mathematics

MDPI

Article

DBTMPE: Deep Bidirectional Transformers-Based Masked Predictive Encoder Approach for Music Genre Classification

Lvyang Qiu, Shuyu Li and Yunsick Sung *

Department of Multimedia Engineering, Dongguk University–Seoul, Seoul 04620, Korea; lvyangqiu@dongguk.edu (L.Q.); lishuyu@dongguk.edu (S.L.)
* Correspondence: sung@dongguk.edu

Abstract: Music is a type of time-series data. As the size of the data increases, it is a challenge to build robust music genre classification systems from massive amounts of music data. Robust systems require large amounts of labeled music data, which necessitates time- and labor-intensive data-labeling efforts and expert knowledge. This paper proposes a musical instrument digital interface (MIDI) preprocessing method, Pitch to Vector (Pitch2vec), and a deep bidirectional transformers-based masked predictive encoder (MPE) method for music genre classification. The MIDI files are considered as input. MIDI files are converted to the vector sequence by Pitch2vec before being input into the MPE. By unsupervised learning, the MPE based on deep bidirectional transformers is designed to extract bidirectional representations automatically, which are musicological insight. In contrast to other deep-learning models, such as recurrent neural network (RNN)-based models, the MPE method enables parallelization over time-steps, leading to faster training. To evaluate the performance of the proposed method, experiments were conducted on the Lakh MIDI music dataset. During MPE training, approximately 400,000 MIDI segments were utilized for the MPE, for which the recovery accuracy rate reached 97%. In the music genre classification task, the accuracy rate and other indicators of the proposed method were more than 94%. The experimental results indicate that the proposed method improves classification performance compared with state-of-the-art models.

Keywords: music genre classification; MIDI; transformer model; unsupervised learning

check for updates

Citation: Qiu, L.; Li, S.; Sung, Y. DBTMPE: Deep Bidirectional Transformers-Based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics* **2021**, *9*, 530. https://doi.org/10.3390/math9050530

Academic Editors: Duarte Valério and Élisa Fromont

Received: 29 December 2020
Accepted: 26 February 2021
Published: 3 March 2021

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

Copyright: © 2021 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

With the increasing industrial applications of music information retrieval, the large amounts of available music data and rich diversity of music genres pose a significant challenge in the effort to build a robust music genre classification system. However, there is no strict definition to distinguish music genres such that a single piece of music may be defined as different genres from different people. Additionally, high-quality musical instrument digital interface (MIDI) files with a genre label are insufficient for a robust music genre classification system. To reduce the aforementioned issues for saving the computation time of the model and the labor-intensive nature of the manual data-labeling task, it is worthwhile to explore how to extract features effectively from a huge amount of unlabeled data to improve the generalization and performance for the music genre classification systems.

In general, there are two main music formats. The different representations of music require different algorithms for processing the music. For the first format, which records audio intensity over time based on sound signals such as MP3 format, the majority of traditional music genre classification systems commonly follow two processes: feature engineering and model classification based on deep-learning technology. For feature engineering processes, the majority of the research has focused on extracting the acoustic properties of music data utilizing techniques such as short-time Fourier transform (STFT), spectrogram, and Mel-spectrogram, which can show that the energy distribution varies

with frequency [1]. Music spectrogram images are similar to Red-Green-Blue (RGB) images, and convolutional neural networks (CNN) [2] has achieved good performance in various research fields. Therefore, CNN is frequently applied to implement music genre classification tasks. Choi et al. proposed a convolutional recurrent neural network (CRNN) architecture utilizing a manual engineering feature as input, which involves a significant handcrafted design in the time–frequency transform [3]. However, these types of music genre classification techniques mainly rely on feature engineering.

Recently, to improve music classification performance, some researchers have attempted to extract music representation directly from raw waveforms. As a powerful and popular learning method, deep-learning technologies have been successfully introduced to computer vision [4], audio processing [5], natural language processing (NLP) [6], and so on. The main reason for these successes is that the algorithms based on deep learning can automatically extract advanced features and context representations from raw music data or processed data. Inspired by this research, raw-data-based music genre classification systems are presently being developed. Kim et al. applied the very deep CNN to the music genre classification task, utilizing raw music data as input [7]. However, although their deep CNN model can take context into account, it cannot avoid the substantial cost of calculation.

The second music format is a symbolic representation based on music scores. The common representations of this type are MIDI, Humdrum, MusicXML, MEI, and so on, where the pitch, duration, start time, and intensity of each note and chord are saved in file. Each of these elements can be analyzed to define the music genre individually or jointly, depending on the requirements of the model. MIDI are a standard music format used for music composition, which does not contain streaming media information and has the characteristics of a small file size and high sound quality. MIDI files can be converted into music scores. Compared with audio files, MIDI provides richer details on music elements, providing deep-learning models with insights into musicology. When music is composed using a computer, MIDI files are more controllable and convenient. Using deep-learning technology to extract features from MIDI files can provide MIDI users and authors a better understanding of music. In addition, research on music information retrieval is an extension of NLP. MIDI files can be regarded as a kind of string-type data, which provides the possibility of applying NLP technology to the music genres' classification.

As early as 2004, McKay and Fujinaga started to analyze the statistical distribution of global features extracted from MIDI files. These global features were applied to several different machine-learning models to perform classification [8]. As a result, they proposed the jMIR, an automatic music classification tool that includes feature extraction and machine-learning features [9]. In fact, it focuses on global statistical features and lacks musicological support in terms of feature extraction. Although McKay and Fujinaga obtained a promising result on a specified dataset, the classification accuracy relies heavily on global statistical features, such as the score of a particular instrument component. This approach contains only a few musicological features. Therefore, this model requires further improvements to deal with datasets having diverse features.

To consider more musicological features, combining Korean traditional musical instruments with original audio and MIDI phrases, Lee et al. utilized machine-learning algorithms such as support vector machine, multi-layer perceptron, decision tree, and random forest to implement music genre classification [10]. For a small number of MIDI files with labels, the model achieved good performance. However, it still cannot solve the problem wherein many unlabeled MIDI files cannot be applied to classification tasks. To apply unlabeled data to training, it is necessary to introduce unsupervised learning into the model. Cilibrasi et al. converted music into string-type data and used them as the input to classification tasks, which proves that it is feasible to apply an NLP-based method for music genre classification [11].

In this paper, a MIDI preprocessing method, Pitch to Vector (Pitch2vec), and a deep bidirectional transformers-based masked predictive encoder (MPE) method is proposed for

music genre classification using MIDI files as input. MIDI files are converted into a vector sequence by Pitch2vec before being input into the MPE. By unsupervised learning, the MPE based on deep bidirectional transformers is designed to automatically extract bidirectional representations, which are considered as musicological insights. The MPE can focus on long-term important relationships by predicting masked MIDI files. Even if the MIDI files are long, the proposed model can still consider the important relationship. Both CNN and RNN process long sequences by local encoding. However, CNN does not have the ability to capture time attributes, and RNN can only establish short-distance dependence because of the gradient vanishing problem. Further, although long short-term memory (LSTM) is more effective to address the RNN's gradient vanishing problem, it has not completely solved it. In contrast, because of using a huge among of data for unsupervised training and the robust model structure of the MPE, the proposed model has a generalization ability for music genre classification tasks.

To verify the performance of the proposed method, combined with the trained MPE, CNN was utilized to build a music genre classifier by supervised learning. This classifier can overcome the limitation of the small amount of labeled data and imbalance distribution dataset and also has the advantages of lower hardware requirements and faster data convergence. Thus, it facilitates rapid deployment of the trained neural network into real-world application software. Experiments were conducted on the Lakh MIDI [12] music dataset. The contributions of this paper are as follows:

1. Pitch2vec is proposed to convert MIDI files into pitch vectors. Therefore, language deep-learning models can be introduced to implement the music genre classification task.
2. A deep bidirectional transformers-based MPE is utilized to obtain bidirectional representations by recovering the masked MIDI files. In addition, many MIDI files without labels can be utilized well, which improves the generalization ability of the model.
3. A multi-level dynamic random masking operation is applied to enable the model to focus on multi-level contexts bidirectionally, which enables the model to understand the contexts well.
4. Combining with the trained MPE, a CNN-based classifier utilizing supervised learning is applied, which not only overcomes the limitation of unbalanced and less labeled data but also consumes less training resources.

The remainder of this paper is organized as follows. Section 2 outlines the related research on music classification based on deep-learning technology. Section 3 introduces the proposed method. Section 4 details the experiments conducted and analyzes the results obtained. Finally, Section 5 concludes this paper.

2. Related Works

In this section, the traditional deep-learning model utilized for music genre classification, which is based on manual features, is introduced. Then, the end-to-end model based on the original waveform is reviewed. In particular, the pre-training model based on the attention mechanism is given more focus, because the large number of unlabeled MIDI files need to be utilized, and MIDI files that are the discretized data are considered as the input data in the proposed method.

2.1. Music Genre Classification Based on Deep Learning

In the previous research of music genre classification, most CNN-based models take spectrograms that are transformed from music as inputs, such as Mel-spectrograms. Some researchers utilized different sub-optimal spectrogram settings according to a given domain in the music classification task. For example, 128 bins of Mel-spectrograms are a common choice. Song et al. proposed a music automatic tagging algorithm utilizing a deep recurrent neural network (RNN) with scattering transformed inputs. They noted that a stack of RNNs presumably benefited to reduce the phase variation in the time-domain convolution [13]. However, when features are extracted by transforming them into a

time-frequency graph, the time-context information of the music is lost. In addition, RNN is a type of time sequence model, which makes the model incapable of parallel computing. Yu et al. proposed a CNN-based parallelized attention model by taking STFT spectrograms as input to a bidirectional RNN [14]. This simple attention structure causes the model to be unable to recognize music genres composed of various melodies and instruments, which leads to the model having to rely on the CNN for classification. Therefore, to learn different features of music and utilize parallel computing, the self-attention mechanism is required that can maintain a long-term context and adjust the duration according to the context.

To avoid the problem of information loss in time-frequency transformation, some researchers have recently tried to utilize raw waveforms as an end-to-end method of input to CNNs. Dieleman and Schrauwen [15] applied the raw waveform to CNN for automatic music classification. When it was compared with the research with the Mel-spectrogram as input, although the context representation learned from the raw waveform seems meaningful, the performance of this method was not better than that of the model utilizing the Mel-filter banks. This can be attributed to the fact that the complexity of the raw waveforms makes the model difficult to classify the feature of the music.

In most previous research, large-size filters were employed in the first convolutional layer, and hundreds of samples were collected at a time (or at the frame level). However, in this case, the filter must learn all possible phase changes within the time-domain filter size. To train a CNN properly with raw waveforms, a deep network [16] and a well-designed network is required to suppress phase changes based on a large amount of labeled music data [17].

Lee et al. recently proposed a different type of neural network that takes the raw waveform as input and has a very small filter, which was called SampleCNN [18,19]. They revealed that in automatic music classification, CNN with raw waveforms as input can perform better when the first convolutional layer takes a small size kernel (for example, a size of 2 or 3) instead of the typical frame-level size kernel (for example, a size of 256 or 512). Pons et al. also demonstrated that this model is effective for the automatic labeling of music on a certain scale [20]. When experiments were conducted on an industrial scale, for which the number of songs was sufficiently large (for example, more than 1 million songs), it was shown that the SampleCNN model can outperform the spectrogram-based model. Kim et al. utilized down-sampling combined with more advanced convolutional building blocks to further improve the SampleCNN for automatic music labeling [7]. They found that when the network layer is sufficiently deep, CNN utilizing raw waveform input can compete with CNN utilizing Mel-spectrogram input. However, this inevitably causes a significant increase in the expenditure on computing resources. These models, which are only trained for classification tasks, only focus on music features that are useful for classification and ignore other features that constitute the music.

Owing to the rapid development of digital media technology, a large amount of unlabeled music data has appeared, and applying these data in the field of music information retrieval has become a new challenge. For example, in 2018, the successful application of pre-training models in the field of NLP proved that the models could learn potential semantic information from a large amount of unlabeled text, and only require a lower amount of individually labeled training data to perform each downstream NLP task. These models include RNN-based Embeddings from Language Models (ELMo) [21], Universal Language Model Fine-Tuning for Text Classification (ULMFiT) [22], transformer [23]-based Generative Pre-Training (GPT) [24] of OpenAI, and Bidirectional Encoder Representation from Transformers (BERT) [25] of Google. BERT is the first technique in which a mask language model (MLM) based on the deep transformer structure was utilized, where the transformer is based on a self-attention mechanism. MLM reconstructs the masked input during the pre-training phase. The success of pre-training language models has created a new paradigm for NLP research [26], that is, the first one may utilize a large amount of unlabeled corpus for the pre-training language model and then utilize a small amount of labeled corpus for fine-tuning to implement given NLP tasks, such as classification,

sequence labeling, evaluation of the relationship between sentences, and machine reading comprehension. As both audio and music are sequence data with time attributes, they have a certain similarity with text data. In tasks such as speaker recognition and phoneme classification, inspired by MLM, Mockingjay [27] masked the input acoustic features and attempted to reconstruct the corresponding linear spectrogram or Mel-spectrogram in the pre-training phase. Similarly, the theory of MLM has been utilized [28] to pre-train the speech recognition model, which was based on unsupervised pre-training, to extract representations containing bidirectional context. Vq-wav2vec [29] combined vector quantization and BERT to improve the performance of downstream tasks. The application of the pre-trained model based on unsupervised learning not only overcomes the lack of labeled data in the given problem but also significantly enhances the model's ability to extract and understand features. In the pre-training process, the model can learn bidirectional representations in music. For example, the features utilized to generate music [30] and predict the next phase of music can play an auxiliary role in the model to understand music data and assist music classification tasks. However, there are still some problems with the BERT pre-training models. BERT's pre-training task MLM makes the model able to encode sequences with context, but at the same time, it also causes the data in the pre-training phase and the data in the fine-tuning phase not to match, because MASK does not appear in the fine-tuning data. In addition, BERT cannot consider the correlation between predictions MASK. Finally, owing to the limitation of the maximum input length, it is suitable for sentence and paragraph level tasks, but it is not suitable for long sequence tasks. These issues prompted us to build an improved model based on unsupervised learning.

2.2. Comparison of Music Genre Classification Based on Deep Learning

Table 1 shows the difference between previous music genre classification methods and the proposed method. There are two main advantages of the proposed method. First, in the Pitch2vec preprocessing, multiple tracks in MIDI files are embedded into pitch vectors based on symbolic representation. Second, compared with previous research, most of them utilize a small-scale dataset for training, which led to the problem of weak model generalization ability. To obtain a bidirectional context representation, which is the basis of the robustness of the model, the proposed method first utilizes a large amount of unlabeled MIDI files to train the deep bidirectional transformers-based model based on unsupervised learning. Then, labeled MIDI files are utilized to perform downstream music genre classification tasks based on supervised learning. To verify the performance of the proposed method, combined with the trained MPE, CNN is utilized to build the music genre classifier by supervised learning. This method requires fewer computing resources than previous research. Therefore, the limitations of previous research, such as insufficient data and imbalanced labeled data, are solved, and the efficiency of industrial application deployment is improved.

Table 1. Difference between previous research and the proposed method.

Research Contents	CRNN [1]	Sample-CNN [6]	MIDI Classification by Machine Learning [7]	Deep RNN [10]	CNN-Based Attention Model [11]	The Proposed Method
Input Feature	Mel-spectrogram	Raw Waveform	MIDI to String	Scattering Transformed	STFT Spectrograms	Pitch2vec Preprocessing
Neural Network	CRNN	Very Deep CNN	Normalized Compression Distance	Deep GRU	Bidirectional RNN based on CNN attention	Deep Bidirectional Transformers-Based MPE

3. MPE Based on Deep Bidirectional Transformers

In this section, the main motivation of the music genre classification system is introduced, and the details of the MPE based on unsupervised learning are presented with two additional processing stages: preprocessing and reconstruction decoding processing.

3.1. Overview

The MPE method consists of three parts: In the Pitch2vec preprocessing, according to the index, the pitches in the MIDI files are converted to the vector sequence. In the reconstruction decoding processing, the reconstruction decoder is built by the fully connect layer to reconstruct the masked MIDI files. During MPE training, the MPE utilizes deep bidirectional transformers to reconstruct masked MIDI, which can extract the bidirectional representations. To solve the problem of data mismatch between MPE training and classification, the multi-level dynamic random masked operation is utilized, which takes into account the characteristics of MIDI. To verify the performance of the proposed method, the music genre classifier was defined, the reconstructing masked MIDI decoder was removed, and CNN was inserted for music genre classification tasks. The CNN-based classifier is combined with the trained MPE, which is based on supervised learning, to achieve high accuracy and robust generalization ability in music genre classification tasks.

3.2. Pitch2vec Preprocessing for MPE and Reconstruction Decoding Processing

During Pitch2vec preprocessing, there are multiple asynchronous tracks in one MIDI file. These tracks play different roles when playing MIDI files, including one melody track, multiple chord tracks, and one drum track. To standardize the number of MIDI tracks, multiple chord tracks are synthesized into one MIDI track. Three main tracks including a melody track, synthesized chord track, and drum track are regarded as the main feature of MIDI files. As shown in Figure 1, through MIDI Track Extractor, the three main tracks of MIDI files are extracted. Second, as shown in Figure 2, according to the duration information corresponding to each set of pitches, Pitch2vec Converter utilizes the pitch information and the pitch corresponding index to construct pitch vectors. Specifically, each pitch and its corresponding duration should be converted into an index value, which is the input of the deep neural network. During MPE training, the deep bidirectional transformers can understand the relationship between different pitch vectors.

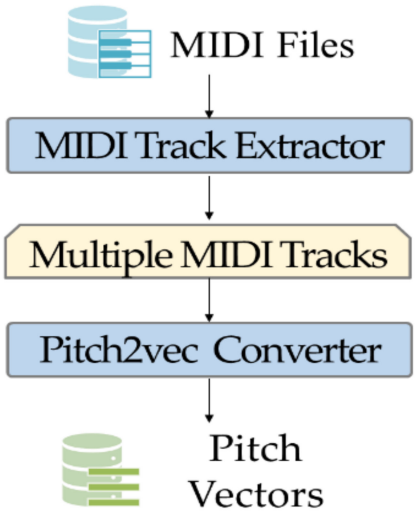


Figure 1. Pitch2vec preprocessing for the MPE.

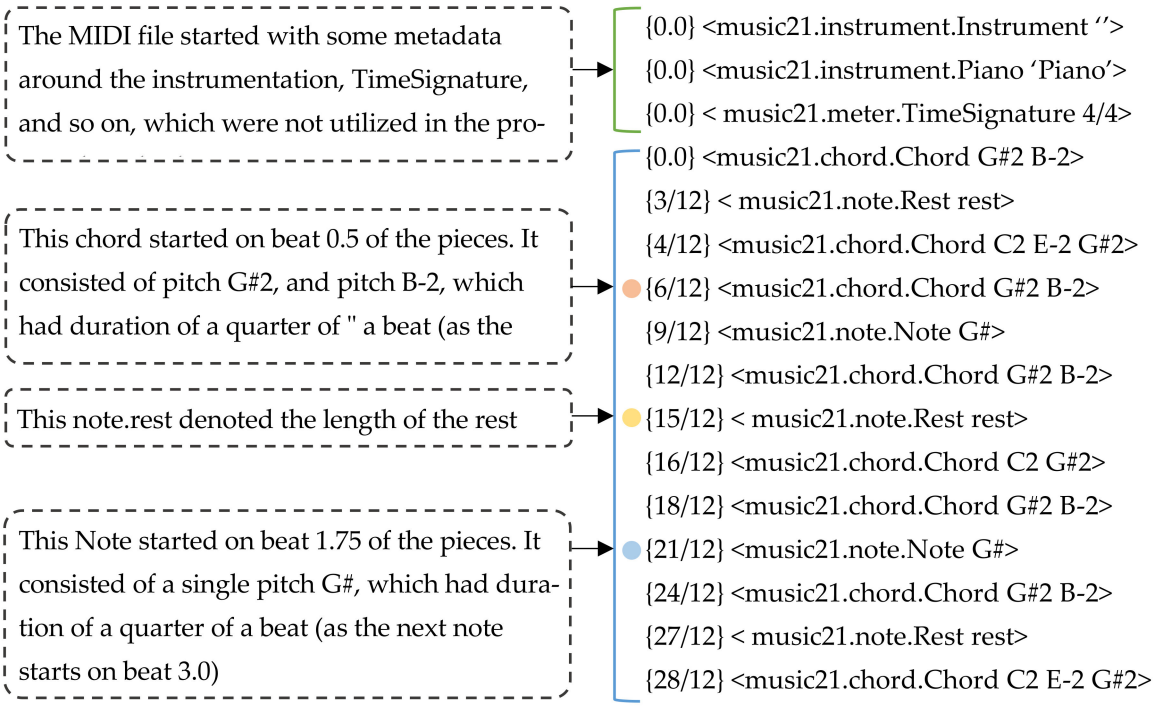


Figure 2. Symbolic musical information of MIDI.

Each pitch and its corresponding duration should be converted into an integer value. By traversing all the MIDI files, all the pitch combinations were found, and they were filtered according to the frequency of the pitch combinations. Finally, 30,000 high-frequency pitch combinations were obtained and numbered in the order of pitch index as shown in Table 2. In this way, similar pitch combinations were numbered to adjacent positions, which made the relationship between pitch combinations closer. To construct the integer duration value, all the durations were increased by 12 times to convert all floats to integers, because samples were taken 12 times per second. Pitches in each MIDI segment could be transformed into pitch vectors.

Table 2. The pitch-index lookup dictionaries for pitch.

Pitch	Index
[PAD]	0
A–1	1
A–1.A0	2
A0	3
A0.A0	4
A0.A1	5
A0.A1.A2	6
A0.A1.C2.E2	7

In the reconstruction decoding processing, as shown in Figure 3, the reconstruction decoder, which follows the MPE, is defined by the fully connected layer to assist the MPE in the training process by reconstructing the masked MIDI files. The input of reconstruction decoder is hidden states of the MPE. The output is reconstructed pitch vectors. After assisting the MPE in training, the reconstruction decoder was replaced by a CNN-based classifier for the music genre classification task.

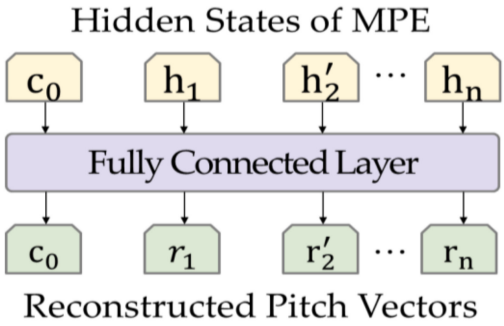


Figure 3. Reconstruction decoding processing follow by MPE. c denotes the classification token. r'_n denotes the reconstruction vector predicted by the model.

3.3. MPE Based on Deep Bidirectional Transformers

The MPE is a deep bidirectional transformers-based model with unsupervised learning, where the transformer is mainly based on the self-attention mechanism. Because of the powerful unsupervised learning technique, the MPE can extract bidirectional representations, which contain all the useful features of MIDI. The structure of the MPE is shown in Figure 4.

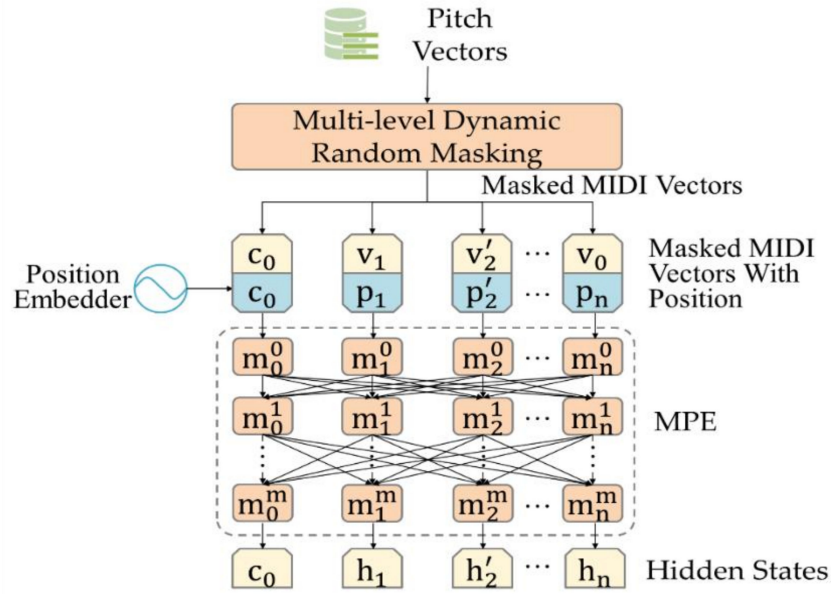


Figure 4. Structure of the MPE. c denotes the classification token that is inputted to CNN. v'_n denotes the masked vector MASK. p'_n denotes the position vector. m_n^m denotes the self-attention transformer model.

First, the input of the MPE comes from the Pitch2vec preprocessing in the form of pitch vectors. After multi-level dynamic random masking operations, these vectors are fed into the trained MPE to calculate a special token, which is called the classification token and is represented as c. c is added in front of every MIDI segment to store the context representations related to music genre classification. h, which implies a hidden state, is also outputted by the MPE, which is utilized to achieve reconstruction. Then, to define the task based on unsupervised learning, the MPE reconstructs the masked data through the transformers-based model to obtain deep bidirectional representations. To force the MPE to consider the characteristics of MIDI files, a multi-level dynamic random masking strategy to mask the training data is proposed. Dynamic masking is adopted, where the masking pattern generated in every epoch is fed to the model, which makes the position of each epoch different. For the purposes of multi-level masking strategy, a

note is the smallest unit for defining music, which can be considered as alphabets in NLP tasks, the note-level random masking strategy is defined. Further, because musical bars can be considered as words in NLP tasks, the bar-level random masking strategy is defined. Therefore, this masking strategy is designed to solve problems in which BERT cannot consider the correlation between predictions MASK. During each training epoch, 15% of the training data are selected for multi-level dynamic random masking operations, which results in a different masking position. Therefore, masking operations can be considered as a dynamic random label, which reduces the degree of mismatch between the MPE training data and the classification data. Specifically, 80% of the selected data are replaced with zero, 10% with random data, and the remaining data are kept unchanged. The masked is represented as v'_n in the figure. The MPE predicts each masked frame based on the data appearing before and after the masked frame. Therefore, the MPE can consider the bidirectional context of the data.

Second, to enable the MPE to consider the relationship of the input sequence order, a position embedder is defined, because transformer encoders consider the weight relationship between every input, while ignoring the input sequence order relationship. The position vectors, which are represented as p_n, are concatenated with pitch vectors before being input into the MPE, which is a frequency sinusoid. Sinusoidal position embedding facilitates the use of relative or absolute position information to extend the model. Location information is important in NLP tasks. The same content can have different meanings in different locations. For the MPE structure, the deep bidirectional transformer encoder is based on the self-attention mechanism that is selected as the basic unit of the model. Each transformer encoder is represented as m_n^m. When compared with the RNN, the self-attention mechanism can help the model to assign different weights to each part of the input in parallel computing and obtain more details and important information, such that the model can make more accurate judgments. Meanwhile, the self-attention mechanism does not add to the computational and storage burdens of the model, because compared with CNN and RNN, the self-attention model has less complexity and fewer parameters. In addition, the deep model benefits by gradually understanding the various linear transformation combinations in the data from low-level features to high-level features. For the shallow transformer encoder layers, they can consider the context between notes. For the deep transformer encoder layers, they can consider the context of the entire sequence.

Finally, to assist the MPE in training, the reconstructing masked MIDI decoder is built by the fully connected layer as shown in Figure 3. The reconstructed parts are represented as r'_n. The reconstructed MIDI tracks are multiplied by the embedding layer and converted into the input dimension. The processed output is compared with the original input sequence to obtain the loss value as the sum of $loss_n$ calculated from note-level random masking and $loss_b$ calculated from bar-level random masking. The parameters of the MPE are then updated based on the backpropagation algorithm. Utilizing a large batch size enables the acquisition of a more precise gradient direction for backpropagation so that the optimization rate and performance of the model can be improved. To achieve this effect, gradient accumulation is performed to accumulate continuously the gradient after each batch calculation. Following accumulation for a certain number of times, the network parameters are updated according to the accumulated gradient. Then. the gradient is cleared, and the next gradient accumulation optimization loop is performed.

After the unsupervised training of MPE, the performance of the proposed MPE was verified by defining the music genre classifier, removing the reconstructing masked MIDI decoder, and inserting the CNN following the trained MPE, to implement music genre classification tasks.

4. Experiment

In this section, the objectives of the experiment are described. The detailed hyperparameters for the MPE training and experimental results of the proposed method were

shown. Besides, the result of the proposed method was compared with previous music genre classification research results.

4.1. Experimental Objectives

Experiments were conducted to verify whether downstream classification tasks could obtain performance well when the MIDI features extracted from the MPE were applied.

4.2. Experimental Environment

The proposed model had two main training processes. During MPE training, first, the max length allowing input length of pitch vectors was 4000, which was considered as the music with a duration of about 5 min. Then, input was divided into multiple subsequences with a length of 128, according to the sequence length. After the multi-level dynamic masking operation, the input sequences were fed to the MPE model. Specifically, as shown in Table 3 of the MPE training parameters, the batch size was set as 1024. The learning rate, which was affected by warmup steps, gradually increased to 0.00176 after 3125 steps. Warmup operation could adjust better initialization parameters for the model. In order to address the problem of memory limitation, the gradient accumulation optimizer was utilized. Through gradient caching and accumulation, training time was utilized to offset memory limitation, so that the training effect could be equivalent to the model with a large training batch size. Considering the influence of the gradient accumulation optimizer, the actual total training steps were equal to total training steps multiplied by gradient accumulation steps. Epochs could be calculated by actual total training steps divided by steps per epoch. The optimizer was chosen as Adam, which was a method for stochastic optimization. Finally, the output of the MPE was the reconstructed data, so the output size was equal to the input size.

Table 3. Parameters during MPE training.

Hyper-Parameter	Value
Max length	4000
Sequence length	128
Input size	(Batch size, sequence length)
Batch size	1024
Learning rate	0.00176
Warmup steps	3125
Total training steps	10,000
Steps per epoch	10,000
Gradient accumulation steps	16
Optimizer	Adam
Output size	(Batch size, sequence length)

To verify the performance of the proposed method, the music genre classifier was defined. Owing to the excellent performance of CNNs in various classification tasks, CNN was employed to construct the classifier in this paper. Combining with the trained MPE, a CNN-based classifier utilizing supervised learning was applied. The CNN-based classifier was a module designed to combine the trained MPE to perform downstream classification tasks, which was based supervised learning. Because the structure of deep transformer-based encoders and bidirectional representations were extracted during reconstructing the masked content, these representations with robust generalization ability allowed the MPE to be applied to various downstream tasks. Therefore, the MPE was applied to the music genre classification task. The MIDI files with labels were fed into the model that consisted of the trained MPE and the CNN-based classifier. First, after the MIDI files were processed by the MIDI splitter, MIDI processor, the vector with position coding was obtained as the neural network input. This vector was fed into the trained MPE to calculate a C token. Then, the CNN structure with the C token could predict the music genre label. After comparing the predicted label with the real label, the loss value was calculated to update

the global parameters in the combined model of the trained MPE and the CNN-based classifier. Comparing with the trained MPE, the CNN-based classifier only needed to adjust fewer parameters. The model can converge quickly, which also means that the CNN-based classifier consumes relatively fewer training resources. In addition, due to MPE considering a lot of features from unlabeled MIDI files, the CNN-based classifier was a model with strong generalization ability.

During performing CNN-based classifier to music genre classification, pitch vectors were divided into multiple sub-sequences with a length of 128 according to max length. The CNN-based classifier held these sub-sequences for training. Input size could be (batch size, max length). Specifically, the training parameters of the CNN-based classifier were shown in Table 4. Due to adjusting only a few parameters, the learning rate was set to a relatively small value that was 5×10^{-6}. The batch size was set as 160. The optimizer was chosen as Root Mean Square Propagation (RMSprop). According to the total MIDI genre that was 13, output size was (batch size, 13).

Table 4. Parameters of CNN-based classifier for music genre classification.

Hyper-Parameter	Value
Max length	128
Input size	(Batch size, max length)
Learning rate	5×10^{-6}
Batch size	160
Optimizer	RMSprop
Output size	(Batch size, 13)

The experiments were performed utilizing Windows 10, i7–7700, NVIDIA TITAN RTX 24 GB, and DDR4 40 GB. The proposed method was developed with Python. The MPE and CNN-based classifier were implemented with tensorflow and keras, which is a deep-learning library. MIDI files were processed by the Music 21 library [31], which not only contains the standard feature extraction tools offered by other toolkits but also allows researchers to customize powerful feature extraction methods.

4.3. Experimental Data

The music genre classification data based on MIDI files lack a proven and reliable dataset. The Lakh MIDI dataset is one of the relatively reliable datasets. This dataset was usually utilized for music genre classification. However, this dataset still has shortcomings such as incomplete label, noise data, and so on. To overcome these shortcomings, the unsupervised MPE can flexibly use the Lakh MIDI dataset as training data. Lakh MIDI dataset [12] is a collection with 176,581 MIDI files. This dataset was matched and aligned to entries in the Million Song Dataset. Among the Lakh MIDI dataset, 11,946 MIDI files have genre labels, which includes 13 categories, such as Pop/Rock, Electronic, Country R and B, Jazz, Latin, International, and so on.

Unlabeled data were utilized during MPE training; the three mains tracks were extracted from each MIDI file and divided into 10s so that we obtained about 400,000 MIDI tracks as the input data of the MPE. To obtain a reliable and robust model and to avoid training dataset including validation and test data in the training dataset, the dataset was split based on the theory of cross-validation. Ninety percent of this data was utilized for the training dataset, and the remaining 10% was divided equally between the validation and test dataset. It was worth noting that MIDI segments from the same MIDI file are not allowed to appear in different partitions of dataset.

Labeled data were utilized in the CNN-based music genre classification. The samples of each MIDI genre distribution were shown in Table 5. Because the data distribution was very unbalanced, data oversampling [32] was performed on the labeled dataset in order to ensure that all genres are included in each training batch. Specifically, oversampling is to increase the amount by replicating samples. The division ratio refers to the strategy in

the reference [3]. The replicated samples only appear in the same partition of the dataset to avoid overfitting. The samples for each MIDI genre distribution are shown in Table 5. The three main tracks were then extracted from each MIDI file and divided into 10s. A total of 178,377 MIDI segments were obtained. Eighty-five percent of MIDI segments, 151,586 MIDI segments, were considered as the training dataset. Five percent of MIDI segments, 8918 MIDI segments, were considered as the validation dataset. Ten percent of MIDI segments, 17,837 MIDI segments, were considered as the test dataset.

Table 5. The distribution of MIDI files with genre label in the Lakh MIDI Dataset.

Genre	Number of MIDI Files	Number of MIDI Files without Oversampling	Number of MIDI Files with Oversampling	Number of MIDI Segments
Pop/Rock	8603	8603	-	110,875
Country	962	962	-	13,364
Electronic	694	694	-	7553
R&B	475	-	950	8690
Latin	293	-	586	6722
Jazz	280	-	560	6856
New Age	230	-	460	4586
Rap	117	-	585	3560
International	86	-	430	4875
Reggae	69	-	345	3345
Folk	64	-	320	3175
Vocal	41	-	246	2634
Blues	32	-	192	2142
Total	11,946	10,259	4674	178,377

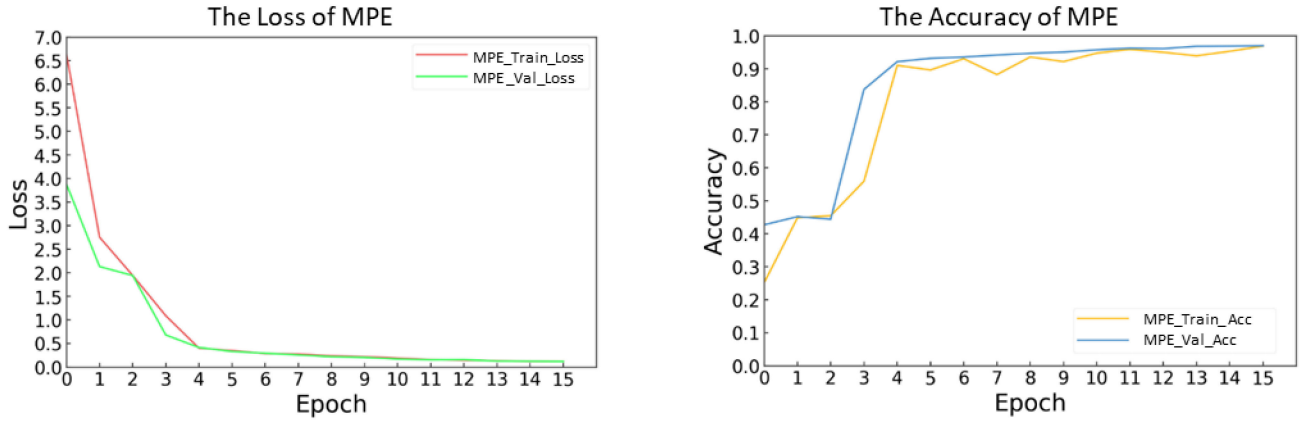
4.4. Experimental Results

The results of the Pitch2vec preprocessing were introduced in this part, as shown in Table 6. First, the basic information of music was utilized. Secondly, the duration information corresponding to each set of pitch and rest in MIDI was utilized to construct pitch vectors.

Table 6. The results of the Pitch2vec preprocessing. [–] denoted the rest of MIDI. ['*.* . . . '] denoted the set of MIDI pitch.

Duration	Original Pitch	Pitch to Vector
0	[–]	0
1/12	[–]	0
2/12	['G4.G5']	29,833
3/12	['G4.G5']	29,833
4/12	['G4.G5']	29,833
5/12	['G4.G5']	29,833
6/12	['G4.G5']	29,833
7/12	['G3.G4']	29,045
8/12	['G3.G4']	29,045
9/12	['G3.G4']	29,045
10/12	['F4.F5']	24,768

Figure 5 showed the training result of the MPE. As shown in Figure 5a, training the MPE took 16 epochs. For training loss of the MPE, the initial value was about 6.62. After four epochs, it started to converge. Finally, it converged to about 0.12. For validation loss of the MPE, the initial value was about 3.95. After four epochs, it started to converge. Finally, it converged to about 0.12.

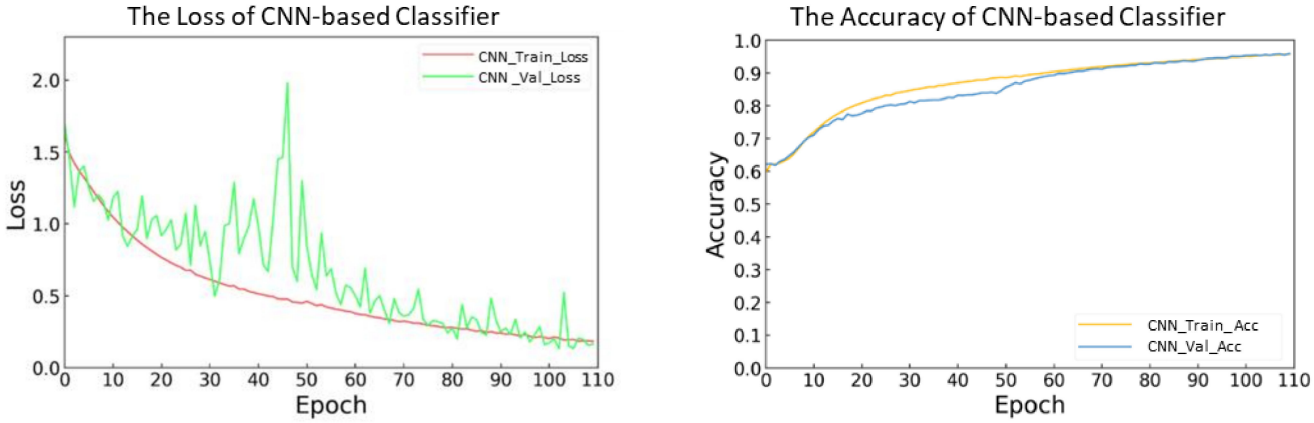


(**a**) (**b**)

Figure 5. Masked predictive encoder (MPE): (**a**) The loss of MPE. (**b**) The accuracy of MPE.

As shown in Figure 5b, for the training accuracy of the MPE, the initial value was about 42%. After four epochs, it started to converge. The final accuracy rate reached about 97%. For the validation accuracy of the MPE, the initial value was 26%. After four epochs, it started to converge. Finally, it converged to about 97%. This figure shows that the MPE could extract the features of music and complete the reconstruction operation well.

Figure 6 shows the music genre classification result of the CNN-based classifier. As shown in Figure 6a, training the CNN-based classifier took 110 epochs. For the training loss of the CNN-based classifier, the initial value was about 1.65. After twenty epochs, it started to converge. Finally, it converged to about 0.2. For the validation loss of the CNN-based classifier, the initial value was 1.70. Overall, the validation loss curve fluctuated sharply, but it was in a downward trend. Finally, it converged to about 0.22.



(**a**) (**b**)

Figure 6. Convolutional neural networks (CNN): (**a**) The loss of CNN-based classifier. (**b**) The accuracy of CNN-based classifier.

As shown in Figure 6b, for the training accuracy of the CNN-based classifier, the initial value was about 59%. After the 20th epoch, it started to converge. The final accuracy rate reached about 95%. For the validation accuracy of the CNN-based classifier. The initial value was about 62%. After twenty epochs, it started to converge. The final accuracy rate reached about 95% and remained stable. This figure shows that combining with the trained MPE, the CNN-based classifier could quickly converge and complete the training, which reduced the deployment time and cost.

Table 7 shows the performance of the CNN-based classifier in the test dataset. The values of accuracy, precision, recall, and f1-score were utilized to evaluate model performance when the distribution of the dataset was imbalanced. Additionally, all values reached a value of more than 94%. This shows that the prediction results of the model were reliable.

Table 7. CNN-based classifier's test result of evaluation indicators for music genre classification.

Evaluation Indicators	Results
Accuracy	0.9427
Precision	0.9429
Recall	0.9422
F1-score	0.9422

Figure 7 shows the confusion matrix that classified MIDI files of 13 genres. An total of 17,837 test MIDI segments came from 1471 MIDI files. By comprehensively considering the different prediction results of different segments in the same MIDI file, it was possible to avoid the prediction error of some MIDI segments leading to the wrong judgment of the MIDI file's genre. The blue color means the numbers of accurate classification.

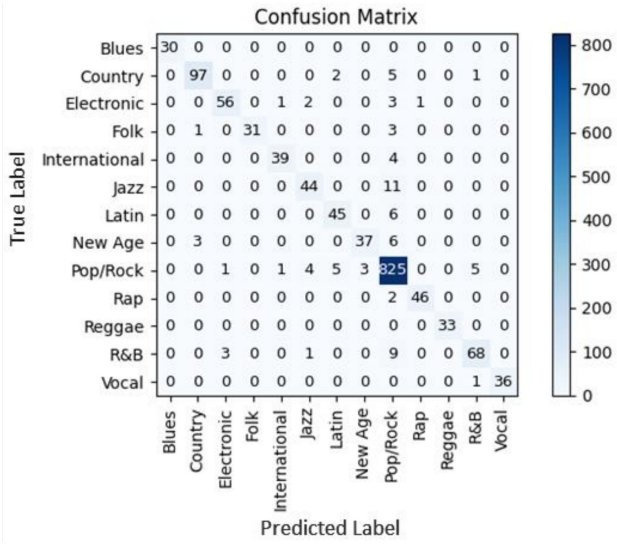


Figure 7. Test result of the CNN-based classifier showed by confusion matrix.

Table 8 shows the comparison between the proposed method and previous research [33,34]. Three evaluation criteria were considered, which include the classification model's variance of the proposed method, the receiver operating characteristic and area under curve (ROC-AUC), and classification accuracy of the proposed method and previous research. The variance of the classification model is represented by the deviation between predicted labels and true labels. ROC-AUC is commonly utilized to evaluate imbalance distribution dataset. Each ROC-AUC was calculated by one genre, and all scores were averaged and represented by one score.

Table 8. Comparison between the proposed method and previous research for music genre classification.

Model	Variance	ROC-AUC	Accuracy
Two LSTM layer [34]	–	–	0.6488
P2 pattern query algorithm [34]	–	0.8160	0.6410
The proposed method	0.22	0.9514	0.9427

In this paper, two examples of music genre classification research [33,34] based on the Lakh MIDI dataset were selected as comparative research. In [33], Bollar et al. utilized two LSTM layers to classify MIDI genres, including five genres, which obtained the accuracy of [33], 0.6488. In [34], Ferraro et al. utilized the P2 pattern query algorithm to classify MIDI genres, including 13 genres; 0.8160 as ROC-AUC of [34] and 0.6410 as the accuracy of [34] were obtained. After the proposed method was converged in training, the variance of the proposed method was 0.22 and remained stable. During performing the proposed method by a test dataset, ROC-AUC was 0.9514, and the accuracy of the proposed method was 0.9427. The experimental results showed that the proposed classification model has improved performance compared with the previous research.

5. Conclusions

In this paper, a MIDI preprocessing method, Pitch to Vector (Pitch2vec), and an MPE method based on deep bidirectional transformers for music genre classification were proposed using MIDI files as input. MIDI files were first converted to pitch vectors according to the pitch index. Therefore, language deep-learning models could be introduced to implement the music genre classification task. Second, the unsupervised MPE was utilized to extract the deep bidirectional context representation from a large number of MIDI files. On the one hand, the multi-level dynamic random masking operation was applied, which enabled the model to focus on multi-level contexts. On the other hand, unlabeled MIDI files could also be utilized to improve the generalization of the model by deep bidirectional transformer-based MPE. Third, to verify the performance of the proposed method, the music genre classifier was defined. Through supervised learning, the CNN-based classifier was combined with the trained MPE to implement the music genre classification task. Although with unbalanced labeled data or a small amount of data, the performance of the model was well when the training and classifying time was saved. Experimental results showed that the classification performance could be boosted by the proposed method compared with the state-of-the-art models. The accuracy rate and other indicators were found to be more than 94%.

Compared with current state-of-the-art music genre classification systems, the proposed method provided a novel solution with strong generalization ability, which combined with the unsupervised method based on a large amount of music data for representation learning and utilized CNN to implement the music genre classification task. Therefore, it is easier to be applied practically for industrial applications. In the future, based on the context that MPE only extracted within a music segment, the context relationship among music segments needs to be considered so that the reconstruction ability of MPE can be further improved. In addition, considering the influence of ontological ambiguity and multiple labels on the classification model, all tracks of MIDI files are encoded into vectors as the input of the classification model, which can obtain a more useful context to support the multi-labels music genre classification task.

Author Contributions: Conceptualization, L.Q., S.L., and Y.S.; methodology, L.Q., S.L., and Y.S.; software, L.Q., S.L., and Y.S.; validation, L.Q., S.L., and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science, ICT, Korea, under the High-Potential Individuals Global Training Program (MSIT) (2019-0-01585, 2020-0-01576) supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from Colin Raffel and are available https://colinraffel.com/projects/lmd/ (accessed on 1 December 2020) with the permission of Colin Raffel.

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

References

1. Nam, J.; Choi, K.; Lee, J.; Chou, S.Y.; Yang, Y.H. Deep Learning for Audio-Based Music Classification and Tagging: Teaching Computers to Distinguish Rock from Bach. *IEEE Signal Process. Mag.* **2019**, *36*, 41–51. [CrossRef]
2. Jang, S.; Li, S.; Sung, Y. FastText-based Local Feature Visualization Algorithm for Merged Image-based Malware Classification Framework for Cyber Security and Cyber defense. *Mathematics* **2020**, *8*, 460. [CrossRef]
3. Choi, K.; Fazekas, G.; Sandler, M.; Cho, K. Convolutional recurrent neural networks for music classification. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2392–2396.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Chuang, Y.S.; Liu, C.L.; Lee, H.Y. SpeechBERT: Cross-modal pre-trained language model for end-to-end spoken question answering. *arXiv* **2019**, arXiv:1910.11559.
6. Kim, E.; Jang, S.; Li, S.; Sung, Y. Newspaper article-based agent control in smart city simulations. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 1–19. [CrossRef]
7. Kim, T.; Lee, J.; Nam, J. Comparison and Analysis of SampleCNN Architectures for Audio Classification. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 285–297. [CrossRef]
8. McKay, C.; Fujinaga, I. Automatic Genre Classification Using Large High-Level Musical Feature Sets. In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain, 10–14 October 2004; pp. 525–530.
9. McKay, C.; Fujinaga, I. jSymbolic: A Feature Extractor for MIDI Files. In Proceedings of the 21st International Cryogenic Engineering Conference and International Cryogenic Material Conference (ICMC), Prague, Czechia, 17–21 July 2006.
10. Lee, J.; Lee, M.; Jang, D.; Yoon, K. Korean Traditional Music Genre Classification Using Sample and MIDI Phrases. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 1869–1886. [CrossRef]
11. Cilibrasi, R.; Vitányi, P.; De Wolf, R. Algorithmic Clustering of Music Based on String Compression. *Comput. Music. J.* **2004**, *28*, 49–67. [CrossRef]
12. The Lakh MIDI Dataset. Available online: https://colinraffel.com/projects/lmd (accessed on 1 December 2020).
13. Song, G.; Wang, Z.; Han, F.; Ding, S.; Iqbal, M.A. Music auto-tagging using deep Recurrent Neural Networks. *Neurocomputing* **2018**, *292*, 104–110. [CrossRef]
14. Yu, Y.; Luo, S.; Liu, S.; Qiao, H.; Liu, Y.; Feng, L. Deep attention based music genre classification. *Neurocomputing* **2020**, *372*, 84–91. [CrossRef]
15. Dieleman, S.; Schrauwen, B. End-to-end learning for music audio. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 6964–6968.
16. Dai, W.; Dai, C.; Qu, S.; Li, J.; Das, S. Very deep convolutional neural networks for raw waveforms. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 421–425.
17. Sainath, T.N.; Weiss, R.J.; Senior, A.; Wilson, K.W.; Vinyals, O. Learning the speech front-end with raw waveform CLDNNs. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
18. Lee, J.; Park, J.; Kim, K.L.; Nam, J. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv* **2017**, arXiv:1703.01789.
19. Lee, J.; Park, J.; Kim, K.L.; Nam, J. SampleCNN: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification. *Appl. Sci.* **2018**, *8*, 150. [CrossRef]
20. Pons Puig, J.; Nieto Caballero, O.; Prockup, M.; Schmidt, E.M.; Ehmann, A.F.; Serra, X. End-to-end learning for music audio tagging at scale. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018), Paris, France, 23–27 September 2018; pp. 637–644.
21. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representa-tions. *arXiv* **2018**, arXiv:1802.05365.
22. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 May 2018.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
24. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; Technical Report; OpenAI: San Francisco, CA, USA, 2018.
25. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language under-standing. *arXiv* **2018**, arXiv:1810.04805.
26. Zhou, M. The Bright Future of ACL/NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 29 July 2019.
27. Liu, A.T.; Yang, S.W.; Chi, P.H.; Hsu, P.C.; Lee, H.Y. Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6419–6423.

28. Jiang, D.; Lei, X.; Li, W.; Luo, N.; Hu, Y.; Zou, W.; Li, X. Improving transformer-based speech recognition using unsupervised pre-training. *arXiv* **2019**, arXiv:1910.09932.
29. Baevski, A.; Schneider, S.; Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv* **2019**, arXiv:1910.05453.
30. Li, S.; Jang, S.; Sung, Y. Automatic Melody Composition Using Enhanced GAN. *Mathematics* **2019**, *7*, 883. [CrossRef]
31. Cuthbert, M.S.; Ariza, C.; Friedland, L. Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR), Miami, Florida, 24–28 October 2011; pp. 387–392.
32. Valerio, V.D.; Pereira, R.M.; Costa, Y.M.; Bertoini, D.; Silla, C.N., Jr. A resampling approach for imbalanceness on music genre classification using spectrograms. In Proceedings of the Thirty-First International Flairs Conference, Melbourne, FL, USA, 21–23 May 2018.
33. Bollar, H.; Misra, S.; Shelby, T. *Music Genre Classification Using Mid-Level Features*; IEEE: New York, NY, USA, 2002. Available online: https://www.hannahbollar.com/files/compProjs/musicGenreClassification.pdf (accessed on 10 February 2021).
34. Ferraro, A.; Lemström, K. On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns. In Proceedings of the 5th International Conference on Digital Libraries for Musicology, Paris, France, 28 September 2018; pp. 34–37.