers). Figure 1 shows the proposed architecture using Inception-ResNet blocks. The stem module serves as a feature extraction component at the beginning of the network. The stem model is responsible for processing the input data and extracting meaningful features that will be further used by the subsequent layers. Figure 2 shows the schematic of the stem module.

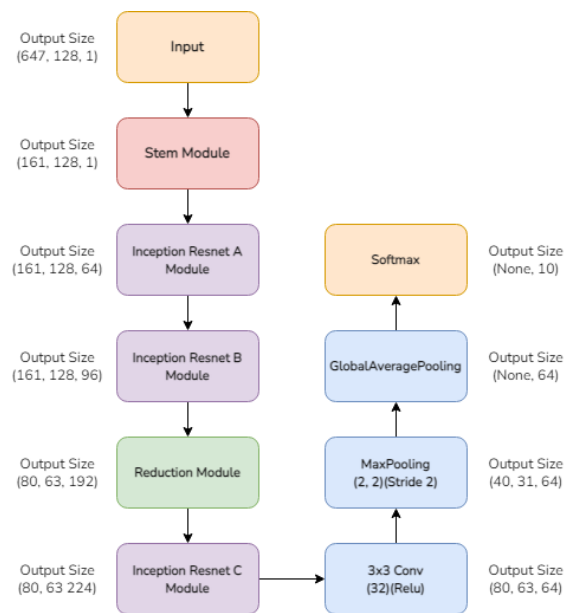


Figure 1. Proposed architecture using Inception-ResNet blocks

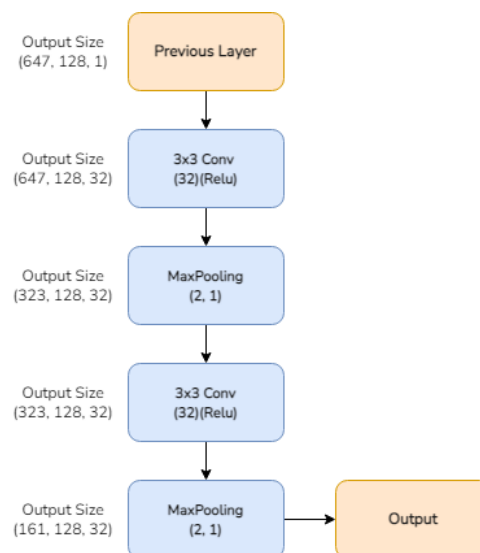


Figure 2. Stem module

The stem module consists of two downsampling stages, reducing the dimensions from (647, 128) to (323, 128), and then from (323, 128) to (161, 128). In each downsampling stage, there is a 3×3 convolutional

layer with rectified linear unit (ReLU) activation. The convolutional layers extract initial features that serve as the foundation for the subsequent layers. The use of max-pooling with a size of (4, 1) in the BBNN model reduces the input dimension by a quarter. This can potentially eliminate some important features or detailed spatial information, especially when the input size is relatively small. In the proposed stem module, the use of two stages of max-pooling with a size of (2, 1) achieves a good balance between reducing the input dimension and retaining important information in the feature representation.

In the proposed architecture, three Inception-ResNet modules are used: Inception-ResNet A (22 layers), Inception-ResNet B (16 layers), and Inception-ResNet C (16 layers). The scheme of the Inception-ResNet module was introduced by Szegedy *et al.* [8]. Figures 3 and 4 show the Inception-ResNet A, Inception-ResNet B, and Inception-ResNet C modules, respectively. In Figure 3, the Inception-ResNet A module functions to extract features at the initial stage. The module consists of three branches with a combination of 1×1 and 3×3 convolutions. Each output from the branches is merged through a 1×1 convolution with a linear activation function. This layer is called the activation scale, which adjusts the magnitude of the module's output adaptively.

In Figure 4(a), the Inception-ResNet B module functions to extract features at the intermediate stage. The module consists of two branches with a combination of 1×1 , 1×7 , and 7×1 convolutions. The use of 1×7 and 7×1 filters instead of a 7×7 filter is conducted to reduce the total computations in the module. In Figure 4(b), the Inception-ResNet C module has the same configuration scheme as the Inception-ResNet B module, which functions to extract features at the final stage. The module consists of two branches with a combination of 1×1 , 1×3 , and 3×1 convolutions.

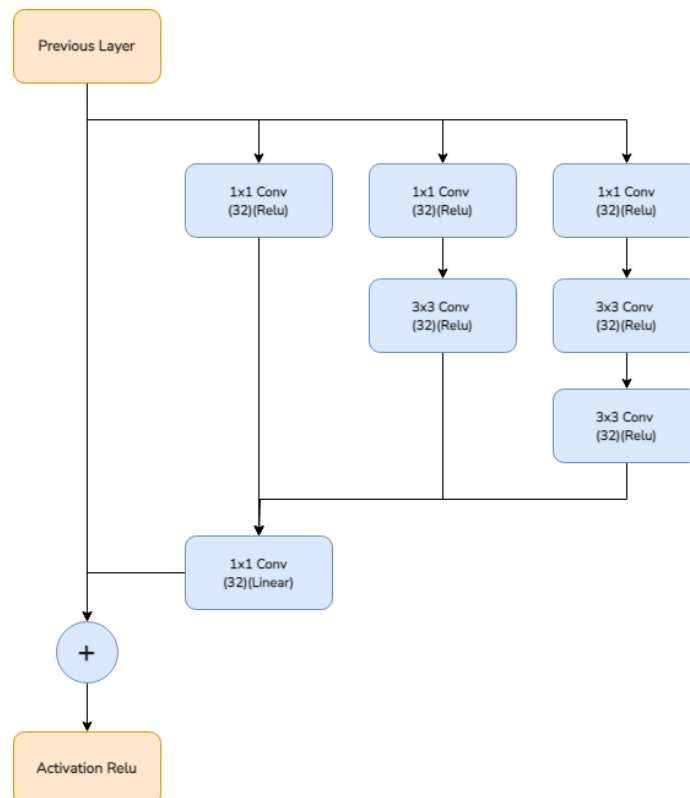


Figure 3. Inception-ResNet A module

Each Inception-ResNet module has a different number and size of filters, allowing for more control over the capacity and complexity of the model. There is a difference between the Inception-ResNet modules and the Inception modules in the BBNN model, where there is no use of convolutional layers with large filters, such as 5×5 . Convolutions with large filters are replaced with two convolutions with filters (1×3 and 3×1) or (1×7 and 7×1) to reduce the total number of parameters. This allows for the creation of deeper models. The reduction module aims to reduce the dimension of the features while preserving and enhancing important information. Figure 5 shows the scheme of the reduction module. To reduce the size of the feature dimension, a convolutional layer with a stride of two is used, which reduces the feature dimension by half of its original size.

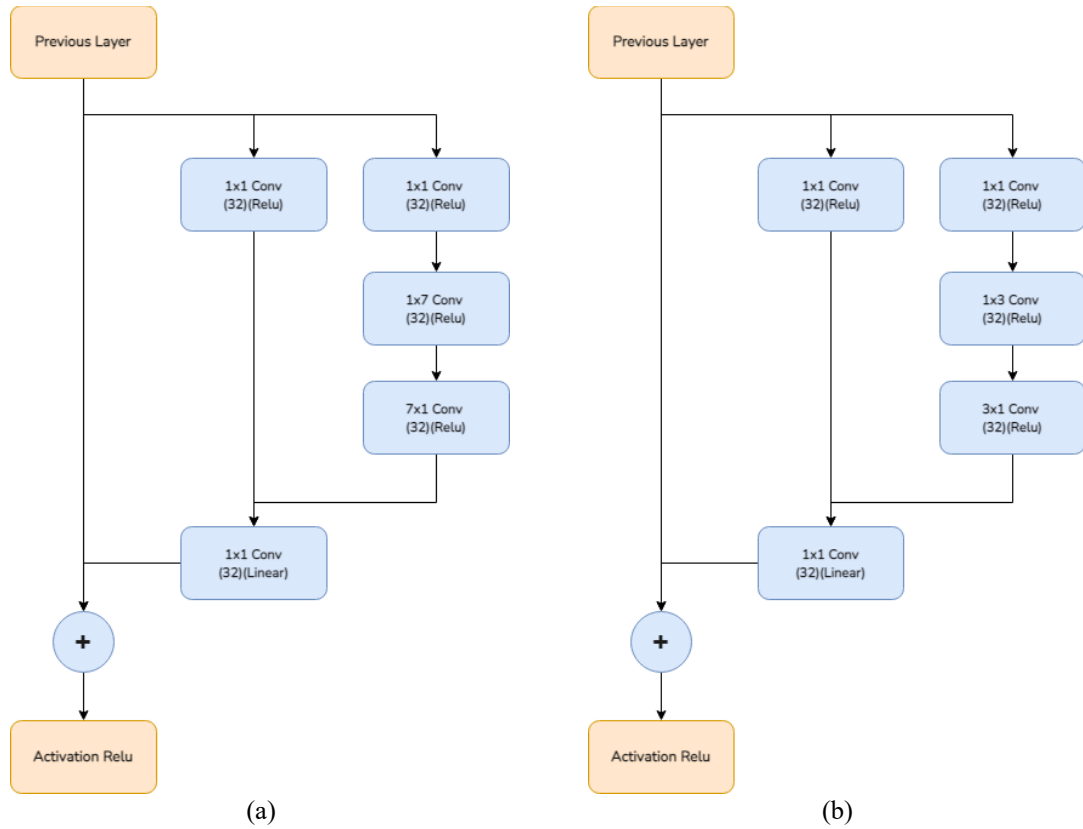


Figure 4. The layer architecture for (a) Inception-ResNet B module and (b) Inception-ResNet C module

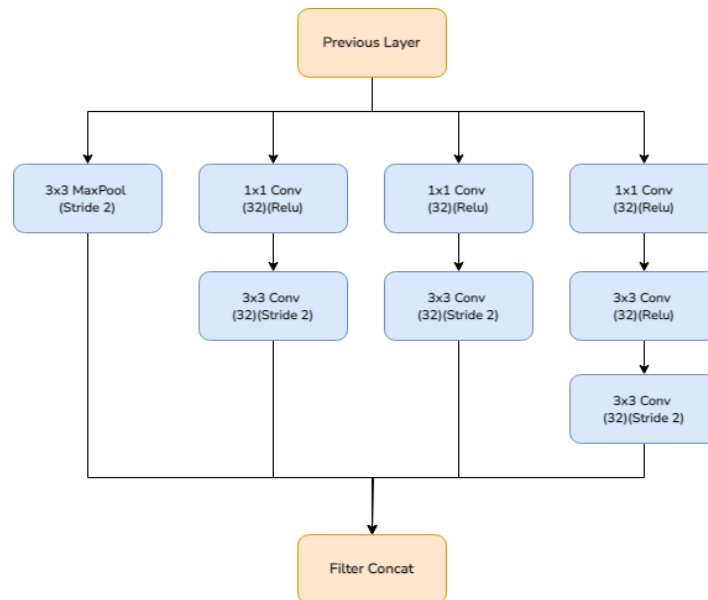


Figure 5. Reduction module

4. EXPERIMENTAL SETUP

4.1. Dataset

The dataset used in this study is the GTZAN dataset. GTZAN is the most widely used public dataset for evaluation in music genre recognition (MGR) research [22]. The files were collected between 2000-2001 from various sources to represent various music recording conditions, such as personal CDs, radio, recordings, and microphones. GTZAN contains 1,000 music tracks in .wav format, each lasting for

30 seconds [23]. GTZAN consists of 10 genres, including blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock [22].

4.2. Preprocessing

The audio dataset with a duration of 30 seconds will be processed into mel-spectrogram form. There are 1,000 audio samples consisting of 10 music genres, with 900 samples used for training and 100 samples used for testing. To process the audio dataset into mel-spectrograms, we need to perform STFT, mel-scaling, and triangle filtering, all of which are available in the Python library called Librosa. There are several parameters used in audio preprocessing, including a window length for Fourier transform of 512 samples, a hop length (number of samples between frames) of 1,024 samples, and a total of 128 mel bands for mel-scaling. The preprocessing scheme can be seen in Figure 6. After going through the audio preprocessing process, the result is obtained in the form of mel-spectrograms with dimensions of 128×647.

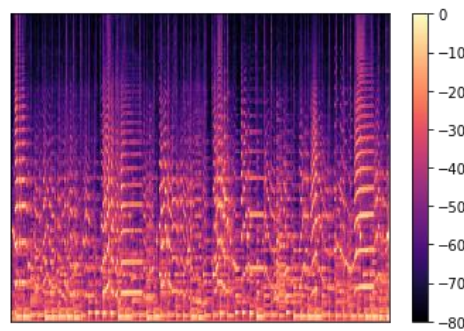


Figure 6. Mel-Spectrograms by preprocessing

4.3. Training and testing

The process of designing the architecture, training, and testing was performed on Google Collaboratory Pro version using a Tesla P100 GPU, 15 GB of RAM, and 2 CPU cores. The model training stage was conducted for 100 epochs using the Adam optimizer with a batch size of 8. An initial learning rate of 0.01 was established and was automatically reduced by a factor of 0.5 if the loss did not decrease for three consecutive epochs. Additionally, the training stage implemented an early stopping mechanism, which stopped the training when the monitored loss did not decrease for 5 epochs. The categorical cross-entropy loss function was used to calculate the loss value for the multiclass classification model.

The model training employed the K-fold cross-validation method. K-fold cross-validation is a commonly used technique in machine learning for more objective model performance evaluation [24]–[26]. In K-fold cross-validation, the dataset is randomly divided into k balanced subsets called folds. The K-fold cross-validation method helps address the uncertainty issue in model evaluation caused by variations in the training and testing data splits. By combining evaluations from k independent iterations, we obtain a more objective overview of the model's performance. In this training, k was set to 10 for the K-fold cross-validation method, and stratified sampling was used to ensure balanced data.

5. RESULTS AND ANALYSIS

5.1. Result

The training process of the Inception-ResNet architecture follows the experimental scheme described in section 4. In each training iteration or epoch, the model is evaluated using validation data that is not used for training the model. Two evaluation metrics are used: accuracy and loss. The main goal of these evaluation metrics is to measure how well the model generalizes and predicts with high accuracy on unseen data. The evaluation metric values obtained from the training process of 100 epochs using the TensorFlow library are shown in Figure 7. Figure 7(a) illustrates the accuracy comparison graph with epoch iterations, where accuracy represents the percentage of correctly predicted data out of the total data. Figure 7(b) displays the loss comparison graph with epoch iterations, where loss is the result of the categorical cross-entropy loss function.

After the training process, the Inception-ResNet model is evaluated using several metrics to assess its performance. The evaluation metrics used for the music genre classification task include accuracy, precision, recall rate, and F1-score. Table 1 shows the evaluation metrics calculated by averaging the accuracy, precision, recall rate, and F1-score metrics from the results of the 10-fold cross-validations process. To assess the model's

performance in predicting audio for each music genre, separate metric calculations are performed for each genre, each consisting of 100 test data, as shown in Table 2. Additionally, a confusion matrix is generated to illustrate the number of correct and incorrect predictions for each category, as shown in Figure 8.

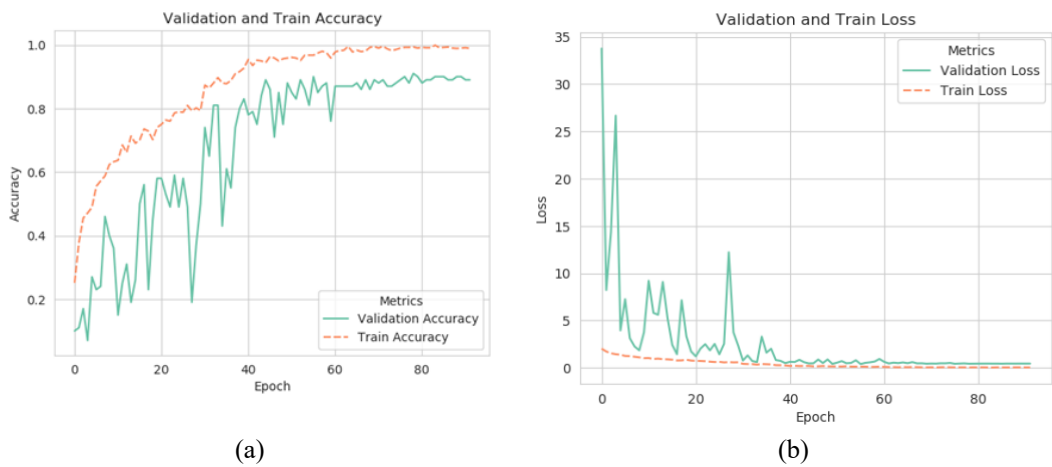


Figure 7. Evaluation metrics result of (a) validation and train accuracy and (b) validation and train loss

Table 1. Evaluation metrics			
Evaluation metrics (%)			
Accuracy	Precision	Recall rate	F1-score
94.10	94.10	94.19	94.08

Table 2. Evaluation metrics per genre				
Genre	Evaluation metrics (%)			
	Accuracy	Precision	Recall rate	F1-score
Blues	94.0	95.9	94.0	94.9
Classical	99.0	98.1	99.0	98.5
Country	95.0	90.4	95.0	92.7
Disco	90.0	92.7	90.0	91.3
Hip-Hop	98.0	98.9	98.0	98.9
Jazz	94.0	95.9	94.0	95.8
Metal	98.0	92.4	98.0	92.4
Pop	96.0	88.1	96.0	91.8
Reggae	91.0	96.8	91.0	93.8
Rock	86.0	92.5	86.0	89.1

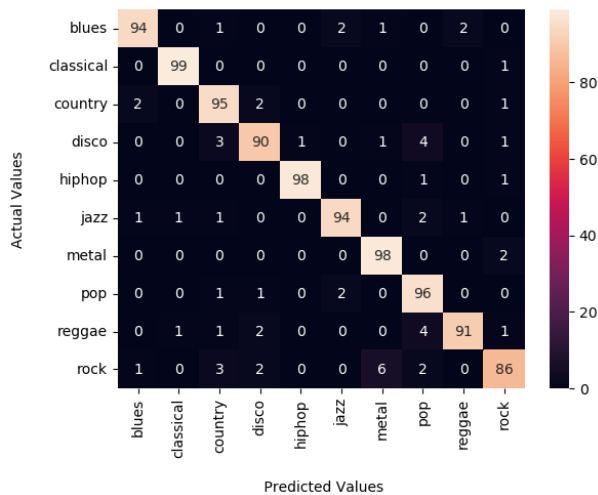


Figure 8. Confusion matrix

The number of parameters in the model refers to the number of weights or parameters that need to be updated during the model training process. These parameters are values set by the model and used for computations during training. Table 3 shows the number of parameters in the Inception-ResNet architecture obtained from the simulation results using the TensorFlow library. Trainable parameters refer to the parameters that change during the training process, including the weights and biases in each convolutional layer and fully connected layer. Non-trainable parameters refer to the parameters that does not change during the training process, including global and constant parameters. Total parameters refer to the sum of trainable parameters and non-trainable parameters. The results of music genre classification prediction using the Inception-ResNet architecture are shown in Figure 9.

Table 3. Total parameters

Parameters	Value
Trainable parameters	147,818
Non-trainable parameters	1.600
Total parameters	149.418

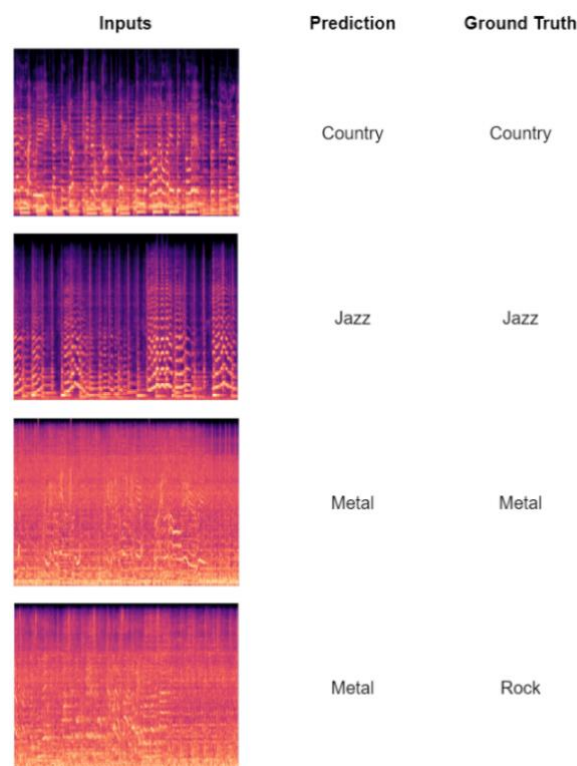


Figure 9. Model predictions

5.2. Analysis

The Inception-ResNet model was evaluated using several evaluation metrics to assess its performance. The evaluation metrics used for music genre classification include accuracy, precision, recall rate, and F1-score. These evaluation metrics consist of four components: true positive (TP), true negative (TN), false negative (FN), and false positive (FP). In the case of music genre classification, the input is transformed into binary vectors using one-hot encoding. Each unique category or level of the categorical variable is represented by a binary vector, where the length of the vector is equal to the number of unique categories. For each data point, only one element in the vector is positive (denoted by 1), indicating the corresponding category, while the other elements are negative (denoted by 0).

Accuracy is a commonly used evaluation metric to measure the performance of a classification model. It calculates the percentage of correct predictions (TP and TN) out of all predictions made by the model. According to Table 1, the accuracy obtained from the model using the test data is 94.10%. This accuracy value indicates that the model performs well and accurately predicts the music genre overall.

However, based on Table 2, there are genres with accuracy under 90%, such as rock, indicating that the model is less accurate in predicting audio with that genre. The confusion matrix in Figure 8 shows that the model misclassified some rock audio as metal, which can be attributed to the similarities between rock and metal genres.

Precision is an evaluation metric used to measure how accurately a classification model identifies positive predictions. It calculates the percentage of TP predictions out of all positive predictions made by the model. According to Table 1, the precision obtained from the model using the test data is also 94.10%. This precision value indicates that the model has a high level of accuracy in classifying audio into their respective classes. However, based on Table 2, there are genres with precision under 90%, such as pop, indicating that the model tends to make mistakes by predicting audio that should not belong to that genre as members of that genre.

Recall rate, also known as sensitivity or TP rate, is an evaluation metric used to measure how well a classification model correctly identifies the overall number of positives. Recall calculates the percentage of TP predictions out of the total number of actual positives. According to Table 1, the recall rate obtained from the model using the test data is 94.19%.

F1-score is an evaluation metric used to combine information about precision and recall into a single number that describes the overall performance of a classification model or selection system. F1-score measures how well the model can achieve a balance between precision and recall. According to Table 1, the F1-score obtained from the model using the test data is 94.08%.

The number of parameters in a model is directly related to computational cost. The more parameters a model has, the more complex it is, and the more computational operations are required during training or prediction. According to Table 3, the total number of parameters obtained is 149,418.

The training graphs in Figure 7 show a consistent increase in validation accuracy and a decrease in validation loss throughout the training process. This indicates that the model can generalize well, as it achieves good performance not only on the training data but also on the validation data. Additionally, there is no significant difference between the validation accuracy and train accuracy values at the final epoch, indicating that the model does not suffer from overfitting.

5.3. Performance comparison Inception-ResNet and bottom-up broadcast neural network

The evaluation metrics of the trained Inception-ResNet model are compared with the BBNN model. Table 4 presents a comparison of the performance between these two models, which have undergone the same training process and dataset pre-processing [6]. Based on Table 4, the proposed architecture model has higher values in each metric and a smaller total number of parameters compared to the BBNN architecture. The proposed architecture utilizes Inception-ResNet modules, which have fewer parameters compared to the InceptionV1 modules used in the BBNN architecture. This allows the authors to create a deeper architecture to enhance the model's capacity in capturing complex features and representations from mel-spectrograms.

Table 4. Performance comparison

Model	Total parameter	Evaluation metric (%)			
		Accuracy	Recall	Precision	F1-score
Inception-ResNet	149,418	94.10	94.10	94.19	94.08
BBNN [5]	185,642	93.90	94.0	93.7	93.7

The reduction in the number of parameters in the Inception-ResNet modules is due to the absence of using convolutional layers with larger filters, such as 5×5 . Instead, the large filters are replaced with two convolutions using filters of size $(1 \times 3$ and $3 \times 1)$ or $(1 \times 7$ and $7 \times 1)$ to reduce the computational cost or total number of parameters. Figures 3 and 4 illustrate the Inception-ResNet modules used in the proposed architecture.

In the BBNN architecture, a single stage of max-pooling with a size of $(4, 1)$ is used, which directly reduces the input dimension by a quarter. This may lead to the loss of some important features or detailed spatial information. On the other hand, in the proposed Inception-ResNet architecture the use of two stages of max-pooling with a size of $(2, 1)$ strikes a good balance between reducing the input dimension and retaining important information in the feature representation. The lower total number of parameters can improve the performance of music genre classification on devices with limited computational resources, such as mobile phones. Additionally, the higher accuracy can enhance the performance of applications that utilize the genre classification model, such as recommender systems, data pipelines, and other applications.

6. CONCLUSION

In this article, we propose a music genre classification framework based on the Inception-ResNet architecture. The proposed framework can capture complex features with a deeper architecture in mel-spectrograms. Even with a smaller number of parameters, the proposed framework manages to outperform the existing model on all measurement metrics including accuracy, recall, precision, and F1-score, based on GTZAN datasets. With a smaller number of parameters, the proposed framework can potentially be applied to devices with limited computational resources.

FUNDING INFORMATION

This work was supported in part by the Universitas Indonesia under Grant PUTI Q2 NKB-796/UN2.RST/HKP.05.00/2023 and Grant LK NKB-2582/UN2.F4.D/PPM.00.00/2023.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Fauzan Valdera	✓	✓	✓					✓	✓		✓			
Ajib Setyo Arifin	✓	✓		✓	✓	✓	✓			✓	✓	✓	✓	✓

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings are available from the corresponding author [ASA] on request.




REFERENCES

- [1] G. Cerati, "Difficult to define, easy to understand: the use of genre categories while talking about music," *SN Social Sciences*, vol. 1, no. 12, 2021, doi: 10.1007/s43545-021-00296-2.
- [2] M. Genussov and I. Cohen, "Musical genre classification of audio signals using geometric methods," in *2010 18th European Signal Processing Conference*, 2010, pp. 497–501.
- [3] Y. Wang, X. Lin, L. Wu, and W. Zhang, "Effective multi-query expansions: collaborative deep networks for robust landmark retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1393–1404, 2017, doi: 10.1109/TIP.2017.2655449.
- [4] L. G. Hafemann, L. S. Oliveira, and P. Cavalin, "Forest species recognition using deep convolutional neural networks," in *International Conference on Pattern Recognition*, 2014, pp. 1103–1107, doi: 10.1109/ICPR.2014.199.
- [5] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, 2017, pp. 141–149, doi: 10.5281/zenodo.1418015.
- [6] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7313–7331, 2021, doi: 10.1007/s11042-020-09643-6.
- [7] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: the all convolutional net," in *arXiv-Computer Science*, pp. 1–14, Apr. 2015.
- [8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017, pp. 4278–4284, doi: 10.1609/aaai.v31i1.11231.
- [9] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing Journal*, vol. 52, pp. 28–38, 2017, doi: 10.1016/j.asoc.2016.12.024.
- [10] G. Huang, Z. Liu, and L. van der Maaten, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [12] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015, doi: 10.1109/TMM.2015.2478068.
- [13] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, 2017, doi: 10.1109/LSP.2017.2713830.




- [14] H. Lee, L. Yan, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems* 22, 2009, pp. 1096–1104.
- [15] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010, pp. 339–344.
- [16] S. Sigita and S. Dixon, "Improved music feature learning with deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 6959–6963, doi: 10.1109/ICASSP.2014.6854949.
- [17] T. L. H. Li, A. B. Chan, and A. H. W. Chun, "Automatic musical pattern feature extraction using convolutional neural network," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2010)*, 2010, pp. 546–550.
- [18] J. Jakubik, "Evaluation of gated recurrent neural networks in music classification tasks," *Advances in Intelligent Systems and Computing*, vol. 655, pp. 27–37, 2018, doi: 10.1007/978-3-319-67220-5_3.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [20] Y. Yu, S. Luo, S. Liu, H. Qiao, Y. Liu, and L. Feng, "Deep attention based music genre classification," *Neurocomputing*, vol. 372, pp. 84–91, 2020, doi: 10.1016/j.neucom.2019.09.054.
- [21] Q. H. Nguyen *et al.*, "Music genre classification using residual attention network," in *2019 International Conference on System Science and Engineering (ICSSE)*, Dong Hoi, Vietnam, 2019, pp. 115–119, doi: 10.1109/ICSSE.2019.8823100.
- [22] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002, doi: 10.1109/TSA.2002.800560.
- [23] B. L. Sturm, "The state of the art ten years after a state of the art: future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, 2014, doi: 10.1080/09298215.2014.894533.
- [24] Y. Zhang and Y. Yang, "Cross-validation for selecting a model selection procedure," *Journal of Econometrics*, vol. 187, no. 1, pp. 95–112, 2015, doi: 10.1016/j.jeconom.2015.02.006.
- [25] T. T. Wong and N. Y. Yang, "Dependency analysis of accuracy estimates in k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2417–2427, 2017, doi: 10.1109/TKDE.2017.2740926.
- [26] T. T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015, doi: 10.1016/j.patcog.2015.03.009.

BIOGRAPHIES OF AUTHORS



Fauzan Valdera    received the Bachelor in Electrical Engineering from the Universitas Indonesia in 2023. His research area is machine learning. He can be contacted at email: fauzan.valdera@ui.ac.id.



Ajib Setyo Arifin    received the Bachelor in Electrical Engineering and Master's Degree from the Universitas Indonesia, in 2009 and 2011, respectively. He has got the Ph.D. degree in Telecommunications in 2015 from the Keio University, Japan. He is an associate professor at Universitas Indonesia. He was a head of telecommunication laboratory in Department of Electrical Engineering. His research areas include wireless sensor networks, wireless communication, signal processing for communication and machine learning. He can be contacted at email: ajib.sa@ui.ac.id.