

# Kurdish Music Genre Recognition Using a CNN and DNN <sup>†</sup>

Aza Kamala <sup>\*,‡</sup>  and Hossein Hassani <sup>‡</sup> 

Department of Computer Science and Engineering, School of Science and Engineering,  
University of Kurdistan Hewlêr, Erbil 44001, Kurdistan Region, Iraq; hosseinh@ukh.edu.krd

\* Correspondence: aza.zuhairamin@ukh.edu.krd

† Presented at the 3rd International Electronic Conference on Applied Sciences, 1–15 December 2022; Available online: <https://asec2022.sciforum.net/>.

‡ These authors contributed equally to this work.

**Abstract:** Music has different styles, and they are categorized into genres by musicologists. Nonetheless, non-musicologists categorize music differently, such as by finding similarities and patterns in instruments, harmony, and the style of the music. For instance, in addition to popular music genre categorization, such as classic, pop, and modern folkloric, Kurdish music is categorized by Kurdish music lovers according to the type of dance that could go with a particular piece of music. Due to technological advancements, technologies such as artificial intelligence (AI) can help in music genre recognition. Using AI to recognize music genres has been a growing field lately. Computational musicology uses AI in various sectors of studying music. However, the literature shows no evidence of addressing any computational musicology research focusing on Kurdish music. In particular, we have not been able to find any work that indicates the usage of AI in the classification of Kurdish music genres. In this research, we compiled a dataset that comprises 880 samples of 8 Kurdish music genres. We used two machine learning models in our experiments: a convolutional neural network (CNN) and a deep neural network (DNN). According to the evaluations, the CNN model achieved 92% accuracy, while the DNN achieved 90% accuracy. Therefore, we developed an application that uses the CNN model to identify Kurdish music genres by uploading or listening to Kurdish music.

**Keywords:** music information retrieval; music recognition; music genre classification; artificial intelligence



**Citation:** Kamala, A.; Hassani, H. Kurdish Music Genre Recognition Using a CNN and DNN. *Eng. Proc.* **2023**, *31*, 64. <https://doi.org/10.3390/ASEC2022-13803>

Academic Editor: Nunzio Cennamo

Published: 2 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The classification of music into genres is based on similarities in style, melody, and culture. Genre is a conceptual classification technique for music and other creative forms [1]. According to Norowi et al. (2005) [2], the purpose of computerized music genre categorization or music genre recognition (MGR) is to structure and organize a vast music collection. MGR is used in a variety of applications such as Tidal, Spotify, and Apple Music, especially for music information retrieval (MIR) [3] and recommendation systems [4].

Musicologists divide works of music into distinct genres based on their lyrics and melodies alone in the absence of automated methods. Nonetheless, as the number of music genres and the amount of music itself has grown, so has the need for automated music categorization [5]. The use of sophisticated machine learning approaches, such as neural networks, for the creation of more efficient and accurate music genre categorization has therefore received more interest in the field of computer science.

In this study, we use and assess the accuracy of deep neural network (DNN) and convolutional neural network (CNN) techniques for automatically categorizing Kurdish music into its various genres. We also develop an application that uses one of the machine learning models to identify Kurdish music genres. The remainder of the paper is structured as follows. Section 2 examines similar works. Section 3 outlines the data collection, training, and assessment methods. The results of the experiments are presented and discussed in Section 4. Finally, Section 5 brings the paper to a close.

## 2. Related Work

For music categorization, Ghosal and Kolekar (2018) [6] examined two distinct machine learning models: the CNN and long short-term memory (LSTM). They examined the performance of the models across a variety of feature types, including the Mel spectrogram, Mel coefficients, and Tonnetz features.

Feng (2014) [7] identified from two to four music genres using a deep belief network (DB), an unsupervised machine learning algorithm. To train the network, they used the GTZAN dataset, which included 1000 pieces of music from 10 distinct genres. They produced 15 samples per song by extracting the Mel frequency cepstral coefficient (MFCC) and constructed a dataset of 15000 samples to train and test the model (60% for training and 40% for testing). There were five layers in their DBF model. A restricted Boltzmann machine (RBM) was used for the iterative training of the connected layers. The model was 98.15% accurate for two genres, 69.16% for three, and 51.8% for four.

Silla et al. (2008) [8] implemented an ensemble learning approach for MGR based on spatial and temporal decomposition and analyzed the outcomes of distinct segments. They split each piece of music into three sections: the beginning, middle, and end. Ensemble learning combines various categorization methods to enhance accuracy. Decision tree, k-nearest neighbor (KNN), naive Bayes, multi-layer perceptron neural network, and support vector machine (SVM) techniques were the learning methods used in the stated experiment. According to the findings, the song's middle section performed better than the other two.

Tzanetakis and Cook (2002) used the KNN algorithm and the Gaussian mixture model for MGR. In their model, they included three audio characteristics: pitch, rhythm, and timbral texture [9].

In their MGR studies, Scheirer and Slaney (1997) [10] created a multidimensional classifier based on numerous models, including the maximum a posteriori (MAP) method, Gaussian mixture model (GMM), and KNN, and added 13 distinct musical characteristics. They trained the classifier using segment-by-segment audio data to reach an error rate of 5.8%. However, by increasing the length of the segments, the error rate was reduced to 1.4%.

In summary, the literature demonstrates the use of several machine learning approaches in MGR. In addition, it illustrates the diversity of data segmentation and dataset preparation techniques. We did not find any Kurdish MGR studies during our research, despite the rising interest in MGR.

## 3. Methodology

### 3.1. Data Collection

To investigate and comprehend the many Kurdish music genres, we sought out fine art professors, students, and artists. We gathered music from a variety of online and offline sources, including YouTube, SoundCloud, and music CDs, due to the absence of a ready-made dataset. We categorized the gathered music into several categories. Ahmadi et al. (2020) [11] cited Kurdish musical styles such as Bend, Gorani Meqam, and Hayran.

### 3.2. Feature Extraction

We utilized Librosa version (0.8.1), a popular Python package for audio and music analysis, to extract features for MRI applications. We categorized Kurdish music based on the MFCC (<https://musicinformationretrieval.com/mfcc.html> (accessed on 28 January 2021)) characteristics [12]. MFCC is a collection of characteristics used to describe the timbre of music. Librosa is used to divide 30-second audio files into 1200 segments. Lengthier segments perform better in audio classification, and hence the number of segments retrieved from music may vary [10].

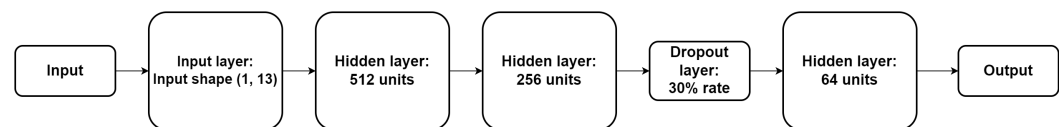
### 3.3. Data Preparation

Initialization involves randomizing the dataset and dividing it into training, validation, and testing datasets, which receive 70%, 10%, and 20% of the original dataset, respectively. The splitting ratio may vary based on the dataset size and findings.

### 3.4. Architecture of the Models

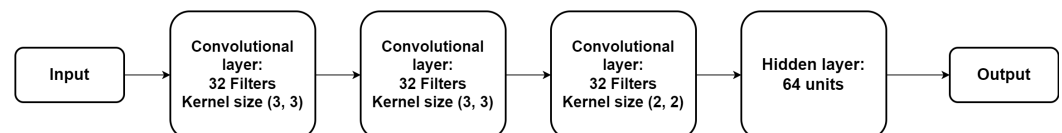
Using the TensorFlow library (<https://www.tensorflow.org/> (accessed on 26 June 2021)) and the Keras (<https://keras.io/> (accessed on 26 June 2021)) API, we implemented two architectures. The architectures were based on suggestions from Velardo (2021) [13]. During the evaluations, we may alter the model's architecture and number of epochs to increase its performance.

In Figure 1, we see a deep neural network (DNN) with four layers as the initial model. We flattened the data with the shape (1, 13) for the input layer and then fed them into the first 512 unit layer. The rectified linear (ReLU) function was used as the activation function. The activation function of the first layer was also used by the second layer, which comprised 256 units. To avoid model overfitting, we included a dropout layer with a rate of 0.3 between the second and third layers. The third layer had 64 units and used a similar activation function to the first two layers. The output layer classified the data using softmax activation.



**Figure 1.** Deep neural network architecture with modifications.

Figure 2 depicts the CNN model, which had five layers. The first layer was a two-dimensional (2D) convolutional layer with 32 filters and a kernel size of (3, 3). Its activation function was ReLu. The 2D max pooling layer had strides of (2, 2) and a pool size of (3, 3). We used a batch normalization layer to standardize the data inputs. The second layer had the same convolutional, max pooling, and batch normalization layers as the first. The third layer was a 2D convolutional layer with 32 filters and a (2, 2) kernel size, where ReLu was the activation function. Following the 2D convolutional layer was a 2D max pooling layer with a pool size of (2, 2). The max pooling layer's output was passed into a batch normalization layer. The fourth layer flattened the previous layer's output and sent it to a dense layer. The 64 unit dense layer had a ReLu activation function. The last layer predicted the genre using softmax.



**Figure 2.** Convolutional neural network architecture with modifications.

### 3.5. Evaluation and Testing

Accuracy ([https://www.tensorflow.org/api\\_docs/python/tf/keras/metrics/Accuracy](https://www.tensorflow.org/api_docs/python/tf/keras/metrics/Accuracy) (accessed on 1 July 2021)), a TensorFlow metric, was used to assess the accuracy of the models. The metric is shown in Equation (1).

$$accuracy = frequency / totalsamples \quad (1)$$

### 3.6. Software Application

We created an application using the Python programming language library PySimpleGUI (<https://www.pysimplegui.org/> (accessed on 12 August 2021)). The application

used one of the proposed machine learning models. The application should listen to music or let the user choose from their device to predict Kurdish music.

## 4. Experiments and Results

### 4.1. Data Collection

We gathered 208 pieces of music across 8 genres. Following the recommendation Silla et al. (2008) [8], we picked samples from the middle of the music to preserve its general tone. Because of the minimal amount of samples that we had for one of the genres, Halparke Se-pey, which had 110 samples, we randomly chose 110 samples from all the other genres too.

### 4.2. Experiments

Throughout our experiments, adjustments were made to the models in order to enhance them. Tables 1 and 2 display these alterations in addition to the characteristics used for training and assessing the models in each experiment.

**Table 1.** The DNN model's experimental outcomes.

No.	Input Shape	Total Samples	Randomization Algorithm	Epochs	Accuracy	Loss
1	(2, 13)	1,056,000	Random Split	30	74%	72%
2	(2, 13)	200,000	Random Split	30	67%	90%
3	(2, 13)	1,056,000	K-Fold Cross-Validation	30	74%	71%
4	(2, 13)	1,056,000	K-Fold Cross-Validation	100	75%	69%
5	(44, 13)	26,400	K-Fold Cross-Validation	30	90%	39%
6	(44, 13)	26,400	K-Fold Cross-Validation	30	90%	32%

**Table 2.** The CNN model's experimental outcomes.

No.	Input Shape	Total Samples	Randomization Algorithm	Epochs	Accuracy	Loss
1	(2, 13, 1)	1,056,000	Random Split	30	67%	88%
2	(2, 13, 1)	200,000	Random Split	30	64%	98%
3	(2, 13, 1)	1,056,000	K-Fold Cross-Validation	30	67%	88%
4	(2, 13, 1)	1,056,000	K-Fold Cross-Validation	100	69%	85%
5	(44, 13, 1)	26,400	K-Fold Cross-Validation	30	92%	23%
6	(44, 13, 1)	26,400	K-Fold Cross-Validation	30	92%	25%

### 4.3. Discussion

The initial experiment for both models performed poorly on the test dataset, and the loss was over 70%. In a second experiment, we trained and evaluated the models using a smaller dataset, but the accuracy dropped.

In the third trial, we replaced the randomization technique with the cross-validation algorithm. The final distributed group (Fold) served as the dataset for the remaining trials.

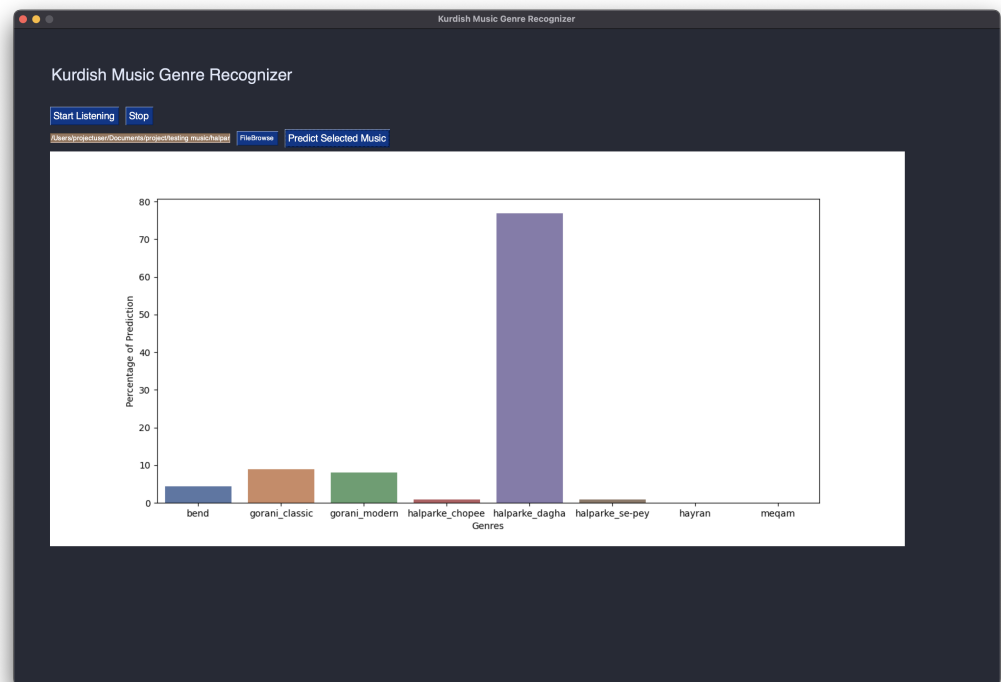
We raised the number of epochs for both the DNN and CNN models to 100 for the fourth experiment [14]. Although the DNN model's validation accuracy was lower than the training accuracy, it did not do well in recognizing genres. In contrast, the CNN model yielded comparable results to the prior experiment.

The classifier showed a lower rate of error while dealing with lengthier samples, such as samples with a duration of 2.4 s [10]. In the fifth trial, we increased the amount of samples from each 30 s track to 30. The fifth trial had a good outcome, with the DNN achieving 90% accuracy and the CNN achieving 92% accuracy.

Looking back on the trials, the sixth experiment produced the best outcome for the DNN model. We attained 90% accuracy with a 32% error rate. In addition, the CNN model performed best in the fifth experiment, since its error rate was lower than that of the sixth experiment, indicating that it made less mistakes on the testing dataset. The model's accuracy was 92% with a loss of 23%.

#### 4.4. Software Application

Figure 3 illustrates the application we developed to recognize Kurdish music genres. The application has a button that says “Start Listening” to start recording music and a button that says “Stop” to stop recording. Users will be able to record music in this manner, and then the program will predict the genre of the music. Users may also pick music from their device by pressing the “File Browse” button. The program will then allow the user to select music from his or her device, and when “Predict Selected Music” is pressed, it will predict the genre. When the program predicts the genre, a bar chart displaying the similarity of the music among genres will be shown.



**Figure 3.** Kurdish music genre recognizer application.

#### 5. Conclusions

As music becomes more widely accessible in big collections, more emphasis is placed on identifying its genres. Consequently, music genre recognition (MGR) has evolved into a topic of research. Researchers have sought to use AI and machine learning to categorize musical genres. However, these methods and approaches have not been used up to this point for Kurdish music genres. We used machine learning techniques in this research to detect Kurdish music genres using two artificial neural network approaches: the DNN and CNN. We gathered and organized 880 pieces of music from 8 distinct Kurdish music genres. Each genre consisted of 110 pieces of music, each lasting 30 s. We created and assessed the models. The CNN model received an accuracy score of 92% and showed better performance than the DNN model, which had a score of 90% accuracy.

The Kurdish MGR is new, and therefore it is open to a wide range of studies. Some of the topics may be of immediate relevance, such as training the models on higher-quality music or working on improving the models to perform better on low-quality or poor-quality music as well because of the existence of a substantial amount of it in Kurdish music. Separating the songs from the music for each genre could be another area of focus that could improve genre detection. In addition, it is possible to examine training the models using longer samples, and this could improve the accuracy of the models. Last but not least, the categorized dataset utilized in this study may be used to develop a recommender system for Kurdish music genres.

**Author Contributions:** The authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** A selective subset of the dataset is available at <https://github.com/KurdishBLARK/KurdishMusicGenreRecognition> (accessed on 1 December 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lena, J.C.; Peterson, R.A. Classification as culture: Types and trajectories of music genres. *Am. Sociol. Rev.* **2008**, *73*, 697–718. [CrossRef]
2. Norowi, N.M.; Doraisamy, S.; Wirza, R. Factors affecting automatic genre classification: An investigation incorporating non-western musical forms. In Proceedings of the International Conference on Music Information Retrieval, London, UK, 11–15 September 2005; pp. 13–20.
3. Downie, J.S. Music information retrieval. *Annu. Rev. Inf. Sci. Technol.* **2003**, *37*, 295–340. [CrossRef]
4. Lorince, J. Consumption of Content on the Web: An Ecologically Inspired Perspective. Ph.D. Thesis, Indiana University, Bloomington, IN, USA, 2016.
5. Puppala, L.K.; Muvva, S.S.R.; Chinige, S.R.; Rajendran, P. A Novel Music Genre Classification Using Convolutional Neural Network. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 8–10 July 2021; pp. 1246–1249. [CrossRef]
6. Ghosal, D.; Kolekar, M.H. Music Genre Recognition Using Deep Neural Networks and Transfer Learning. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 2087–2091.
7. Feng, T. Deep Learning for Music Genre Classification. Available online: [https://courses.engr.illinois.edu/ece544na/fa2014/Tao\\_Feng.pdf](https://courses.engr.illinois.edu/ece544na/fa2014/Tao_Feng.pdf) (accessed on 3 February 2021).
8. Silla, C.N.; Koerich, A.L.; Kaestner, C.A. A machine learning approach to automatic music genre classification. *J. Braz. Comput. Soc.* **2008**, *14*, 7–18. [CrossRef]
9. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
10. Scheirer, E.; Slaney, M. Construction and evaluation of a robust multifeature speech/music discriminator. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 21–24 April 1997; IEEE: Piscataway, NJ, USA, 1997; Volume 2, pp. 1331–1334.
11. Ahmadi, S.; Hassani, H.; Abedi, K. A corpus of the Sorani Kurdish folkloric lyrics. In Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), Marseille, France, 11–16 May 2020; pp. 330–335.
12. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.P.; McVicar, M.; Battenberg, E.; Nieto, O. librosa: Audio and music signal analysis in python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 June 2015; Volume 8, pp. 18–25.
13. Velardo, V. Deep Learning for Audio with Python. 2021. Available online: <https://github.com/musikalkemist/DeepLearningForAudioWithPython> (accessed on 28 January 2021).
14. Taub, D. Must Accuracy Increase after Every Epoch? 2021. Available online: <https://stackoverflow.com/questions/45605003/must-accuracy-increase-after-every-epoch> (accessed on 29 June 2021).