Scientific Journal of Informatics
Vol. 11, No. 4, Nov 2024

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Music Genre Classification Using Mel Frequency Cepstral Coefficients and Artificial Neural Networks: A Novel Approach

Alamsyah[1*], Fahmi Ardiansyah[2], Abdul Kholiq[3]
[1,2]Department of Computer Science, Universitas Negeri Semarang, Indonesia
[3]Department of Guidance and Counseling, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: Music is an artistic expression with many categories in various genres and styles, characterized by its melodic and harmonic compositions. Music genre classification is crucial because genres serve as descriptors commonly used to organize large music collections, especially on the internet and in widely used applications like JOOX and Spotify. The aim of this research is to implement the Mel Frequency Cepstral Coefficients (MFCC) feature extraction method to generate numerical features from a set of specific music tracks. This collection of information will then be classified using machine learning.
Methods: The method used in this study begins with combining the "GTZAN Dataset - Music Genre Classification" with additional data from TikTok and YouTube. The total dataset consists of 1,200 audio files, divided into 12 classes. The MFCC extraction process generates numerical representations of acoustic characteristics, which are then processed using Artificial Neural Networks.
Result: The experiments demonstrate that increasing the amount of data is crucial, as it can enhance both variation and accuracy. The average accuracy achieved in this study is 91.42%, while the highest accuracy reaches 92.16%. These findings indicate that this study outperforms previous studies.
Novelty: The novelty of this research lies in the integration of dynamic social media data (TikTok and YouTube) to enrich the standard GTZAN dataset, the repetition of the MFCC feature extraction process, and the combination of MFCC with Artificial Neural Networks (ANN).

Keywords: Music genre classification, Mel frequency cepstral coefficients, Artificial neural networks
Received September 2024 / **Revised** November 2024 / **Accepted** November 2024

INTRODUCTION

Music, as a form of expression of human emotions and feelings, plays a significant role in various contexts, such as music recommendation systems, automatic music generation, music therapy, and music visualization. According to Yang et al. (2022) [1], music can be viewed as an extension of the human thought process. Choosing the right music genre can increase calmness and efficiency in daily human activities. Recognizing music genres requires a careful approach in feature extraction, given the complexity of audio signals. Several studies have utilized the Mel Frequency Cepstral Coefficients (MFCC) method combined with classification models such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN). The MFCC method has proven effective and can achieve a recognition rate of more than 85% [2]–[4].

In a related study, Bawitlung and Dash [5] used Convolutional Neural Networks (CNN) for music genre classification based on the GTZAN audio dataset. Their study achieved 85% accuracy through careful CNN model design and feature extraction using MFCC. Another study by Cai and Zhang [6] proposed a new classification framework that integrates auditory image features with traditional acoustic and spectral features. This approach resulted in improved accuracy, reaching 91.8% on several datasets, including GTZAN and ISMIR2004.

The following researchers, Mu [7] applied the K-Nearest Neighbors (KNN) algorithm, Miri et al. [8] explored various deep learning models such as CNN on MFCC features, Xie et al. [9] used Residual Gated CNN and Transformer models, and Liu et al. [10] introduced Locally Activated Gated Neural Networks (Lgnet) for music genre classification. Finally, Kakarla and colleagues [11] used Recurrent Neural

[*] Corresponding author.
Email addresses: alamsyah@mail.unnes.ac.id (Alamsyah)
DOI: 10.15294/sji.v11i4.13660

Networks (RNN), specifically exploring Long Short-Term Memory (LSTM) and Independent RNN (IndRNN) for music genre classification, achieving an accuracy of 82.71%.

Although many previous studies have used MFCC [12]–[14] in combination with CNN and RNN for music genre classification, challenges such as overfitting and dataset limitations are still common. Many previous studies are limited to the GTZAN dataset or pre-defined datasets with limited genres, as shown by Miri et al. [8] and Liu et al. [10]. Furthermore, exploring the integration of dynamic social media data, such as from TikTok and YouTube, is needed to increase data variation and model complexity. Therefore, this study aims to fill this gap by integrating data from more diverse and broader sources, which is expected to improve model accuracy and generalization. This study offers a novel contribution to improving music genre classification by utilizing optimized MFCC [15]–[17] and ANN techniques through the inclusion of additional social media data. With the rapid growth of digital technology and the multimedia industry today, it is hoped that the findings of this study will help in achieving the desired music genre classification for users.

METHODS

This section describes the dataset used, the proposed method, and the evaluation process.

Dataset

This study utilizes the GTZAN Dataset - Music Genre Classification obtained from Kaggle. The dataset consists of 10 subjects or classes (10 genres/types), where each class contains 1,000 music tracks (1,000 audio files). Additionally, the dataset was expanded with 2 more classes by adding 200 audio files downloaded from TikTok and YouTube. This brings the total data to 1,200 music tracks. In this study, two experiments were conducted. The first experiment involved extracting MFCC features once for each song, while in the second experiment, MFCC extraction was repeated 10 times for each song. The second experiment resulted in a total of 720,000 pieces of music information, obtained from 12,000 rows × 60 columns. All this music data was then divided into testing and training datasets, with 80% used for training and 20% for testing. Table 1 provides a detailed breakdown of the dataset distribution.

Table 1. Dataset distribution

Type	Amount
Audio files	1,200
MFCC Coefficients	20
MFCC Variance	240,000
MFCC Mean	240,000
Filename	12,000
Length (66149)	12,000
Chroma STFT	24,000
RMS	24,000
Spectral Centroid	24,000
Spectral Bandwidth	24,000
Roll Off	24,000
Zero Crossing Rate	24,000
Harmony	24,000
Perceptual	24,000
Tempo	12,000
Labels	12,000
Total Features	720,000
Data Type	String, Float

Pre-processing

Before extracting features using MFCC, the music files captured in .wav, .au, or .mp3 formats undergo a pre-processing stage. This involves reading the audio data using libraries such as librosa.display and IPython.display.

In this pre-processing step, two key pieces of information are extracted: the sample data from the audio file and its sample rate. For this study, only the first second of each audio file is analyzed. The amount of data extracted from each file is determined by multiplying the sample rate by the duration (in seconds). In this case, feature extraction is set at 720,000 for 12,000 rows, meaning that each audio file will contain 60 feature columns.

However, since the wavelengths are normalized and truncated to the same length, certain columns containing wavelength-related information (e.g., Length, Filename, and Label genre) are unnecessary and are thus removed from the dataset. As a result, the processed dataset includes 57 columns for each of the 12,000 audio samples, totaling 68,400 feature values that represent the characteristics of each song, as shown in Table 2.

Table 2. MFCC vector values

Frame to - n	Signal value (y)
0	-0.28111
1	-0.39162
2	-0.45754
…	…
57	-0.33805

Mel frequency cepstral coefficients (MFCC) feature extraction

MFCC are based on the Mel frequency scale, which is designed to emulate the human auditory system. This approach helps compensate for channel distortions and has become one of the most widely used techniques for feature extraction in speech, audio, and video recognition. MFCC works by modeling the distribution of spectral energy coefficients and uses critical band frequency resolution along with logarithmic power waves, which are key parameters in analyzing acoustic signals. The Mel frequency scale itself is constructed using paired sinusoidal waves.

The MFCC extraction process involves several key steps:

1. Pre-processing: The initial step involves preparing the signal before feature extraction begins. This includes normalizing the audio signal.
2. Framing: The audio signal is divided into small overlapping frames that maintain the periodicity of the information, helping to preserve important details in each segment.
3. Windowing: Each frame is multiplied by a window function (such as the Hamming window) to reduce discontinuities at the beginning and end of each frame, which could lead to distortions.
4. Discrete Fourier Transform (DFT): The time-domain audio frames are transformed into the spectral frequency domain using DFT. The resulting spectral magnitudes are then filtered through a filter bank to focus on the most relevant frequency components.
5. Logarithmic Transformation and Discrete Cosine Transform (DCT): The output of the filter bank is subjected to a logarithmic transformation, followed by the inverse DFT, specifically the DCT, which results in a set of coefficients called the MFCC. These coefficients, including their average values (MFCC-Mean), are treated as multidimensional features for each frame.

The complete output matrix produced by this process is the MFCC, which provides a compact representation of the audio signal's spectral properties.

Using MFCC is particularly effective because, over small-time intervals, the audio signal does not change much, making it easier to analyze frame lengths between 20 to 40 milliseconds. If the frame is too short, the spectral estimate may be inaccurate due to insufficient samples. Conversely, a longer frame may capture sudden changes in the audio signal, causing potential distortions.

To calculate the power spectrum for each frame, periodograms can be used to identify the dominant frequencies within that frame. The relationship between frequency and the Mel scale, denoted as Mel(f), is expressed in Equation (1).

$$Mel(f) = 2595 \, . \, log_{10}\left(1 + {}^{f}/_{100}\right) \quad (1)$$

Where *Mel*(*f*) represents the Mel frequency and f is the frequency in Hz. This transformation allows the MFCC to use a filter bank in which higher frequencies occupy more bandwidth than lower frequencies, but the time resolution remains constant. The final step is to apply the DCT to filter the resulting output. The MFCC extraction process can be broken down into the following stages:

1. Normalized Processing: This step normalizes the input audio signal $x(n)$ to prepare it for subsequent stages.
2. Framing: The normalized audio vector is converted into a matrix of frames. Each time series is transformed into a sequence of overlapping frames, with zero padding applied if necessary.
3. Windowing is applied to reduce frame discontinuities that occur after framing. Common window functions include the Hamming window (default), rectangular window, and triangular window. The Hamming window is mathematically defined in Equation (2).

$$w(n) = 0.5 \,.\left(1 - cos\,({}^{2\pi n}/_{(N\,-\,1)})\right), 0 \leq n \leq N - 1 \qquad (2)$$

4. Fast Fourier Transform (FFT): The FFT is applied to the windowed data, transforming it into the frequency domain, as shown in Equation (3).

$$X_a(k) = \sum_{n=0}^{N-1} x(n)e^{-jw_k},\ 0 \leq k \leq N \qquad (3)$$

Where N is N is the length $x(n)$ above that is the input sound of the $\omega_k = \frac{2\pi}{N}k$ signal.

5. Filter Bank Processing: The Mel filter bank, designed to replicate human hearing, is applied to the power spectrum. The filter's center frequencies are distributed along the Mel scale and are calculated using Equation (4).

$$f(l) = \frac{N}{Fs} f^{-1}\left(f(L_l) + l\,.\frac{f(U_l)-f(L_l)}{M+1}\right) \qquad (4)$$

Where Fs is the actual gain frequency, N is the FFT length, M is the number of filters, $f(U_l)$ is the upper frequency of a single filter, $f(L_l)$ is the lower limit frequency of a single filter, and f^{-1} is the representation of inverse the *f* function, which can be calculated from the Equation (4).

6. Logarithmic Transformation: The power spectrum is transformed using a logarithmic function to simulate the human ear's logarithmic response, as shown in Equation (5).

$$S(m) = ln(\sum_{k=0}^{M-1}|X_a(k)|^2 H_m(K)),\ 0 \leq m \leq M \qquad (5)$$

Where $H_m(k)$ is the frequency of the triangle filter and $\sum_{k=0}^{M-1} H_m\,(K) = 1$, M is the number of right triangles, the value is generally 22-26.

7. MFCC Calculation: The MFCC coefficients are computed using the Discrete Cosine Transform (DCT), as shown in Equation (6).

$$C(n) = \sum_{m=0}^{N-1} s(m)\cos\left(\frac{\pi n(m-0.5)}{M}\right), n = 1,2,\ldots,L \qquad (6)$$

Where L is the sum of MFCC coefficients, the values are generally 13–39, and M is the number of triangular filters [18].

Geometric Transformation:
The value of the cepstral coefficients will be used as input parameters for training each class in the machine learning model.

Evaluation of the confusion matrix

The Evaluation Method is a way to determine the evaluation metrics used in the challenge to assess how well the model recognizes audio data in a specific music genre or class. One commonly used evaluation method is the confusion matrix.

RESULTS AND DISCUSSIONS

The experiment begins with the pre-processing stage by testing a specific song to be used as an example or as a tester for classification prediction, utilizing feature functions from the librosa library in Python. One example is the song Tetap Dalam Jiwa by Isyana Sarasvati, in .mp3 format. There is a parameter for the number of MFCCs, as shown in Figure 1. MFCC [19] coefficients from mfcc_1 to mfcc_20 can be observed in the framing waveforms represented by white lines, where each line represents MFCC features from 1 to 20, with specific power levels in decibels ranging from -400 dB to +100 dB. In the following section, there is a time parameter in seconds, with an approximate duration of 3 minutes and 20 seconds.

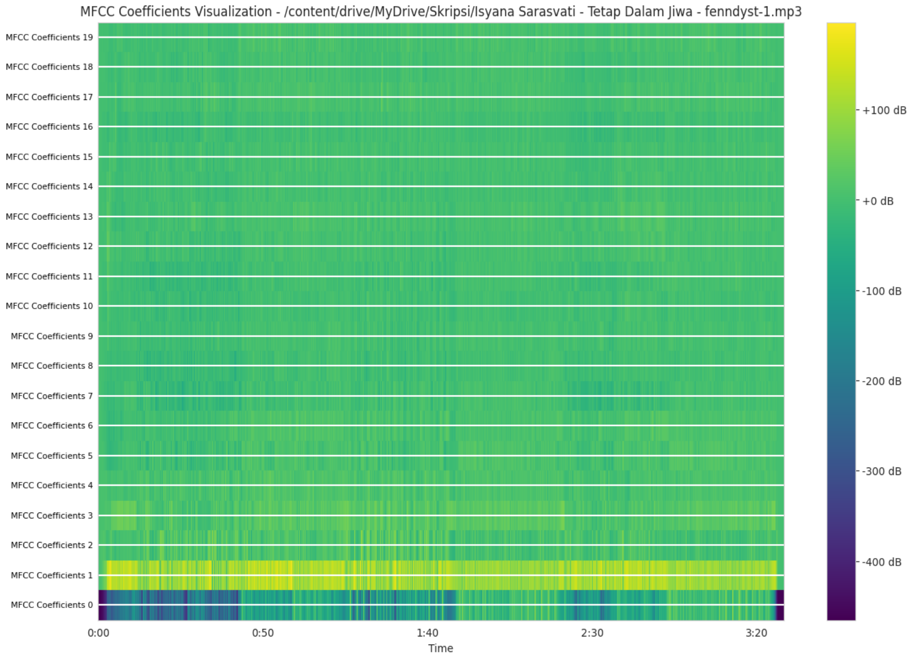


Figure 1. Visualization of 20 MFCC features from isyana sarasvati's song "stay in the soul"

The results of the MFCC for each sampling rate can be visualized in the form of a mel frequency wave graph, as shown in Figure 2.

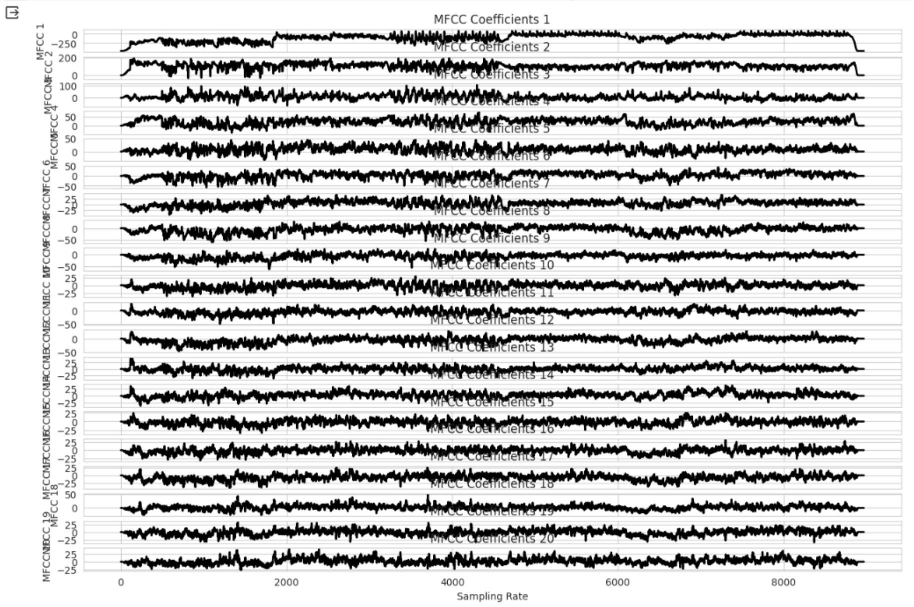
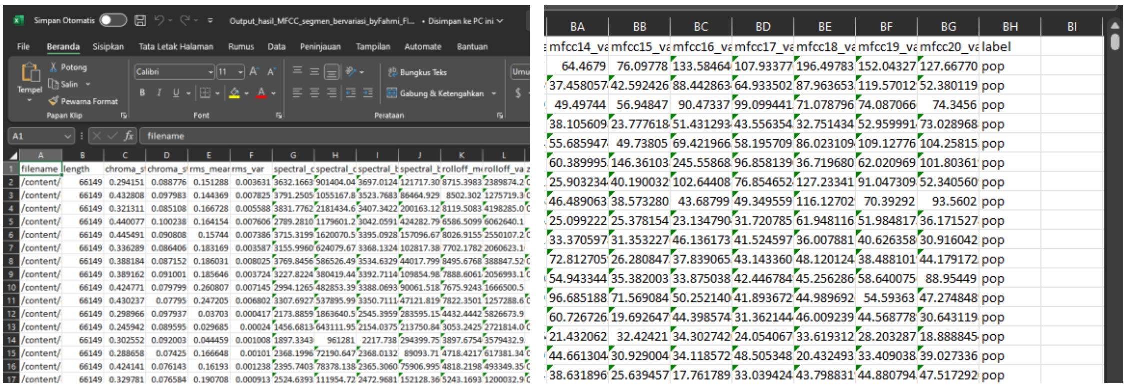


Figure 2. Mel frequency spectrogram of isyana sarasvati's song "stay in the soul"

The number of spectral coefficients typically used is 12 and 24. Higher-order coefficients tend to perform better than lower-order ones [20]. The selection of 20 coefficients in this study was made to maintain consistency with the GTZAN dataset available on Kaggle. This dataset only provides 20 coefficients in the features_30_sec file, so this number was chosen to ensure equality in the number of acoustic features among the music samples in genre classification. It is important to ensure that no feature dominance skews the training and testing process of the model. Additionally, the length used in this analysis is 66,149, to ensure that each music track has a fair segment and a uniform signal wavelength. This ensures that comparisons between music tracks in genre classification are not influenced by differences in signal length. This is illustrated in Figure 3.

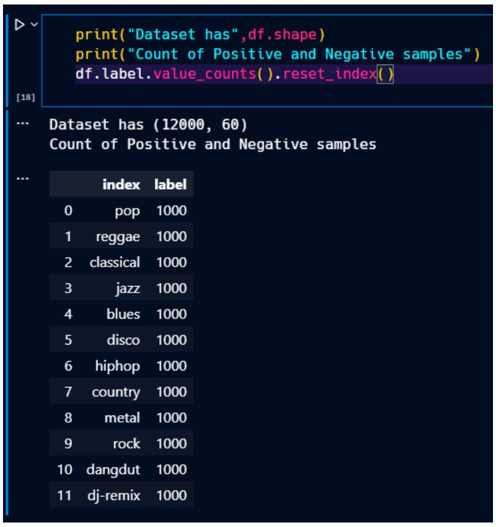


	A	B	C	D	E	F	G	H	I	J	K	L
1	filename	length	chroma_s	chroma_s	rms_mear	rms_var	spectral_c	spectral_c	spectral_b	spectral_b	rolloff_m	rolloff_va
2	/content/	66149	0.294151	0.088776	0.151288	0.003631	3632.1663	901404.04	3697.0124	121717.30	8715.3983	2389874.2
3	/content/	66149	0.432808	0.097983	0.144369	0.007825	3791.2505	1055167.8	3523.7683	86464.929	8502.302	1275719.3
4	/content/	66149	0.321311	0.085108	0.166728	0.005588	3831.7762	2181434.6	3407.3422	200163.12	8119.5083	4198285.0
5	/content/	66149	0.440077	0.100238	0.164154	0.007606	2789.2810	1179601.2	3042.0591	424282.79	6586.5099	6062640.1
6	/content/	66149	0.445491	0.090808	0.15744	0.007386	3715.3199	1620070.5	3395.0928	157096.67	8026.9155	2550107.2
7	/content/	66149	0.336289	0.086406	0.183169	0.003587	3155.9960	624079.67	3368.1324	102817.38	7702.1782	2060623.1
8	/content/	66149	0.388184	0.087152	0.186031	0.008025	3769.8456	586526.49	3534.6329	44017.799	8495.6768	388847.52
9	/content/	66149	0.389162	0.091001	0.185646	0.003724	3227.8224	380419.44	3392.7114	109854.98	7888.6061	2056993.1
10	/content/	66149	0.424771	0.079799	0.260807	0.007145	2994.1265	482853.39	3388.0693	90061.518	7675.9243	1666500.5
11	/content/	66149	0.430237	0.07795	0.247205	0.006802	3307.6927	537895.99	3350.7111	47121.819	7822.3501	1257288.6
12	/content/	66149	0.298966	0.097937	0.03703	0.000417	2173.8859	1863640.5	2545.3959	283595.15	4432.4442	5826673.9
13	/content/	66149	0.245942	0.089595	0.029685	0.00024	1456.6813	643111.95	2154.0375	213750.84	3053.2425	2721814.0
14	/content/	66149	0.302552	0.092003	0.044459	0.001008	1897.3343	961281	2217.738	294399.75	3897.6754	3579432.9
15	/content/	66149	0.288658	0.07425	0.166648	0.00101	2368.1996	72190.647	2368.0132	89093.71	4718.4217	617381.34
16	/content/	66149	0.424141	0.076143	0.16193	0.001238	2395.7403	78378.138	2365.3060	75906.995	4818.2198	493349.35
17	/content/	66149	0.329781	0.076584	0.190708	0.000913	2524.6393	111954.72	2472.9681	152128.36	5243.1693	1200032.9

BA	BB	BC	BD	BE	BF	BG	BH	BI
mfcc14_va	mfcc15_va	mfcc16_va	mfcc17_va	mfcc18_va	mfcc19_va	mfcc20_va	label	
64.4679	76.09778	133.58464	107.93377	196.49783	152.04327	127.66770	pop	
37.458057	42.592426	88.442863	64.933502	87.963653	119.57012	52.380119	pop	
49.49744	56.94847	90.47337	99.099441	71.078796	74.087066	74.3456	pop	
38.105609	23.777618	51.431293	43.556354	32.751434	52.959991	73.028968	pop	
55.685947	49.73805	69.421966	58.195709	86.023109	109.12776	104.25815	pop	
60.389995	146.36103	245.55868	96.858139	36.719680	62.020969	101.80361	pop	
25.903234	40.190032	102.64408	76.854652	127.23341	91.047309	52.340560	pop	
46.489063	38.573280	43.68799	49.349559	116.12702	70.39292	93.5602	pop	
25.099222	25.378154	23.134790	31.720785	61.948116	51.984817	36.171527	pop	
33.370597	31.353227	46.136173	41.524597	36.007881	40.626358	30.916042	pop	
72.812705	26.280847	37.839065	43.143360	48.120124	38.488101	44.179172	pop	
54.943344	35.382003	33.875038	42.446784	45.256286	58.640075	88.95449	pop	
96.685188	71.569084	50.252140	41.893672	44.989692	54.59363	47.274848	pop	
60.726726	19.692647	44.398574	31.362144	46.009239	44.568778	30.643119	pop	
21.432062	32.42421	34.302742	24.054067	33.619312	28.203287	18.888845	pop	
44.661304	30.929004	34.118572	48.505348	20.432493	33.409038	39.027336	pop	
38.631896	25.639457	17.761789	33.039424	43.798831	44.880794	47.517292	pop	

Figure 3. Results of MFCC extraction with consistent signal length

In the dataset, there are columns such as filename, length, chroma_mean, chroma_var, rms_mean, rms_var, spectral_centroid_mean, spectral_centroid_var, spectral_bandwidth_mean, spectral_bandwidth_var, and mfcc1 through mfcc20, with the genre label information at the end. The total number of columns is 60, while the row index is 1,200 x 10 = 12,000 rows, resulting in a total of 720,000 data points. This is illustrated in Figure 4. To explore the data, it can be differentiated based on the label, where the genre label contains 12 classes, each with 1,000 samples, totaling 12,000 rows and 60 columns.



```
print("Dataset has",df.shape)
print("Count of Positive and Negative samples")
df.label.value_counts().reset_index()
```

[18]

```
Dataset has (12000, 60)
Count of Positive and Negative samples
```

	index	label
0	pop	1000
1	reggae	1000
2	classical	1000
3	jazz	1000
4	blues	1000
5	disco	1000
6	hiphop	1000
7	country	1000
8	metal	1000
9	rock	1000
10	dangdut	1000
11	dj-remix	1000

Figure 4. Results of data check in the dataset

A correlation heatmap for the average variables in Mel Frequency Cepstral Coefficients (MFCC) is used to analyze the relationships between different features or dimensions in the MFCC data. By examining the correlation heatmap, one can identify patterns of positive or negative relationships between these variables. Additionally, this heatmap aids in feature selection by highlighting variables with high correlations, thus facilitating decision-making regarding dimensionality reduction or the removal of features that may not provide additional information. This offers crucial insights for signal processing and model development decision-making. This is illustrated in Figure 5.

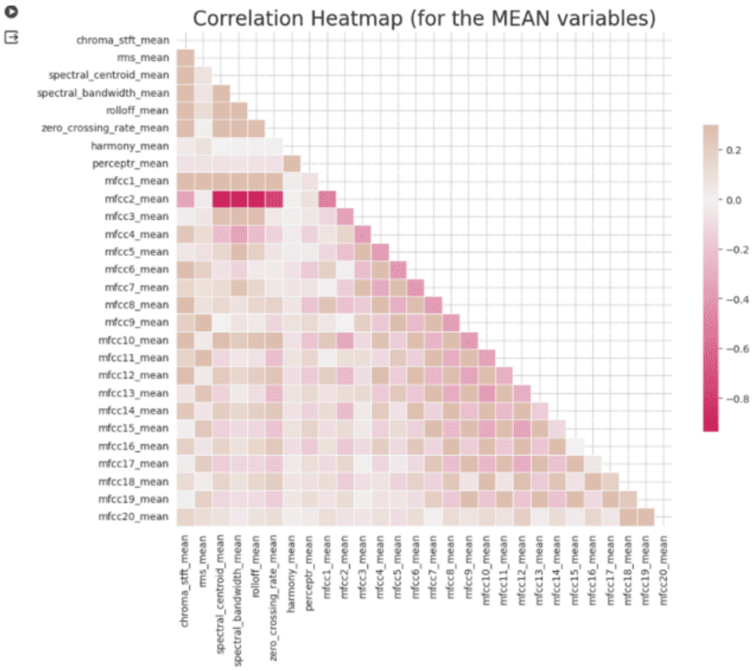


Figure 5. Correlation heatmap of MFCC mean variables

The box plot for each music genre, as seen in Figure 6, provides information on beats per minute or tempo within the music itself. From pop music to DJ remixes, each genre has its own rhythmic flow that can vary—some are slow, while others are fast. If we draw a conclusion, the larger the box plot (e.g., blues), the more variation in tempo exists within that genre, indicating a wide range of songs with different tempos. On the other hand, a narrower box plot (e.g., disco) suggests that the genre is more uniform, with songs having similar tempos. From this, it can be concluded that the music genres with the highest tempo variability are classical, reggae, and blues, while genres with more stable and moderate tempos include disco. For slower tempo genres, we observe DJ-remix, country, or national anthems.

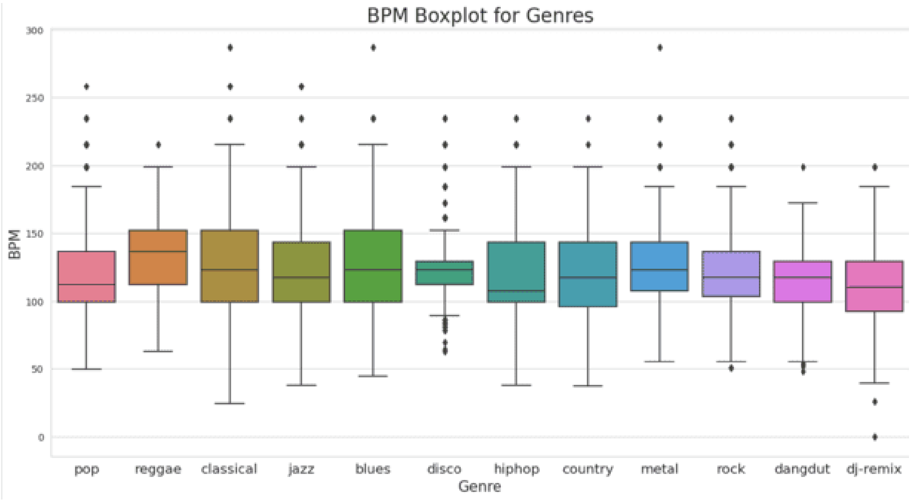


Figure 6. BPM or tempo results for each genre

In Figure 7, the application of Principal Component Analysis (PCA) to the music genres offers several advantages. PCA helps in understanding the distribution and variation of song data across different genres. By identifying the principal components, PCA emphasizes the most significant variations within the data, making it easier to analyze and grasp the relationships between different music genres. This technique enables simplified data representation, improved computational efficiency, and provides deeper insights into the musical characteristics that are unique to each genre.

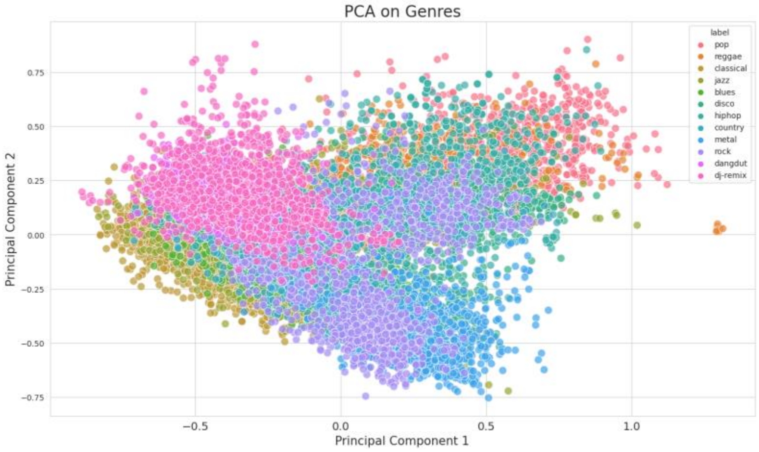
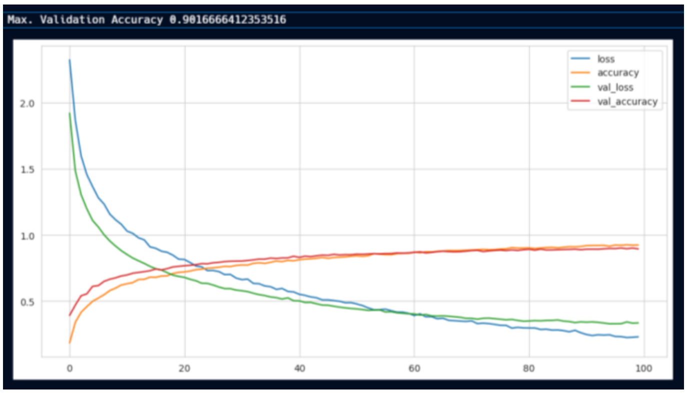


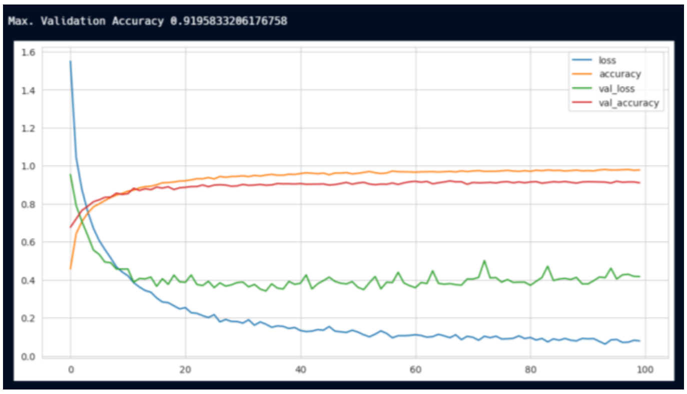
Figure 6. Results of PCA showing variation distribution across genres

The experiment was conducted using two different optimizers, namely SGD and Adam, with 100 epochs. This time, a dropout of 0.3 (DO = 0.3) was applied, which improved the model's stability over long iterations. The results are shown in Figures 8 and 9. Using dropout helped stabilize training and led to better results at higher epochs.



```
Max. Validation Accuracy 0.9016666412353516
```

Figure 8. Accuracy history of the ANN model with SGD optimizer

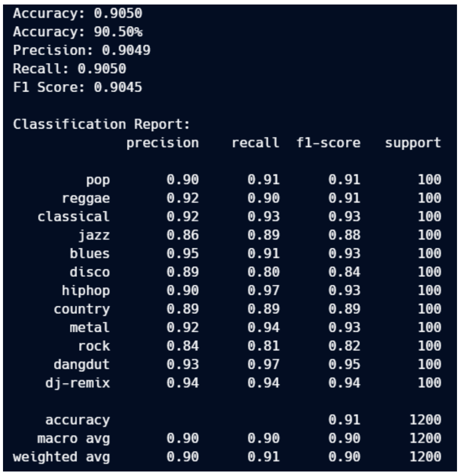


```
Max. Validation Accuracy 0.9195833206176758
```

Figure 9. Accuracy history of the ANN model with adam optimizer

In Figure 10, the maximum validation accuracy achieved was 91.95%. When compared to Figure 11, where the validation accuracy reached 92.16%, it is evident that the Adam optimizer provided slightly better results. Figure 12 displays the confusion matrix, with the red boxes highlighting the recall values for each genre. The classification report containing detailed metrics such as recall, accuracy, F1-score, support, and precision for each genre. The recall values for each genre are as follows: pop 91%, reggae 90%, classical

93%, jazz 89%, blues 91%, disco 80%, hip-hop 97%, country 89%, metal 94%, rock 81%, dangdut 97%, and DJ-remix 94%.



```
Accuracy: 0.9050
Accuracy: 90.50%
Precision: 0.9049
Recall: 0.9050
F1 Score: 0.9045

Classification Report:
              precision    recall  f1-score   support

         pop       0.90      0.91      0.91       100
      reggae       0.92      0.90      0.91       100
   classical       0.92      0.93      0.93       100
        jazz       0.86      0.89      0.88       100
       blues       0.95      0.91      0.93       100
       disco       0.89      0.80      0.84       100
      hiphop       0.90      0.97      0.93       100
     country       0.89      0.89      0.89       100
       metal       0.92      0.94      0.93       100
        rock       0.84      0.81      0.82       100
     dangdut       0.93      0.97      0.95       100
    dj-remix       0.94      0.94      0.94       100

    accuracy                           0.91      1200
   macro avg       0.90      0.90      0.90      1200
weighted avg       0.90      0.91      0.90      1200
```

Figure 10. Comprehensive classification report of the ANN model

The overall results are summarized in Table 3. An additional experiment was conducted with more epochs, specifically 100 epochs, yielding the following results.

Table 3. Accuracy results – different optimizers

E	D	O	L	A	VL	VA	Max V_A
1	null	Adam	0.01	0.99	0.56	0,91	0,9216
2	0.3	Adam	0.07	0.97	0.41	0,90	0,9195
3	0.3	SGD	0.22	0.92	0.33	0,89	0,9016
Average			0.10	0.97	0.43	0.90	0.9142

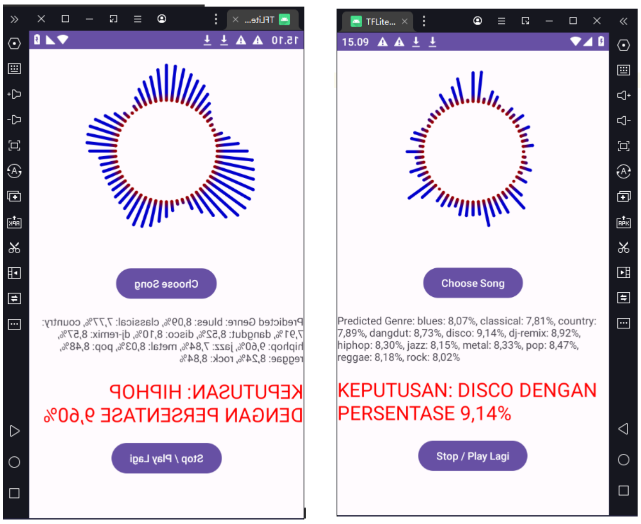
Description: L = Loss, E = Experiment, A = Accuracy, DO = Dropout, VL = Validation Loss, O = Optimizer, and VA = Validation Accuracy.

In Table 3, it is evident that when evaluating the model, it's crucial to not only focus on accuracy but also consider loss and validation loss. These metrics provide a more comprehensive picture when deciding which ANN model and parameters perform best for the task. This approach ensures optimal results in predicting song genre classification.

From the data, we observe that in the first experiment, the maximum validation accuracy reached 92.16%, which is quite high. However, the validation loss is also the highest. The second experiment shows a more balanced performance, with moderate accuracy and low validation loss. The third experiment has the lowest validation loss (33%), though the accuracy is slightly lower than in the first experiment. Despite this, model_3 is chosen for the song genre classification task due to its significantly lower validation loss.

The average results from all three experiments are as follows. The model demonstrated an average loss of 10.28%, with an overall accuracy of 97.37%. The validation loss was observed to be 43.85%, while the validation accuracy averaged 90.68%. Additionally, the maximum validation accuracy across the experiments reached an average of 91.42%. These metrics indicate a robust performance, with high accuracy and relatively low loss, suggesting that the model was well-optimized for the task at hand. Thus, when selecting the best model, it is recommended to prioritize minimizing validation loss, even if the difference in accuracy is marginal.

Once the song is selected, the system records the file using specific functions. Feature extraction is then performed on the song, and the model predicts its genre. The prediction is displayed in the "prediction text view" within the Android XML layout, as shown in Figure 11 (from the LDPlayer emulator) and Figure 12 (from the actual Android device).



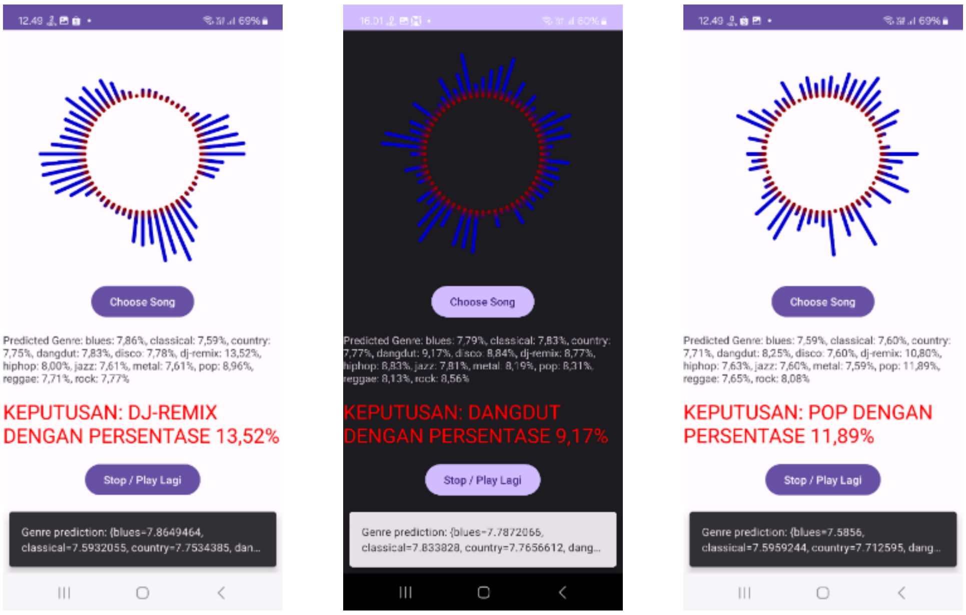
Predicted Genre: blues: 8,09%, classical: 7,77%, country: 7,91%, dangdut: 8,52%, disco: 8,10%, dj-remix: 8,57%, hiphop: 9,60%, jazz: 7,84%, metal: 8,03%, pop: 8,48%, reggae: 8,24%, rock: 8,84%

KEPUTUSAN: HIPHOP DENGAN PERSENTASE 9,60%

Predicted Genre: blues: 8,07%, classical: 7,81%, country: 7,89%, dangdut: 8,73%, disco: 9,14%, dj-remix: 8,92%, hiphop: 8,30%, jazz: 8,15%, metal: 8,33%, pop: 8,47%, reggae: 8,18%, rock: 8,02%

KEPUTUSAN: DISCO DENGAN PERSENTASE 9,14%

Figure 11. Song type category classification results on emulator



Predicted Genre: blues: 7,86%, classical: 7,59%, country: 7,75%, dangdut: 7,83%, disco: 7,78%, dj-remix: 13,52%, hiphop: 8,00%, jazz: 7,61%, metal: 7,61%, pop: 8,96%, reggae: 7,71%, rock: 7,77%

KEPUTUSAN: DJ-REMIX DENGAN PERSENTASE 13,52%

Genre prediction: {blues=7.8649464, classical=7.5932055, country=7.7534385, dan...

Predicted Genre: blues: 7,79%, classical: 7,83%, country: 7,77%, dangdut: 9,17%, disco: 8,84%, dj-remix: 8,77%, hiphop: 8,83%, jazz: 7,81%, metal: 8,19%, pop: 8,31%, reggae: 8,13%, rock: 8,56%

KEPUTUSAN: DANGDUT DENGAN PERSENTASE 9,17%

Genre prediction: {blues=7.7872066, classical=7.833828, country=7.7656612, dang...

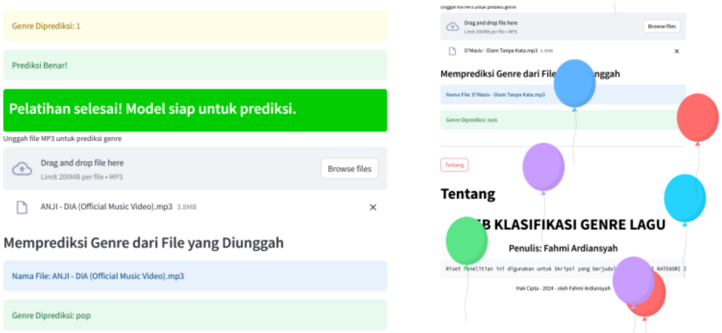
Predicted Genre: blues: 7,59%, classical: 7,60%, country: 7,71%, dangdut: 8,25%, disco: 7,60%, dj-remix: 10,80%, hiphop: 7,63%, jazz: 7,60%, metal: 7,59%, pop: 11,89%, reggae: 7,65%, rock: 8,08%

KEPUTUSAN: POP DENGAN PERSENTASE 11,89%

Genre prediction: {blues=7.5856, classical=7.5959244, country=7.712595, dang...

Figure 12. Song type category classification results on samsung galaxy smartphone

In the blue status info section, the song's filename is displayed (e.g., "ANJI – DIA (Official Music Video).mp3"), based on the user input, as shown in Figure 13. Below that, in the green status_info section, the music genre prediction is displayed. However, it should be noted that in some cases, the predicted results may not align with expectations.



Genre Diprediksi: 1

Prediksi Benar!

Pelatihan selesai! Model siap untuk prediksi.

Unggah file MP3 untuk prediksi genre

ANJI - DIA (Official Music Video).mp3 3.8MB

Memprediksi Genre dari File yang Diunggah

Nama File: ANJI - DIA (Official Music Video).mp3

Genre Diprediksi: pop

Unggah file MP3 untuk prediksi genre

D'Masiv - Diam Tanpa Kata.mp3

Memprediksi Genre dari File nggah

Nama File: D'Masiv - Diam Tanpa Kata.mp3

Genre Diprediksi: rock

Tentang

B KLASIFIKASI GENRE LAGU

Penulis: Fahmi Ardiansyah

Riset Penelitian ini digunakan untuk Skripsi yang berjudul KATEGORI

Hak Cipta - 2024 - oleh Fahmi Ardiansyah

Figure 16. System implementation results on streamlit-based web application

The system implementation on both Android and web platforms was successful, yielding intriguing predictive results. However, it was observed that in some cases, songs by the vocalist D'MASIV were incorrectly classified as the rock genre, even though they belong to the pop genre. This misclassification is likely due to the time limitation in processing, where only the first 30 seconds of the song were analyzed. As a result, the feature extraction during this time frame may not capture the full essence and characteristics of the song. To improve prediction accuracy and better align with the actual genre of the music, further computational refinement may be necessary. Despite this limitation, the study offers valuable insights into music genre classification across both Android and web platforms.

This research demonstrates that the correct combination of methods and approaches can significantly enhance the accuracy of music genre classification. Using the Librosa library from the pre-processing stage, feature extraction with the MFCC method, and dividing the dataset into training, development, and testing sets with specific percentages, the model was fine-tuned by selecting appropriate hyperparameters such as Epochs, Batch Size, Dropout Rate, Activation Functions, and Optimizers. By applying the Feedforward Propagation Fully Connected Multilayer Perceptron (FFP-FCMLP) [21] algorithm in an ANN with a carefully calibrated composition, the study achieved high accuracy in genre classification [9], [20], [22]–[24] surpassing results from previous studies. For a detailed comparison, refer to Table 4.

Table 4. Comparison of results

Research	Accuracy (%)
(Bawitlung & Dash, 2024) [5]	85
(Cai & Zhang, 2022) [6]	91.8
(Mu, 2023) [7]	89.03
(Miri et al., 2020) [8]	75
(Xie et al., 2023) [9]	83
(Liu et al., 2023) [10]	82.71
(Kakarla et al., 2022) [11]	84
Proposed Method	92.16

CONCLUSION

Based on the research results and experiments conducted in this study, the utilization of MFCC successfully extracted relevant audio features. With the combination of MFCC and Artificial Neural Networks (ANN), this study achieved the highest accuracy of 92.16% and an average accuracy of 91.42%, demonstrating that this method is highly effective in classifying music genres. Additionally, increasing the amount of data through feature extraction repetition (10 times per song) and adding data from dynamic sources like TikTok and YouTube significantly enhanced data diversity and model accuracy. This highlights the importance of data variation in improving classification model performance.

REFERENCES

[1] C. Yang, F. Möttig, J. Weitz, C. Reissfelder, and S. T. Mees, “Effect of Genre and amplitude of music during laparoscopic surgery,” *Langenbeck’s Arch. Surg.*, vol. 407, no. 5, pp. 2115–2121, Mar. 2022, doi: 10.1007/s00423-022-02490-z.

[2] S. Rajesh and Nalini N. J., “Recognition of Musical Instrument Using Deep Learning Techniques,” *Int. J. Inf. Retr. Res.*, vol. 11, no. 4, pp. 41–60, Oct. 2021, doi: 10.4018/IJIRR.2021100103.

[3] H. Xiang, “The collection of theater music data and genre recognition under the internet of things and deep belief network,” *J. Supercomput.*, vol. 78, no. 7, pp. 9307–9325, May 2022, doi: 10.1007/s11227-021-04261-x.

[4] M. S. Sidhu, N. A. A. Latib, and K. K. Sidhu, “MFCC in audio signal processing for voice disorder: a review,” *Multimed. Tools Appl.*, Apr. 2024, doi: 10.1007/s11042-024-19253-1.

[5] A. Bawitlung and S. K. Dash, “Genre Classification in Music using Convolutional Neural Networks,” 2024, pp. 397–409. doi: 10.1007/978-981-99-7339-2_33.

[6] X. Cai and H. Zhang, “Music genre classification based on auditory image, spectral and acoustic features,” *Multimed. Syst.*, vol. 28, no. 3, pp. 779–791, Jun. 2022, doi: 10.1007/s00530-021-00886-3.

[7] X. Mu, “Implementation of Music Genre Classifier Using KNN Algorithm,” *Highlights Sci. Eng. Technol.*, vol. 34, pp. 149–154, Feb. 2023, doi: 10.54097/hset.v34i.5439.

[8] K. Miri, E. Enriquez, and A. Donohue, “Music Genre Classification Using A Convolutional Neural Network,” 2017. [Online]. Available: https://cs230.stanford.edu/projects_spring_2020/reports/38954030.pdf

[9] C. Xie *et al.*, "Music genre classification based on res-gated CNN and attention mechanism," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 13527–13542, Jul. 2023, doi: 10.1007/s11042-023-15277-1.

[10] Z. Liu, T. Bian, and M. Yang, "Locally Activated Gated Neural Network for Automatic Music Genre Classification," *Appl. Sci.*, vol. 13, no. 8, p. 5010, Apr. 2023, doi: 10.3390/app13085010.

[11] C. Kakarla, V. Eshwarappa, L. Babu Saheer, and M. Maktabdar Oghaz, "Recurrent Neural Networks for Music Genre Classification," 2022, pp. 267–279. doi: 10.1007/978-3-031-21441-7_19.

[12] K. A. Al-karawi and D. Y. Mohammed, "Improving short utterance speaker verification by combining MFCC and Entrocy in Noisy conditions," *Multimed. Tools Appl.*, vol. 80, no. 14, pp. 22231–22249, Jun. 2021, doi: 10.1007/s11042-021-10767-6.

[13] M. A. Nasr, M. Abd-Elnaby, A. S. El-Fishawy, S. El-Rabaie, and F. E. Abd El-Samie, "Speaker identification based on normalized pitch frequency and Mel Frequency Cepstral Coefficients," *Int. J. Speech Technol.*, vol. 21, no. 4, pp. 941–951, Dec. 2018, doi: 10.1007/s10772-018-9524-7.

[14] U. Kumaran, S. Radha Rammohan, S. M. Nagarajan, and A. Prathik, "Fusion of mel and gammatone frequency cepstral coefficients for speech emotion recognition using deep C-RNN," *Int. J. Speech Technol.*, vol. 24, no. 2, pp. 303–314, Jun. 2021, doi: 10.1007/s10772-020-09792-x.

[15] S. Raj, P. Prakasam, and S. Gupta, "Multilayered convolutional neural network-based auto-CODEC for audio signal denoising using mel-frequency cepstral coefficients," *Neural Comput. Appl.*, vol. 33, no. 16, pp. 10199–10209, Aug. 2021, doi: 10.1007/s00521-021-05782-5.

[16] S. P. Mishra, P. Warule, and S. Deb, "Speech emotion recognition using MFCC-based entropy feature," *Signal, Image Video Process.*, vol. 18, no. 1, pp. 153–161, Feb. 2024, doi: 10.1007/s11760-023-02716-7.

[17] S. Patnaik, "Speech emotion recognition by using complex MFCC and deep sequential model," *Multimed. Tools Appl.*, vol. 82, no. 8, pp. 11897–11922, Mar. 2023, doi: 10.1007/s11042-022-13725-y.

[18] Z. Yang and Y. Huang, "Algorithm for speech emotion recognition classification based on Mel-frequency Cepstral coefficients and broad learning system," *Evol. Intell.*, vol. 15, no. 4, pp. 2485–2494, Dec. 2022, doi: 10.1007/s12065-020-00532-3.

[19] T. Mourad, "Arabic Speech Recognition by Stationary Bionic Wavelet Transform and MFCC Using a Multi-layer Perceptron for Voice Control," 2022, pp. 69–81. doi: 10.1007/978-3-030-93405-7_4.

[20] V. H. da Silva Muniz and J. B. de Oliveira e Souza Filho, "Robust handcrafted features for music genre classification," *Neural Comput. Appl.*, vol. 35, no. 13, pp. 9335–9348, May 2023, doi: 10.1007/s00521-022-08069-5.

[21] W. Liu, H. Moayedi, H. Nguyen, Z. Lyu, and D. T. Bui, "Proposing two new metaheuristic algorithms of ALO-MLP and SHO-MLP in predicting bearing capacity of circular footing located on horizontal multilayer soil," *Eng. Comput.*, vol. 37, no. 2, pp. 1537–1547, Apr. 2021, doi: 10.1007/s00366-019-00897-9.

[22] V. T. M and S. T. R, "Music genre classification using convolution temporal pooling network," *Multimed. Tools Appl.*, Sep. 2024, doi: 10.1007/s11042-024-20163-5.

[23] N. Narkhede, S. Mathur, A. Bhaskar, and M. Kalla, "Music genre classification and recognition using convolutional neural network," *Multimed. Tools Appl.*, Apr. 2024, doi: 10.1007/s11042-024-19243-3.

[24] Y. Li, Z. Zhang, H. Ding, and L. Chang, "Music genre classification based on fusing audio and lyric information," *Multimed. Tools Appl.*, vol. 82, no. 13, pp. 20157–20176, May 2023, doi: 10.1007/s11042-022-14252-6.