International Journal of Informatics and Computation (IJICOM)
Vol. 7, No. 2, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Enhancing Music Genres Classification with MFCC and CNN

Bulkis Kanata[1], Sudi M. Al Sasongko[2], Mujni Ahmad Ali[3]

Abstract
Music genre classification aims to group music genres with a high degree of accuracy. Music genre classification is a critical challenge in pattern recognition and digital signal processing. In this paper, we introduce music genres classification using Mel-Frequency Cepstral Coefficient (MFCC) as an extraction feature and using the algorithm Convolutional Neural Network (CNN) as a classification model. The MFCC feature was chosen because of its ability to represent the frequency characteristics of audio signals that correspond to human auditory perception, where the music genre dataset was processed into an MFCC representation before being trained on a CNN model. In this study, we compare three different CNN model to determine the best architecture. The results showed that model architecture 1 obtained the best accuracy during training at 97.15%, while model architecture 2 obtained a training accuracy of 95.74% and model architecture 3 obtained a training accuracy of 95.18%. In testing with new data, model architecture 3 obtained the highest accuracy compared to the other 2 models, with 81%, which indicates good generalization ability. This study proves that the combination of MFCC and CNN is effective for music genre classification with high accuracy.

Keywords:
Music Genre, Classification, MFCC, CNN

1. Introduction

The classification of music genres has become one of the most dynamic fields in digital signal processing and artificial intelligence, as it combines elements of human perception, acoustics, and computational modeling. Music not only plays a vital cultural and emotional role but also influences psychological well-being and cognitive performance, especially among students and younger audiences [1]. However, understanding how to computationally recognize and classify music genres remains a challenge due to the complexity of audio signals, which contain variations in timbre, rhythm, pitch, and dynamics. These characteristics necessitate robust feature extraction methods that can accurately represent the perceptual elements of sound for machine interpretation. Consequently, researchers have increasingly turned to advanced signal processing techniques and deep learning models to improve the accuracy and scalability of genre classification systems.

Feature extraction is a fundamental step in any music classification system. The Mel Frequency Cepstral Coefficients (MFCC) have become a dominant technique due to their ability to mimic the human auditory system's response to sound frequencies [11]. Studies such as those by Sasilo et al. [2] and Ajinurseto et al. [7] demonstrate that MFCCs effectively represent spectral characteristics of audio signals, yielding high recognition accuracy in both speech and music domains. However, while MFCCs successfully extract

Corresponding Author: Mujni Ahmad Ali (mujniahmadali@gmail.com)
1 Bulkis Kanata, University of Mataram, Mataram, Indonesia (uqikanata@te.ftunram.ac.id)
2 Sudi M. Al Sasongko, University of Mataram, Mataram, Indonesia (mariyantosas@unram.ac.id)
3 Mujni Ahmad Ali, University of Mataram, Mataram, Indonesia (mujniahmadali@gmail.com)

perceptual features, they often require complementary algorithms such as the Gaussian Mixture Model (GMM) or Convolutional Neural Networks (CNNs) to classify audio features into distinct categories. A key limitation identified in earlier research is that MFCC alone struggles with high variability in musical styles, which introduces noise and reduces accuracy across large datasets.

In response to these limitations, the use of deep learning—particularly CNNs—has gained traction in music classification research. Elbir and Aydin [3] demonstrated that CNN-based architectures significantly outperform traditional machine learning approaches by automatically learning hierarchical representations from spectrograms, leading to improved classification accuracy. Similarly, Liu et al. [4] proposed a Bottom-up Broadcast Neural Network, emphasizing that hierarchical feature propagation allows better detection of local and global audio patterns. These findings indicate that CNNs are capable of handling the high dimensionality and non-linearity of audio data, making them a promising approach for genre classification tasks. Nonetheless, CNN performance is highly dependent on network architecture design and training efficiency, which are still open challenges in the field.

Comparative studies between deep learning and traditional algorithms reinforce the superiority of CNNs in classifying diverse musical datasets. Lau and Ajoodha [5] showed that CNNs outperform Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) by 15–20% in overall accuracy, primarily due to their automatic feature-learning capabilities. Further, Luis and Rokhman [13] successfully applied CNNs to classify traditional Indonesian music, achieving accuracy rates above 90%—a testament to CNN's adaptability to both modern and traditional music data. However, these models often require substantial computational resources and careful hyperparameter tuning to avoid overfitting, as observed by Soekarta et al. [14]. Hence, there is still a need to balance model complexity with computational efficiency, especially when deploying models on lower-end devices or real-time systems.

In parallel, the advancement of spectrogram-based CNN architectures has enabled significant improvements in classification performance. Purnama [10] employed ResNet-50 and VGG-16 architectures on spectrogram representations, resulting in notable improvements in genre recognition accuracy compared to baseline CNN models. Seo et al. [19] further emphasized the importance of combining multiple spectral features—such as MFCC, chroma, and mel-spectrograms—to capture diverse audio textures and enhance model generalization. These hybrid approaches demonstrate that integrating multiple audio descriptors within CNN pipelines leads to more robust and reliable genre classification systems.

The role of data preprocessing and feature fusion has also become increasingly vital. Mardiani et al. [6] highlighted that effective data preprocessing—such as normalization, silence trimming, and temporal segmentation—can increase classification accuracy by up to 25%. Similarly, Paleva and Prasetio [9] combined Short-Time Fourier Transform (STFT) with MFCC to improve stress-level recognition in speech, demonstrating that hybridized audio features enhance the discriminative power of classification models. Such studies reveal that preprocessing directly influences the model's learning efficiency and output reliability, suggesting that attention to early-stage data handling is as critical as model design.

Recent works have also focused on optimizing CNN training strategies to improve performance without increasing model size. Egele et al. [12] proposed early stopping and hyperparameter optimization techniques that significantly reduced training time by 40% while maintaining accuracy levels. Zhang et al. [18] further advanced this idea by developing lightweight CNN-based systems optimized for low-computing devices, achieving efficient genre classification without compromising recognition quality. These studies align with current trends in making AI models more accessible and sustainable for real-world applications, such as music streaming platforms and mobile applications.

Finally, integrating MFCCs with CNN architectures continues to be one of the most effective approaches in modern audio classification. The combination allows CNNs to process perceptually rich features extracted via MFCC while automatically learning deeper representations from raw audio or spectrogram inputs. Studies by Ayu et al. [8] and Vita Via et al. [20] confirm that MFCC-CNN hybrid systems consistently achieve accuracy rates above 90%, demonstrating strong potential for real-time implementation. However, researchers continue to explore model interpretability, scalability, and adaptability across different music genres and cultural datasets. Therefore, the current study aims to develop an optimized MFCC-CNN framework that enhances both accuracy and computational efficiency in music genre classification tasks.

2. Related Works

Research on music genre classification has evolved rapidly with the advancement of digital signal processing and deep learning, producing a range of methodologies for feature extraction and model optimization. Early studies primarily relied on traditional machine learning algorithms using handcrafted features such as MFCCs, Chroma, and Spectral Centroid to represent audio characteristics. For example, Sasilo et al. [2] applied MFCCs with a Gaussian Mixture Model (GMM) for speech recognition, demonstrating an accuracy of 86%, which validated the robustness of MFCC in capturing sound frequency patterns. Similarly, Ajinurseto et al. [7] implemented MFCC for desktop-based voice recognition, showing that this approach effectively differentiates between various sound inputs. However, these traditional models exhibited performance limitations when applied to larger and more diverse datasets, mainly due to their inability to generalize across musical variations. These findings motivated the adoption of deep learning approaches capable of automatic feature learning from raw or minimally processed audio data.

With the rise of deep learning, Convolutional Neural Networks (CNNs) have become a central focus for music genre classification. Elbir and Aydin [3] pioneered the use of CNNs in this field, combining deep learning with automated feature extraction to improve classification accuracy by over 20% compared to traditional algorithms. Liu et al. [4] later introduced the Bottom-Up Broadcast Neural Network (BUBNN), which enhances feature propagation and fusion across convolutional layers. Their approach effectively captured both temporal and spectral dependencies, improving classification accuracy on benchmark datasets. Similarly, Lau and Ajoodha [5] performed a comparative study between traditional algorithms such as SVM, Random Forest, and deep learning architectures. Their results showed CNNs outperforming all classical models, with a recorded improvement in accuracy from 72% to 92%, emphasizing CNN's strength in hierarchical feature learning and noise resistance.

Hybrid and enhanced CNN architectures have also gained attention for their capacity to further improve model generalization. Purnama [10] explored ResNet-50 and VGG-16 architectures for music genre classification based on spectrogram analysis. By leveraging transfer learning and fine-tuning techniques, Purnama's approach achieved an accuracy of 94.3%, demonstrating the effectiveness of deep residual networks in capturing high-level musical features. Seo et al. [19] conducted a comparative survey across multiple CNN architectures, integrating MFCC, Mel-spectrogram, and Chroma energy features to achieve state-of-the-art results. Their findings highlighted that models trained with diverse feature sets yielded higher precision and recall scores, confirming that multimodal audio representations improve classification reliability.

Another significant development is the use of hybrid feature extraction and data augmentation techniques to enhance CNN performance. Mardiani et al. [6] demonstrated that preprocessing steps such as normalization, silence trimming, and noise reduction can increase classification accuracy by up to 25%. In another study, Paleva and Prasetio [9] combined Short-Time Fourier Transform (STFT) with MFCC to capture both temporal and

spectral information, which improved recognition of stress-level variations in speech. Translating this approach into the domain of music classification, the fusion of STFT and MFCC features allows CNNs to interpret both short-term and long-term frequency patterns more accurately. This hybridization approach not only boosts recognition accuracy but also reduces false positives during genre identification.

Several researchers have extended CNN-based approaches to include regional and cultural music datasets, demonstrating their adaptability beyond Western music genres. Luis and Rokhman [13] applied CNNs to classify traditional Indonesian music genres using spectrogram representations. Their model achieved accuracy levels above 90%, outperforming standard MFCC-based classifiers and proving CNN's ability to learn complex rhythmic and harmonic patterns. Similarly, Soekarta et al. [14] explored hyperparameter tuning in CNNs and found that fine-tuning learning rates, dropout ratios, and kernel sizes improved accuracy by up to 12%. These findings reinforce the importance of architecture optimization in deep learning applications for genre classification, especially when dealing with culturally diverse datasets.

More recent studies have focused on improving computational efficiency and training stability in deep learning models. Egele et al. [12] introduced an early discarding method during neural network hyperparameter optimization, reducing training time by 40% without significantly affecting accuracy. Zhang et al. [18] applied lightweight CNNs for adaptive music recommendation systems, targeting low-computing environments such as mobile devices. Their models achieved comparable classification accuracy (around 90%) with only 50% of the computational cost of standard CNNs. These studies highlight the growing trend of designing efficient, scalable architectures that can perform complex genre classification tasks without extensive computational resources.

Several works have also explored integrating MFCC-based preprocessing with CNNs to improve classification precision. Ayu et al. [8] conducted an extensive analysis using MFCC-CNN hybrids, revealing that combining perceptual frequency features with spatial convolution enables models to achieve accuracies exceeding 92%. Similarly, Vita Via et al. [20] investigated how time-duration segmentation influences CNN performance, concluding that shorter, well-segmented audio samples improve classification consistency and reduce computational time. This reinforces that the combination of MFCC with CNN not only enhances interpretability but also optimizes training efficiency, making it suitable for real-time or embedded applications.

Recent studies have expanded the application of MFCC and CNN beyond genre recognition to tasks such as recommendation systems and adaptive playlist generation. T. Zhang et al. [18] developed a low-computation adaptive music recommendation framework using CNNs trained on MFCC features, which dynamically adjusted suggestions based on user listening patterns. This approach demonstrated CNN's versatility in handling not only classification tasks but also broader personalization functions in digital music platforms. Furthermore, W. Seo et al. [19] provided an in-depth comparative analysis of multiple spectral feature combinations within CNN frameworks, confirming that the fusion of Mel-spectrograms, chroma, and MFCC achieved the highest F1-scores across various datasets. These findings collectively underscore the critical role of CNNs combined with MFCC in delivering high-accuracy, scalable, and efficient music genre classification solutions.

Overall, the reviewed literature demonstrates that integrating MFCC with CNN architectures offers a robust and efficient framework for music genre classification. While traditional algorithms remain useful for simpler tasks, CNN-based models have proven superior in capturing non-linear audio relationships and high-level feature abstractions. Therefore, the present study focuses on developing an optimized MFCC-CNN hybrid model that enhances both accuracy and computational performance, contributing to the evolving field of intelligent audio analysis and music information retrieval.

3. Proposed Method

In this study, we propose a method for music genre classification that combines Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction and a Convolutional Neural Network (CNN) for classification. The proposed approach is designed to accurately classify short-duration music clips into ten distinct genres, sourced from two datasets: the GTZAN dataset from Kaggle and selected music clips from YouTube channels.

The workflow of the proposed method begins with data collection and preprocessing. A total of 1,000 music clips were gathered, with 100 clips representing each genre. The GTZAN dataset provides the genres blues, classical, country, disco, and hip-hop, while the YouTube dataset contributes jazz, metal, pop, reggae, and rock. All audio files were converted to .wav format and trimmed to 10-second durations to ensure uniform input for the CNN models. Subsequently, the dataset was divided into training (80%) and testing (20%) subsets to facilitate model training and evaluation.

In this study, we conduct feature extraction using MFCC to transform raw audio signals into meaningful acoustic features. This process involves pre-emphasis, frame blocking, windowing, Fast Fourier Transform (FFT), mel-frequency wrapping, and cepstral coefficient calculation. The resulting MFCC features capture essential frequency and harmonic characteristics of the music, serving as input to the CNN models for further processing.

We tested three sequential CNN architectures and compared them to determine the optimal configuration. Each model consists of convolutional layers with varied filter sizes, pooling layers to reduce dimensionality, and dropout and batch normalization layers to prevent overfitting. A flatten layer converts 2D feature maps into 1D vectors, which are then passed to fully connected layers for final classification into the ten music genres. Differences among the architectures, such as the number of convolutional layers, filter sizes, and dropout rates, allow for the evaluation of performance across multiple configurations. In this study, we construct a mathematical formulation of the key components MFCC and CNN as follows:

1. MFCC Feature Extraction

MFCC converts an audio signal $x(t)$ into a sequence of feature vectors that represent perceptual frequency content:

$$\text{MFCC}(n) = \sum_{k=1}^{K} log\big(S(k)\big) \cdot cos\left[n\Big(k - \frac{1}{2}\Big)\frac{\pi}{K}\right] \quad (1)$$

where:

- $S(k)$ = Mel-scaled power spectrum obtained from the Short-Time Fourier Transform (STFT),
- K = number of Mel filter banks,
- n = cepstral coefficient index (typically ($n = 1,2,\ldots,13$)).

The Mel frequency scale maps real frequency f (Hz) to perceptual frequency m (Mels):

$$m = 2595log_{10}\left(1+\frac{f}{700}\right)$$

This transformation emphasizes frequency ranges important to human hearing, producing features that are robust for genre classification.

2. CNN-Based Classification

After extracting MFCC features, the CNN performs hierarchical feature learning and classification through the following core operations:

Convolution Layer:

$$z_{i,j}^{(l)} = f\left(\sum_{m,n} x_{i+m,j+n}^{(l-1)} \cdot w_{m,n}^{(l)} + b^{(l)}\right) \quad (2)$$

where:

- $x^{(l-1)}$ = input feature map (MFCC matrix),
- $w^{(l)}$ = convolution kernel (filter),
- $b^{(l)}$ = bias term,
- $f(\cdot)$ = activation function, e.g., ReLU $f(x) = \max(0, x)$

Pooling Layer (Downsampling):

$$p_{i,j}^{(l)} = \max_{(m,n)\in R} z_{i+m,j+n}^{(l)}$$

which reduces spatial dimensions while retaining dominant features.

Fully Connected Layer and Softmax Classification:

$$\hat{y}_c = \frac{e^{z_c}}{\sum_{k=1}^{C} e^{z_k}} \quad (3)$$

where $\hat{y}_c$ is the probability that the input belongs to class c (music genre), and C is the total number of genres.

In this model, MFCC extracts perceptually meaningful frequency-domain features from raw audio, effectively reducing noise and dimensionality. The CNN then learns spatial correlations between these MFCC features through convolution and pooling, enabling hierarchical representation learning. Finally, the Softmax layer classifies the audio sample into a predefined genre. This integration allows for efficient and accurate music genre classification by combining signal-processing precision (MFCC) with deep-learning adaptability (CNN).

4. Experimental Setup

1. Data Collection

The research data was taken from Kaggle, namely the GTZAN dataset, and also took the dataset via YouTube channels NPR Music, Majestic Casual, Nuclear Blast Records, and Green Day. The total dataset used in this study was 1000 datasets, with each genre having 100 data points. The dataset sources in this study were 5 genres from the GTZAN dataset, which consisted of the following genres: blues, classical, country, disco, hiphop, and 5 genres from YouTube consist of genres jazz, metal, pop, reggae, and rock. This dataset in .wav format will be the main input for the classification process.

2. Pre-processing of Data Processing

In this stage, the file format is converted to the '.wav' extension using Audacity. The data is then trimmed to an even 10-second duration. This is because during training, the durations must be the same, according to the existing CNN architecture. The data is then divided into training and test data with a ratio of 8:2, with 800 training data points and 200 test data points.

3. Data Extraction with MFCC Feature

In this stage, we adopt MFCC as a method that represents audio as coefficients based on the sound frequencies [16]. MFCC works by converting speech signals into a frequency spectrum through a feature extraction process [17] [18]. In its implementation, MFCC is often combined with other algorithms, such as Short Time Fourier Transform (STFT), to produce a time-repeated representation of sound frequencies, providing more detailed information about the variations in the sound signal [19]. An illustration of the stages of MFCC can be seen in Figure 2.

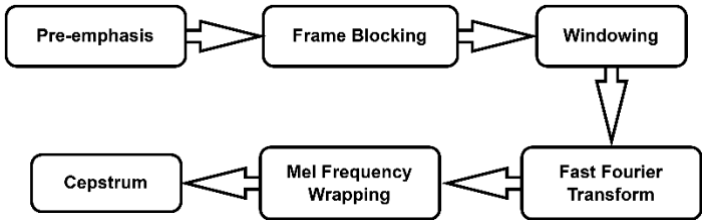
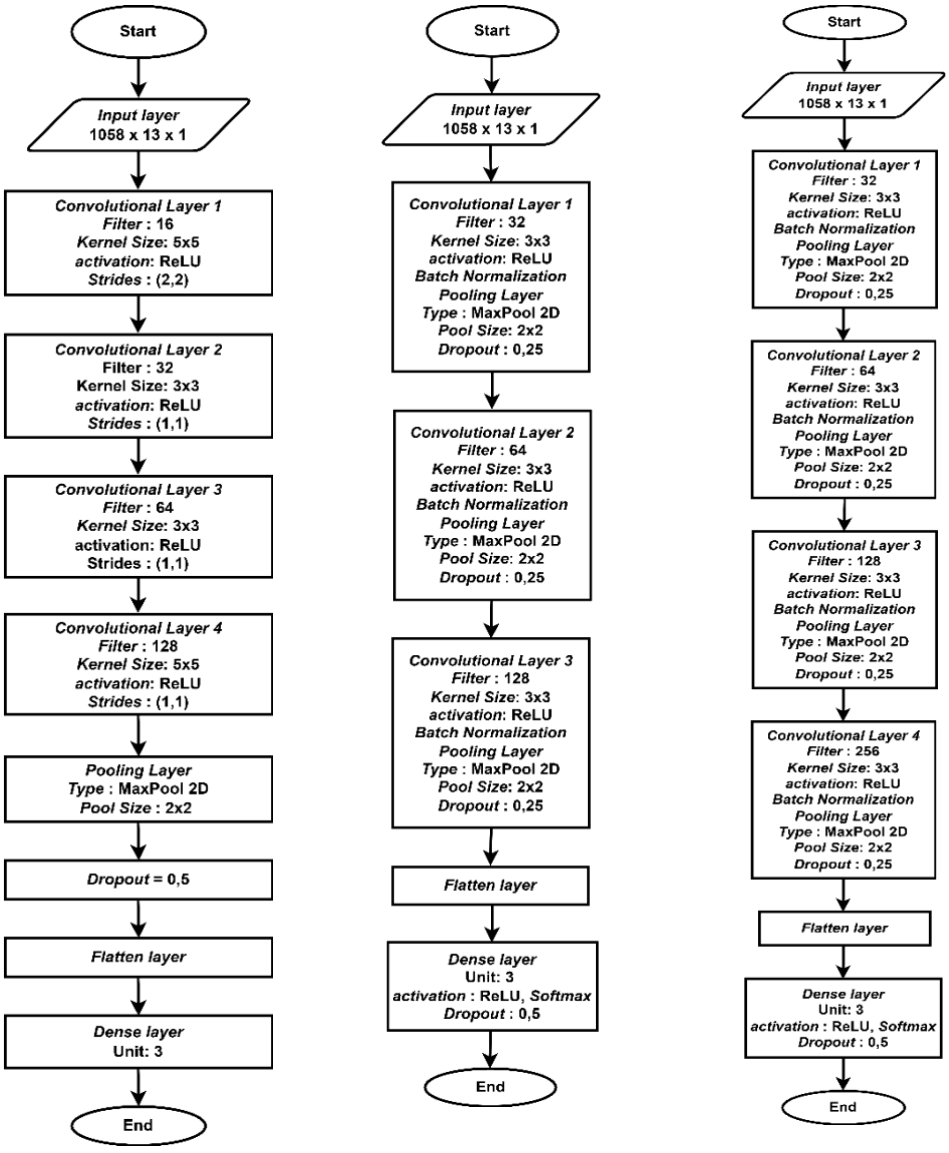


Fig. 2. Stage Illustration Mel-Frequency Cepstral Coefficient

4. CNN Model Architecture

In this study, we tested three different CNN architectures to measure classification results. Fig. 3 shows the architecture of the three CNN models.



Model 1 Model 2 Model 3

Fig. 3 Different models of CNN architecture

The three architectures in Figures 3 to 5 are models used in research by varying hyperparameters that exist to obtain the best accuracy during training and testing. The CNN model architecture consists of feature learning and classification. Feature learning is composed of several convolutional layers, a pooling layer, a dropout layer, and batch normalization, where, in the architecture used, the number of filters used in the convolution layer is varied, as well as the addition of dropout on each layer. While classification consists of a flattened layer and a fully connected which is at the stage of flattening, the result of feature learning will be made into 1D, which will then be connected to all the layers that exist in the fully connected layer to be classified into several desired classes.

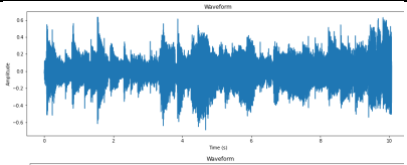
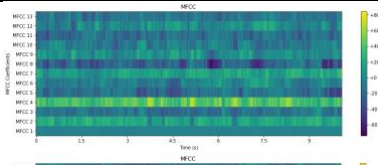
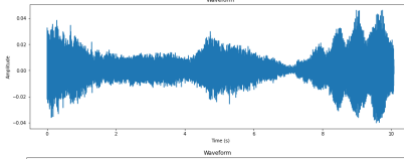
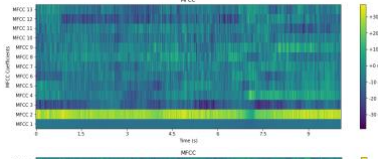
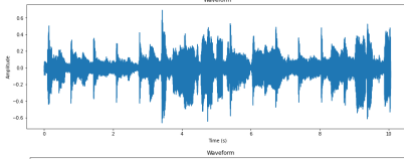
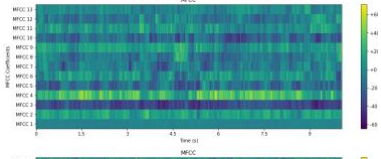
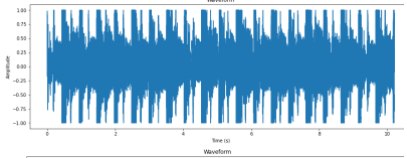
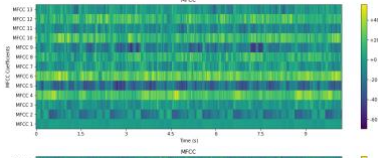
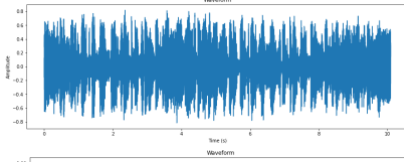
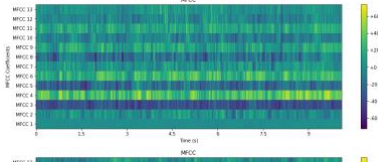
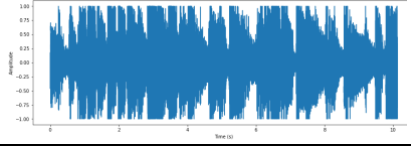
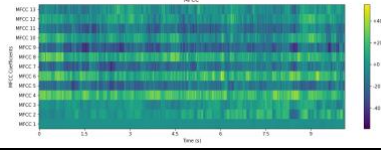
5. Evaluation Model

Training a CNN model involves important parameters such as the number of epochs and computation time [20]. While more epochs increase computation time, they can improve model accuracy, depending on the data and performance hardware. After training, the model is tested with untrained test data, and the results are evaluated using a confusion matrix, which includes metrics such as accuracy, precision, recall, and f1-score.

5. Result and Analysis

The results of this data collection include 10 music genres, with 100 data points per genre, and a total of 1,000 data points. Each data point has the same duration, 10 seconds. The entire data set is divided into training and test data, with 80% weighting for training and 20% for testing. MFCC feature extraction is a crucial step in audio signal processing, especially for speech classification and recognition purposes. MFCC features capture important characteristics of an audio signal, enabling machine learning models to effectively distinguish between different voice types. Table 1 shows the MFCC results for each genre.

Table 1. Plotting MFCC Type Genre

Genre	Plot Signal	MFCCs
blues		
classic		
country		
disco		
hiphop		
jazz		

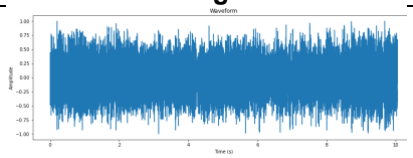
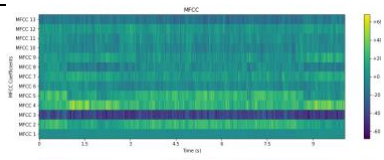
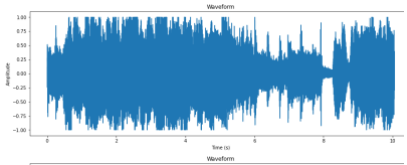
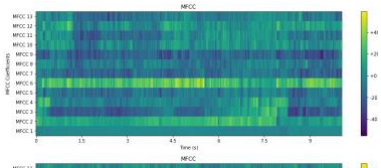
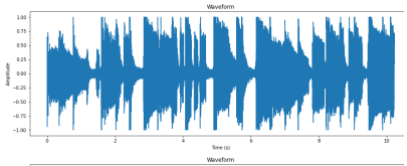
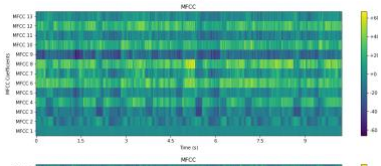
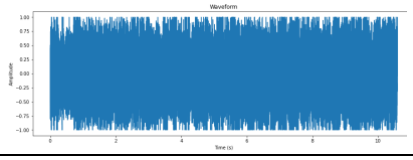
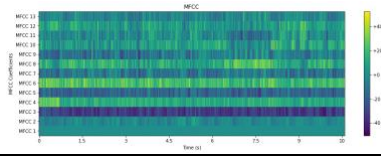
Genre	Plot Signal	MFCCs
metal		
pop		
reggae		
rock		

Table 1 shows the results of the MFCC extraction, with the vertical axis representing the number of MFCC coefficients per segment and the horizontal axis representing the number of frames or time steps. The coefficients and frames have positive and negative array values. Positive values indicate the coefficients represent sonorant sounds, as their spectral energy is concentrated at low frequencies, while negative values indicate fricative sounds, where their spectral energy is concentrated at high frequencies.

This stage involves training using training data from the previously created architecture. The pre-training stage involves extracting MFCC features. The output generated from the MFCC extraction will be used as input to the CNN. The following Tables 2, 3, and 4 present the results of the three model architectures.

Table 2. Output Model 1 Architecture

Layer (type)	Output Shape	Parameter
conv2d (Conv2D)	(None, 5, 7, 16)	416
conv2d_1 (Conv2D)	(None, 5, 7, 32)	4,640
conv2d_2 (Conv2D)	(None, 5, 7, 64)	18,496
conv2d_3 (Conv2D)	(None, 5, 7, 128)	204,928
max_pooling2d (MaxPooling2D)	(None, 2, 3, 128)	0
dropout (Dropout)	(None, 2, 3, 128)	0
flatten (Flatten)	(None, 768)	0
dense (Dense)	(None, 128)	98,432
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 10)	650
Total params: 335,818 (1.28 MB)		
Trainable params: 335,818 (1.28 MB)		
Non-trainable params: 0 (0.00 B)		

Table 3. Output Model 2 Architecture

Layer (type)	Output Shape	Parameter
conv2d (Conv2D)	(None, 9, 13, 32)	320
batch_normalization (BatchNormalization)	(None, 9, 13, 32)	128
max_pooling2d (MaxPooling2D)	(None, 5, 7, 32)	0
dropout (Dropout)	(None, 5, 7, 32)	0
conv2d_1 (Conv2D)	(None, 5, 7, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 5, 7, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 3, 4, 64)	0
dropout_1 (Dropout)	(None, 3, 4, 64)	0
conv2d_2 (Conv2D)	(None, 3, 4, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 3, 4, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 128)	0
dropout_2 (Dropout)	(None, 2, 2, 128)	0
Flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131,328
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dense_3 (Dense)	(None, 10)	650
dense_4 (Dense)	(None, 10)	650
Total params: 266,698 (1.02 MB)		
Trainable params: 266,250 (1.02 MB)		
Non-trainable params: 448 (1.75 KB)		

Table 4. Output Model 3 Architecture

Layer (type)	Output Shape	Parameter
conv2d (Conv2D)	(None, 9, 13, 32)	320
batch_normalization (BatchNormalization)	(None, 9, 13, 32)	128
max_pooling2d (MaxPooling2D)	(None, 5, 7, 32)	0
dropout (Dropout)	(None, 5, 7, 32)	0
conv2d_1 (Conv2D)	(None, 5, 7, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 5, 7, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 3, 4, 64)	0
dropout_1 (Dropout)	(None, 3, 4, 64)	0
conv2d_2 (Conv2D)	(None, 3, 4, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 3, 4, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 128)	0
dropout_2 (Dropout)	(None, 2, 2, 128)	0
conv2d_3 (Conv2D)	(None, 2, 2, 256)	295,168
batch_normalization_3 (BatchNormalization)	(None, 2, 2, 256)	1024

Layer (type)	Output Shape	Parameter
max_pooling2d_3 (MaxPooling2D)	(None, 1, 1, 128)	0
dropout_3 (Dropout)	(None, 1, 1, 128)	0
Flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 256)	65,792
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_5 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_6 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
dense_4 (Dense)	(None, 10)	330
Total params: 499,114 (1.90 MB) Trainable params: 498,154 (1.90 MB) Non-trainable params: 960 (3.75 KB)		

In the training process, the model is trained using 100 epochs with a batch size of 64. The results obtained were that model 1 obtained the best accuracy epoch of 98th, which is 0.9715 or 97.15%. In model 2, the best accuracy was obtained at epoch 81st is 0.9574 or 95.74%. Then in model 3, the best accuracy is obtained at epoch 94th is 0.9518 or 95.18%. The accuracy value indicates the model's correct predictions compared to the total predicted values. A good accuracy value, approaching 1.0 or 100%, is considered excellent, indicating the model consistently makes correct predictions. The accuracy values generated from the three CNN architectures are visualized in Figs. 4, 5, and 6 as follows.

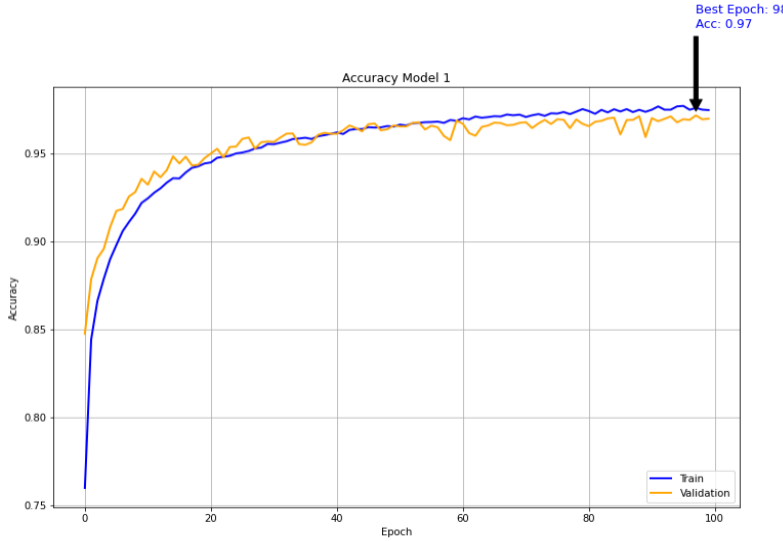


Fig. 4. Training and Validation Accuracy Model 1

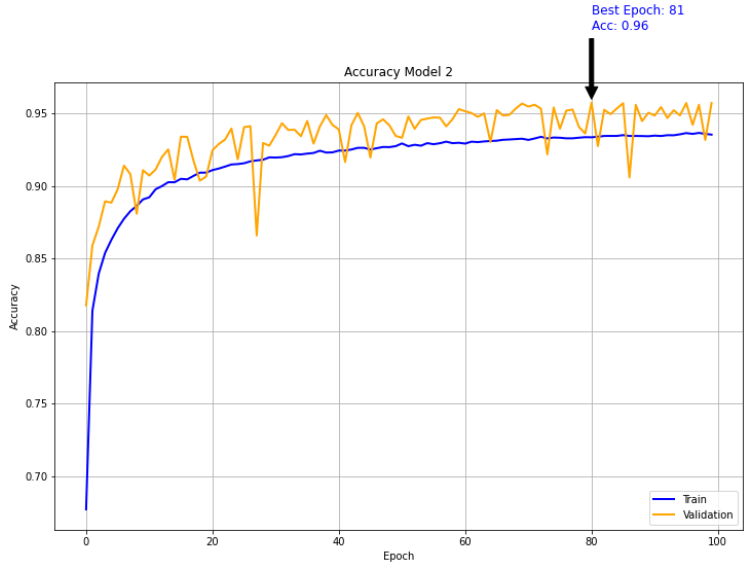


Fig. 5. Training and Validation Accuracy Model 2

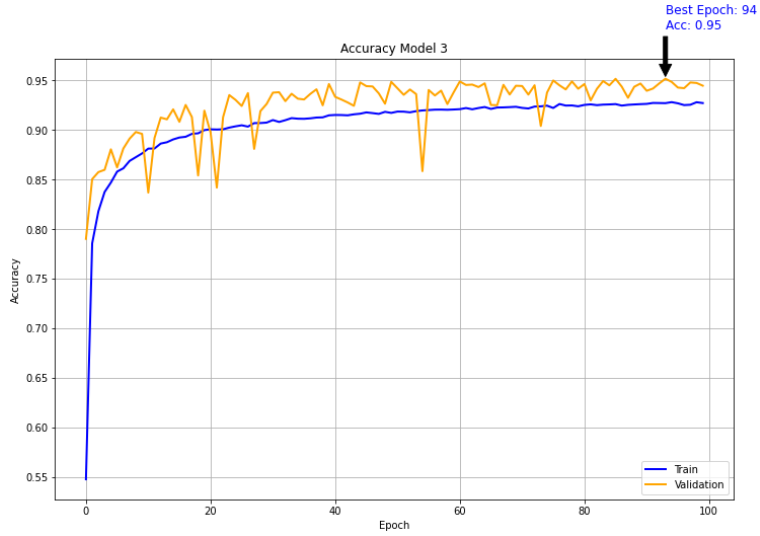


Fig. 6. Training and Validation Accuracy Model 3

In the testing and evaluation stage, we tested the model using 200 test data sets, with each class consisting of 20 audio data sets. The test results will be saved in CSV format. Table 5 presents the test results for Model 3.

Table 5. Sample Test Results on Model 3

fname	label	blues	classic	country	disco	hiphop	jazz	metal	pop	reggae	rock	y_pred
3511d08e.wav	blues	0.2183	0.0894	0.0071	0.0002	0.5640	0.0011	0.1179	0.0006	0.0001	0.0004	hiphop
cbc5a1c0.wav	blues	0.5479	0.1591	0.0089	0.0002	0.2575	0.0011	0.0233	0.0006	0.0006	0.0003	blues
989a85d8.wav	classic	0.1598	0.8135	0.0028	9,21E+10	0.0187	0.0009	0.0025	0.0009	0.0003	0.0001	classic
02655cf5.wav	classic	0.1434	0.6508	0.0550	0.0010	0.0827	0.0103	0.0225	0.0039	0.0057	0.0241	classic
69aca3cb.wav	country	0.0018	0.0007	0.7100	0.0001	0.2595	0.0002	0.0156	0.0005	0.0013	0.0098	country
6ab3333e.wav	country	0.0064	0.0009	0.8337	7.72E+10	0.1191	0.0002	0.0385	0.0001	0.0005	0.0001	country
64f79ff5.wav	disco	1.39E+05	1.10E+05	2,15E+04	0.9997	8.36E+04	3,17E+10	6.40E+04	1.06E+09	0.0001	2.75E+11	disco

fname	label	blues	classic	country	disco	hiphop	jazz	metal	pop	reggae	rock	y_pred
c332c367.wav	disco	9,60E+08	4.89E+10	6.83E+09	0.9688	9.24E+09	0.0020	1,13E+10	0.0001	0.0174	0.0114	disco
4ed8f9fb.wav	hiphop	0.0693	0.0962	0.0038	0.0001	0.4359	0.0007	0.3927	0.0004	0.0003	0.0003	hiphop
a52da687.wav	hiphop	0.0380	0.0022	0.0069	4.34E+10	0.7558	5.29E+10	0.1964	4,02E+10	0.0001	0.0001	hiphop
bb41627e.wav	jazz	0.0194	0.0047	0.01918	0.0029	0.0137	0.2117	0.0021	0.3990	0.2383	0.0885	pop
5edac94b.wav	jazz	0.0062	0.0025	0.0019	0.0042	0.0014	0.3794	0.0007	0.3448	0.1334	0.1251	jazz
100e989b.wav	metal	0.0018	0.0007	0.4268	9,20E+10	0.1491	7.01E+10	0.4205	6.95E+09	0.0002	0.0003	country
322885e1.wav	metal	0.0037	0.0011	0.0076	2.61E+11	0.2447	1.07E+11	0.7424	1.85E+10	3.29E+11	0.0001	metal
6e6b91ef.wav	pop	0.0016	0.0009	0.0123	0.0039	0.0067	0.0886	0.0015	0.3333	0.1951	0.3556	rock
2aef129e.wav	pop	0.0010	0.0009	0.0124	0.0054	0.0115	0.0935	0.0014	0.3772	0.1880	0.3083	pop
d6179156.wav	reggae	0.0006	0.0005	0.0237	0.2909	0.0069	0.3100	0.0006	0.0192	0.2984	0.0487	jazz
a82b1989.wav	reggae	0.0004	0.0002	0.0011	0.0695	0.0006	0.1418	0.0002	0.0355	0.7119	0.0385	reggae
c2fe82a3.wav	rock	6,98E+10	6,66E+10	0.0081	0.0023	0.0015	0.0015	0.0002	0.0020	0.0298	0.9540	rock
745b7fa9.wav	rock	1,21E+10	5.49E+09	1,45E+11	3.78E+11	4,88E+08	6,53E+10	5,58E+09	0.0326	0.0010	0.9662	rock

100%| | 200/200 [34:28<00:00, 10.34s/it]Accuracy Score: 0.805

Table 2 shows the results of testing model 3. The predicted result was 0.805, or 80.5%, or rounded to 81%, which represents a correct prediction accuracy rate. After testing the model, the model was evaluated. Table 6 presents the classification report to provide information about evaluation metrics such as accuracy, precision, recall, and F1-score.

Table 6. Classification Report Model 3

Genre	precision	recall	f1-Score	Support
blues	0,81	0,65	0,72	20
classic	0,95	1,00	0,98	20
country	0,95	1,00	0,98	20
disco	0,91	1,00	0,95	20
hiphop	0,74	0,85	0,79	20
jazz	0,81	0,65	0,72	20
metal	1,00	0,55	0,71	20
pop	0,76	0,65	0,70	20
reggae	0,74	0,70	0,72	20
rock	0,59	1,00	0,74	20
accuracy	0,81			200
macro avg	0,83	0,81	0,80	200
weighted avg	0,83	0,81	0,80	200

Table 6 shows the model has an accuracy of 81% for the F1-score. The evaluation metric values are obtained from calculations involving parameters True Positive, False Positive, True Negative, dan False Negative. Plot from the confusion matrix can be seen in Fig. 7 as follows:

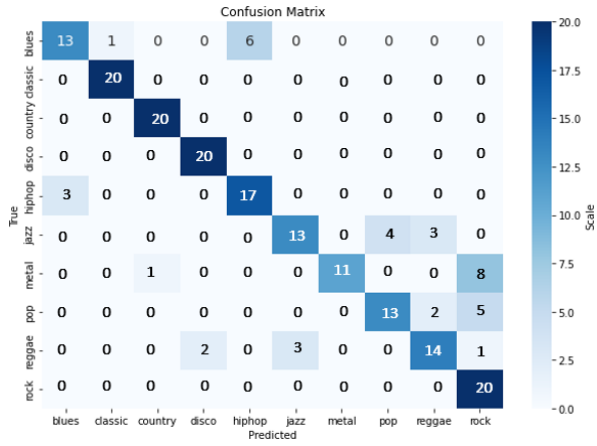


Fig. 7. Confusion Matrix Model 3 Results

Table 7. Testing and Evaluation Results of the Three Models Used.

Model	Accuracy	Precision	Recall	F-1 Score
Model 1	0,80	0,81	0,80	0,79
Model 2	0,80	0,82	0,79	0,79
Model 3	0,81	0,83	0,81	0,80

According to Table 7, model 1 shows an accuracy of 80%, a precision of 0.81, a recall of 0.80, and an F-1 Score of 0.79. Model 1 shows an accuracy of 80%, meaning it can correctly classify 80% of the test data. Precision 0.81 indicates that this model is quite good at predicting the correct class (not too many false positives). With a recall of 0.80, the model is able to capture 80% of the total correct examples. F1-Score 0.79 indicates a balance between precision and recall, but is slightly lower, indicating that the model may still miss some predictions or make errors in some predictions.

Model 2 also has 80% accuracy, the same as Model 1, but higher precision at 0.82. This indicates that this model is slightly better at avoiding false positives (wrong predictions for a particular class). However, recall is lower at 0.79, indicating that this model does not capture all the data that should be classified correctly. F1-Score 0.79 indicates a balance between precision and recall, which is almost the same as Model 1, but this model tends to focus more on reducing prediction errors. (false positives).

Model 3 performed best of the three models with 81% accuracy, indicating a better ability to correctly classify the data. Precision 0.83 means this model is the best at predicting the correct class, with a slight false positive compared to Model 1 and Model 2. With a recall of 0.81, this model is also better at capturing data examples that should be classified correctly than the other two models. F1-Score 0.80 indicates that this model has the best balance between precision and recall, making it a superior choice for handling consistently accurate classification.

6. Conclusion

This study test three model architectures to classify music genres which showed that each model produced varying levels of accuracy. Model 1 demonstrated the highest accuracy compared to the other two models. Model 1 uses a simple model architecture that enables it to recognize data patterns more quickly and effectively, resulting in the highest accuracy of 97.15%, despite requiring a long computation time. Meanwhile, Models 2 and 3 apply regularization techniques in the convolutional layer. The use of higher dropout aims

to reduce overfitting, so the model can perform better when tested with new data. However, with the use of a larger dropout, the training accuracy produced by these two models decreased slightly compared to Model 1. Model 2 produced an accuracy of 95.74%, and Model 3 95.18%. In testing, Model 3 produced the best accuracy of 81%, higher than Models 2 and 3, which only produced an accuracy of 80%. This shows that Model 3 has good generalization ability with better performance due to its more complex architecture. This generalization ability is very important because it shows that the model not only memorizes the patterns of the training data, but is also able to recognize patterns in new data. This study proves that the combination of MFCC and CNN is effective for music genre classification.

References

[1] "The Dynamics of Music's Influence on Students' Psychological Well-Being: A Literature Analysis of Neurological and Emotional Responses."

[2] A. A. Sasilo, R. A. Saputra, and I. P. Ningrum, "Speech Recognition System Using Mel Frequency Cepstral Coefficients and Gaussian Mixture Model Method," Komputika: J. Comput. Syst., vol. 11, no. 2, pp. 203–210, Aug. 2022, doi: 10.34010/komputyika.v11i2.6655.

[3] A. Elbir and N. Aydin, "Music Genre Classification and Music Recommendation by Using Deep Learning," Electron. Lett., vol. 56, no. 12, pp. 627–629, Jun. 2020, doi: 10.1049/el.2019.4202.

[4] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up Broadcast Neural Network For Music Genre Classification," Jan. 2019. [Online]. Available: http://arxiv.org/abs/1901.08928

[5] D. S. Lau and R. Ajoodha, "Music Genre Classification: A Comparative Study Between Deep Learning and Traditional Machine Learning Approaches," in Lecture Notes in Networks and Systems, Springer Science and Business Media Deutschland GmbH, 2022, pp. 239–247, doi: 10.1007/978-981-16-2102-4_22.

[6] E. Mardiani et al., "Application of Supervised Learning Algorithm for Music Listening Data Classification," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 3, no. 2, pp. 115–124, Sep. 2023, doi: 10.57152/malcom.v3i2.879.

[7] G. Ajinurseto, L. O. Bakrim, and N. Islamuddin, "Application of Mel Frequency Cepstral Coefficients Method in Desktop-Based Speech Recognition System," Infomatek, vol. 25, no. 1, pp. 11–20, Jun. 2023, doi: 10.23969/infomatek.v25i1.6109.

[8] G. Ayu, V. M. Giri, and L. Radhitya, "Classification of Music Genres Using Machine Learning Techniques."

[9] H. R. Paleva and B. H. Prasetio, "Application of Short Time Fourier Transform on MFCC for Stress Level Speech Recognition System," 2024. [Online]. Available: http://j-ptiik.ub.ac.id

[10] N. Purnama, "Music Genre Recommendations Based on Spectrogram Analysis Using Convolutional Neural Network Algorithm with RESNET-50 and VGG-16 Architecture," JISA, 2022.

[11] P. Thu and Z. Tun, "Audio Feature Extraction Using Mel-Frequency Cepstral Coefficients," IJCIRAS, vol. 2, p. 12, 2020. [Online]. Available: https://zenodo.org/badge/61622039.svg

[12] R. Egele, F. Mohr, T. Viering, and P. Balaprakash, "The Unreasonable Effectiveness of Early Discarding After One Epoch in Neural Network Hyperparameter Optimization," Neurocomputing, vol. 597, Sep. 2024, doi: 10.1016/j.neucom.2024.127964.

[13] R. Luis and N. Rokhman, "Traditional Music Regional Classification Using Convolutional Neural Network (CNN)," IJCCS, vol. 16, no. 4, p. 379, Oct. 2022, doi: 10.22146/ijccs.73910.

[14] R. Soekarta, S. Aras, and A. N. Aswad, "Hyperparameter Optimization of CNN Classifier for Music Genre Classification," RESTI J. Syst. Eng. Inf. Technol., vol. 7, no. 5, pp. 1205–1210, Oct. 2023, doi: 10.29207/resti.v7i5.5319.

[15] S. Muraru and C. L. Cocianu, "Spoken Digit Recognition Using the k-Nearest-Neighbor Method and Mel Frequency Cepstral Coefficients," Economic Informatics, vol. 28, no. 2/2024, pp. 5–16, Jun. 2024, doi: 10.24818/issn14531305/28.2.2024.01.

[16] S. N. Luqman et al., "Comparison of Music Genre Classification Algorithms on Spotify Using CRISP-DM," 2021.

[17] T. Mayuna and A. Witanti, “Identification of Music Genres on the Spotify Platform Using the K-Nearest Neighbor Method,” 2023.
[18] T. Zhang, X. Liu, Z. Guo, and Y. Tian, “Adaptive Music Recommendation: Applying Machine Learning Algorithms Using Low Computing Device,” Journal of Software Engineering and Applications, vol. 17, no. 11, pp. 817–831, 2024, doi: 10.4236/jsea.2024.1711045.
[19] 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.
[20] Y. Vita Via, I. Yuniar Purbasari, and A. Putra Pratama, “Analysis of Convolution Neural Network (CNN) Algorithm on Music Genre Classification Based on Time Duration.”