applied sciences

MDPI

Article

Deformer: Denoising Transformer for Improved Audio Music Genre Classification

Jigang Wang [1], Shuyu Li [1] and Yunsick Sung [2,*]

[1] Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 04620, Republic of Korea; 2021120330@dgu.ac.kr (J.W.); lishuyu@dongguk.edu (S.L.)
[2] Division of AI Software Convergence, Dongguk University-Seoul, Seoul 04620, Republic of Korea
* Correspondence: sung@dongguk.edu; Tel.: +82-2-2260-3338

Abstract: Audio music genre classification is performed to categorize audio music into various genres. Traditional approaches based on convolutional recurrent neural networks do not consider long temporal information, and their sequential structures result in longer training times and convergence difficulties. To overcome these problems, a traditional transformer-based approach was introduced. However, this approach employs pre-training based on momentum contrast (MoCo), a technique that increases computational costs owing to its reliance on extracting many negative samples and its use of highly sensitive hyperparameters. Consequently, this complicates the training process and increases the risk of learning imbalances between positive and negative sample sets. In this paper, a method for audio music genre classification called Deformer is proposed. The Deformer learns deep representations of audio music data through a denoising process, eliminating the need for MoCo and additional hyperparameters, thus reducing computational costs. In the denoising process, it employs a prior decoder to reconstruct the audio patches, thereby enhancing the interpretability of the representations. By calculating the mean squared error loss between the reconstructed and real patches, Deformer can learn a more refined representation of the audio data. The performance of the proposed method was experimentally compared with that of two distinct baseline models: one based on S3T and one employing a residual neural network-bidirectional gated recurrent unit (ResNet-BiGRU). The Deformer achieved an 84.5% accuracy, surpassing both the ResNet-BiGRU-based (81%) and S3T-based (81.1%) models, highlighting its superior performance in audio classification.

Keywords: music information retrieval; genre classification; pre-training; transformer; audio music

Citation: Wang, J.; Li, S.; Sung, Y. Deformer: Denoising Transformer for Improved Audio Music Genre Classification. *Appl. Sci.* **2023**, *13*, 12673. https://doi.org/10.3390/app132312673

Academic Editors: Isabel Barbancho and Lorenzo J. Tardón

Received: 19 October 2023
Revised: 22 November 2023
Accepted: 23 November 2023
Published: 25 November 2023

1. Introduction

Developments in multimedia technology have resulted in a sharp increase in the variety of digital music and its listening volume, necessitating urgent advancements in music information retrieval (MIR), which involves utilizing computer technology to automatically analyze, recognize, retrieve, and understand music. Audio music genre classification is a MIR task that involves assigning labels to each piece of music based on characteristics such as genre [1,2], mood [3,4], and artist type [5,6]. Audio music genre classification enables the automatic categorization of audio music based on different styles or types, facilitating a deeper understanding and organization of music libraries.

The evolution of deep learning has profoundly affected music genre classification, ushering in an era of automatic feature learning. Convolutional neural networks (CNNs) are proficient in discerning the complex spatial features inherent in audio data [7–10]. However, they are limited in their ability to account for the long-term temporal information inherent in musical compositions. To address this limitation, convolutional recurrent neural networks (CRNNs) [11–13], which combine the strengths of both CNNs and recurrent neural networks (RNNs), are employed in music classification. In the specific context of music genre classification, CRNNs have demonstrated a marked advantage over CNNs,

proficiently discerning both localized features and short-term temporal inter-relationships. Unfortunately, CRNNs still struggle to capture the long-term temporal dependencies that are often crucial in complex musical compositions.

Transformer-based music genre classification approaches, which are fortified with attention mechanisms, have been introduced to address these issues; they have achieved success, particularly in recognizing long-term information in music. Various transformer-based models, such as MusicBERT [14] and MidiBERT [15], have been developed to focus on different aspects of music genre classification. MusicBERT is equipped with specialized encoding and masking techniques that capture complex musical structures, whereas MidiBERT focuses on single-track piano scores. These models can effectively recognize long-term dependencies in music, especially in the context of symbolic music data such as the Musical Instrument Digital Interface (MIDI). Most existing transformer-based models for music classification are primarily tailored for symbolic music data such as MIDI, and there is a notable lack of models that can handle continuous audio data.

A Swin transformer-based approach has emerged as a targeted solution to solve the issues of traditional transformer-based models in handling continuous high-dimensional audio data [16]. This advanced architecture employs a pre-training strategy known as momentum contrast (MoCo), which is a form of contrastive learning. This strategy aims to create similar representations for similar data points while pushing dissimilar data points apart in the feature space by maintaining a dynamic dictionary. Unfortunately, the MoCo pre-training strategy presents its own set of challenges. First, it incurs significantly increased computational costs, owing to the need to maintain and update this large dictionary. Second, the contrastive loss function can be sensitive to hyperparameter choices, thereby complicating the model optimization process. Third, MoCo-based approaches typically suffer from low interpretability, making it difficult to understand the model decisions or identify the learned features that contributed to the classification results.

Additionally, denoising has been extensively researched. Denoising approaches based on self-supervised learning [17,18] via the noise-removal process can effectively capture features and learn deep representations. They have many similarities with self-supervised pre-training strategies, thus making the integration of the denoising concept into pre-training feasible.

In this paper, a novel method for audio music genre classification is proposed. The proposed method is characterized by denoising, which not only reduces computational costs compared to MoCo-based strategies but also offers a robust performance that is uninhibited by hyperparameter dependency. Uniquely, the proposed method incorporates a prior decoder, which substantially enhances the interpretability of the decision-making process. The main contributions of this method are as follows.

- The proposed method includes a novel pre-trained model called Deformer and utilizes unsupervised learning to fully leverage unlabeled data for pre-training.
- The proposed method design includes a prior decoder that assists Deformer in completing the pre-training effectively; it harnesses the potential of transformers in processing image-like audio data. Notably, this prior decoder improves the interpretability of the results obtained by the method.
- The proposed method was experimentally proven to not only lower the computational cost but also achieve better results compared with existing approaches.

The remainder of this paper is organized as follows. Section 2 describes related work on audio-based music genre classification, and Section 3 introduces the proposed music-classification method based on audio data. Then, Section 4 details the experimental process and results. Finally, Section 5 concludes the proposed paper.

2. Related Works

This section provides an overview of the evolution of classification techniques for musical audio data, tracing the transition from methods relying on manual feature extraction to end-to-end models.

2.1. Music Genre Classification Based on Audio Data

Researchers achieved music classification by converting audio data into spectrograms and Mel Frequency Cepstral Coefficients (MFCC) images and subsequently applied texture analysis approaches for feature extraction. Various classifiers, such as K-nearest neighbors (KNN), Gaussian models, and support vector machines (SVMs), were utilized for classification. Notably, the KNN algorithm was effective in classifying classical music [19]. This approach was extended by introducing local binary patterns (LBPs) to extract textural features from spectrograms [20]. The extended version explored partitioning techniques to capture local information, emphasizing the importance of local features in enhancing the classification performance. Although these approaches were effective in specific scenarios, they were generally constrained by their focus on timbral features and failed to capture aspects of music that were potentially crucial for a more comprehensive understanding and music classification.

Informative musical patterns could be automatically identified using CNNs [7]. Nonetheless, these rudimentary CNN models were restricted in generalizing previously unencountered music datasets. To overcome the limitations of these models in generalization and handling long-term temporal information, researchers proposed a hybrid model that combined residual neural networks (ResNets) and gated recurrent units (GRUs). This model used visual spectrograms as inputs and aimed to analyze music data more comprehensively. This approach [13] could improve the performance of music-recommendation systems through more accurate genre classification, thereby addressing the shortcomings of traditional machine learning and basic CNN models in handling the complexity of music data.

By contrast, an approach has been proposed using S3T [16], which is a self-supervised pre-training approach with the Swin Transformer that effectively handles long-term information in music classification. This approach primarily aimed to learn meaningful music representations from a large corpus of unlabeled music data. It employed the momentum-based paradigm MoCo to serve as a feature extractor in the time–frequency domain of music and utilized a music data-augmentation pipeline and two specially designed preprocessors to further optimize the learning of music representations. However, this approach faced challenges such as an increased computational overhead owing to the management of large dynamic dictionaries, sensitivity to hyperparameter selection in the optimization process, and a lack of model interpretability.

Progress in music classification had shifted from a focus on timbral features to more advanced and automated feature identification and extraction. Each stage of this evolution was associated with different challenges, ranging from limited generalization and a narrow focus on certain musical aspects to increased computational demands and complex training requirements. This underscored the ongoing need for obtaining more accurate and computationally efficient solutions and overcoming the persistent challenges of balancing model complexity and interpretability.

2.2. Comparison of Music Genre Classification Based on Audio Data

The comparison primarily considered two dimensions: one from the perspective of data types and another from the perspective of model structures. Regarding the perspective of input data, some researchers treated music classification as a visual task, converting audio music data into spectrograms [13,16,19,20] and MFCC images [7]. These approaches emphasized the visual characteristics of music data. Spectrograms could capture the local characteristics of audio signals in a time–frequency dimension in an intuitive and computationally efficient manner, thereby providing a robust and information-rich feature representation for music-classification tasks.

Second, there was an evolution from initially employing classifiers [19], such as KNN, the Gaussian Mixture Model (GMM), and SVM, to incorporating deep CNN structures and combining them with RNNs. Regarding RNNs, they faced challenges such as difficulties in learning long-term dependencies. These shortcomings were addressed by introducing

S3T-based models, which offered advantages in capturing long-range dependencies. Unfortunately, these models faced challenges in terms of computational costs and robustness as they evolved from simple to complex and singular to multifaceted. Especially when considering how to effectively capture and process the long-term information of music, these technological variances and advancements became pivotal in addressing the challenge. Notably, the proposed method offered a distinct approach that specifically addressed current limitations by expanding upon existing methodologies. The specifications of the proposed and existing approaches are listed in Table 1.

Table 1. Differences between existing approaches and the proposed method.

Research Contents	[19]	[20]	[7]	[13]	[16]	The Proposed Method
Data Rep.	Spec. & MFCC	Spec.	MFCC	Spec.	Spec.	Spec.
Model	KNN, GMM, SVM	SVM	CNN	ResNet-BiGRU	Swin Transformer	Deformer
Pre-training	-	-	-	-	√	√

3. Denoising Transformer-Based Audio Music Genre Classification

The architecture and training strategies for the Deformer-based method are detailed next. First, the data representation techniques are discussed; then, the pre-training and fine-tuning stages are outlined.

3.1. Overview

A method utilizing pre-training techniques based on Deformer was proposed for audio music genre classification. The proposed method consists of pre-training and fine-tuning stages, as shown in Figure 1. First, unlabeled or labeled audio data are preprocessed into a normalized Mel spectrogram, and a noise-injection operation is applied in the pre-training stage, during which Deformer learns deep representations of audio music from unlabeled audio data. For this, a prior decoder is utilized to restore the denoised Mel spectrogram from the low-dimensional hidden states, which is obtained from Deformer. In the fine-tuning stage, the pre-trained Deformer and classifier are further trained using labeled audio data to perform music genre classification. The flowchart for the proposed method is shown in Figure 2.

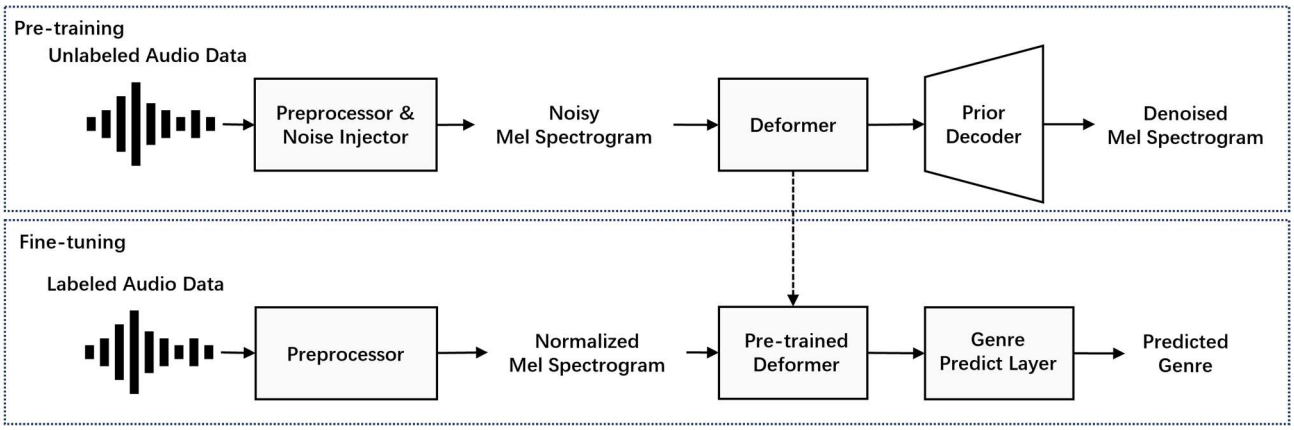

Figure 1. Overview of the method: pre-training and fine-tuning stages.

3.2. Preprocessing and Noise Injection

The preprocessing is employed in the pre-training and fine-tuning stages, as illustrated in Figure 3. Initially, audio data are converted into Mel spectrograms with dimensions W and H, corresponding to time and frequency, respectively. These spectrograms are then resized using the librosa library to new dimensions, H' and W', which are determined

based on the experimental hardware. It is worth noting that the values of H' and W' should be carefully chosen; excessively large dimensions may incur a larger calculation resource usage. Finally, the resized Mel spectrograms are normalized by scaling the values to fit within a range from zero to one.

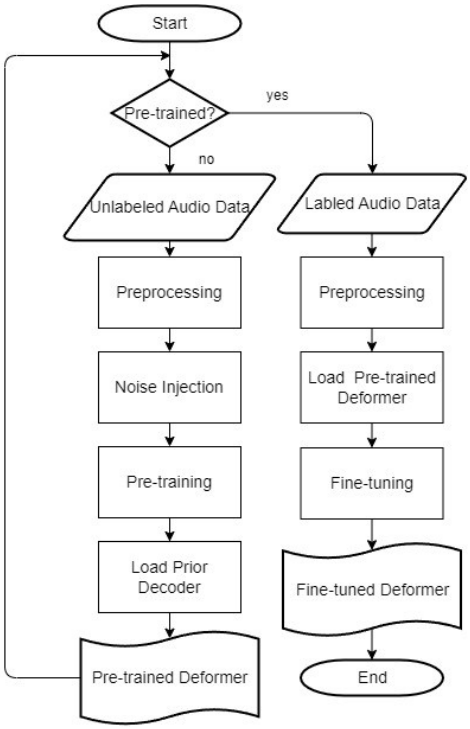

Figure 2. Flowchart of pre-training and fine-tuning under the proposed method.

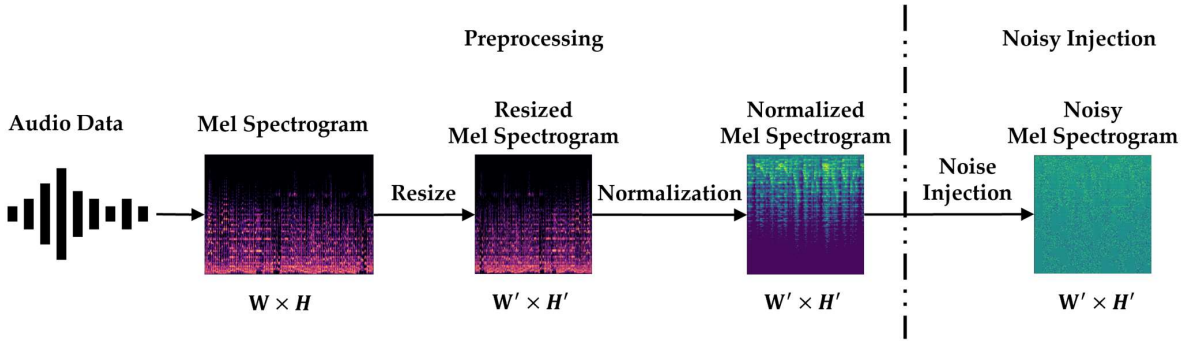

Figure 3. Preprocessing and noise injection of data.

Noise injection is operated additionally only in the pre-training stage. They are divided by N equal-sized patches through a matrix-division operation. Each patch is derived by dividing the Mel spectrogram into N sections, resulting in patches with dimensions of $W'/\sqrt{N} \times H'/\sqrt{N}$, where W' and H' are the width and height of the transformed spectrogram, respectively. Patches $\{p_1, p_2, p_3, ..., p_n, \ldots, p_N\}$, follow the order from left to right and top to bottom. The noise ratio, $\beta\%$, dictates the fraction of patches that receive noise. The $\beta\%$ of patches are injected with noise z with a Gaussian distribution $\mathcal{N}(0,\ 1)$; otherwise, they remain unchanged. Finally, a noisy Mel spectrogram, which is the combination of $\{p_1{}', p_2{}', p_3{}', ..., p_n{}', \ldots p_N{}'\}$, is obtained:

$$p_n{}' = \begin{cases} p_n + z \; if \; \beta\% \\ p_n \; else \end{cases}, \; z \sim \mathcal{N}(0,\ 1) \tag{1}$$

3.3. Pre-Training Stage

The objective of the pre-training stage is to enable Deformer to understand the deep representation of audio music through unsupervised denoising. The role of the prior decoder within this framework is to restore patches from low-dimensional hidden states obtained from Deformer during pre-training. An autoencoder (AE), which comprises an encoder and a decoder, was designed to train the decoder, as shown in Figure 4. The encoder consists of max-pooling and convolutional layers, and the decoder consists of convolutional and up-sampling layers. The encoder compresses the input patches into low-dimensional vectors, and the decoder restores the low-dimensional hidden states to the original patches. The mean squared error (MSE) loss is calculated to update the AE parameters.

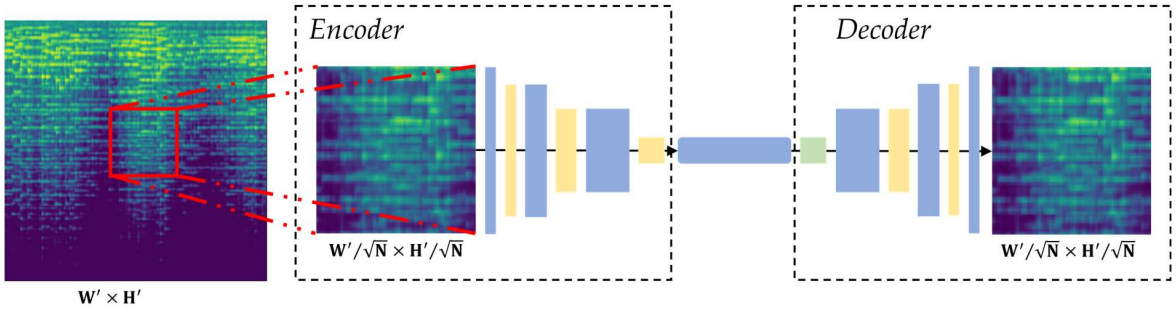

Figure 4. Structure of the autoencoder.

To complete the denoising, patches of noisy Mel spectrograms $\{p_1', p_2', p_3', ..., p_n', \ldots p_N'\}$ are passed into Deformer, as shown in Figure 5. The position embedding layer, which is trainable, utilizes absolute numerical embedding to integrate positional information into these patches. The transformer layers utilize multi-head self-attention and feed-forward neural networks to relate to this layer. Subsequently, the prior decoder restores these low-dimensional hidden states back into the restored patches $\{p_1^*, p_2^*, p_3^*, ..., p_n^*, \ldots p_N^*\}$.

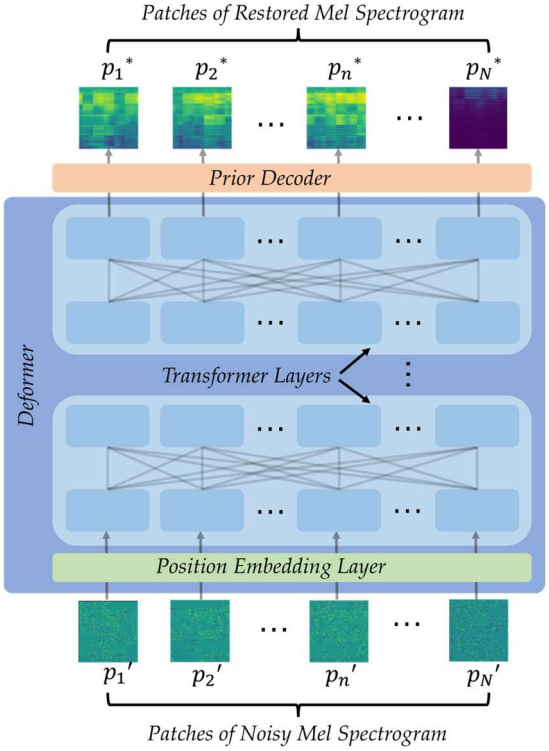

Figure 5. Pre-training stage of Deformer.

Then, the MSE loss is calculated based on the restored patches $\{p_1{}^*, p_2{}^*, p_3{}^*, ..., p_n{}^*, \ldots p_N{}^*\}$ and original patches $\{p_1, p_2, p_3, ..., p_n, \ldots, p_N\}$ for training Deformer. This training strategy allows Deformer to gain a deep understanding of the contextual relationships and interdependencies among patches.

Algorithm 1 describes the pre-training stage, where Deformer(•) represents Deformer and θ represents Deformer parameters. I denotes the number of training steps. $p_n{}'$ represents one of the patches from the noisy Mel spectrograms, and $p_n{}^*$ is the restored patch. MSE(•) represents the loss function to calculate the loss L between $p_n{}^*$ and p_n, where $p_n{}^*$ is only generated by the injected noise z and p_n is the patch of the normalized Mel spectrogram without noise. θ is updated based on a gradient, which is calculated as $-\eta^*\nabla L(\theta)$, where η is the learning rate.

Algorithm 1 Pre-training

Input: $\{p_1{}', p_2{}', p_3{}', ..., p_n{}', \ldots p_N{}'\}$
Output: $\{p_1{}^*, p_2{}^*, p_3{}^*, ..., p_n{}^*, \ldots p_N{}^*\}$
1: Initialize Deformer (θ)
2: for i = 1 to I do:
3: Forward pass: $p_n{}^*$ = Deformer $(p_n{}', \theta)$
4: If $p_n{}' = p_n + z$:
5: Compute $L = \text{MSE}(p_n{}^*, p_n)$
6: Update $\theta = \theta - \eta^*\nabla L(\theta)$
7: End for

3.4. Fine-Tuning Stage

In the fine-tuning stage, the Deformer that has already learned the deep representation of audio music is applied to music genre classification. Figure 6 shows the process of fine-tuning the pre-trained Deformer to a classification network. Different from the pre-training stage, the fine-tuning stage is not unsupervised learning, and the prior decoder is not utilized. Normalized Mel spectrogram patches $\{p_1, p_{2,}p_3, ..., p_n, \ldots, p_N\}$ without added noise are fed into the model, along with a classifier token (*cls*) [21]. The *cls* token serves as a condensed representation of all input patches. As opposed to pre-training, which involves decoding layers, fine-tuning employs a genre prediction layer connected to the final Deformer position. This layer is a linear component that uses a SoftMax function to predict the probability distribution for the genre classes based on the *cls* token's hidden state. The cross-entropy loss is then computed using the predicted and target genres to fine-tune the Deformer and the genre prediction layers, enhancing Deformer's ability to classify music genres effectively.

Algorithm 2 details the fine-tuning process, where p_N represents the normalized patches, g represents the target genre, g' represents the predicted genre of Deformer, and *cross_entropy*(•) calculates the loss of g' and g for performing updates.

Algorithm 2 Fine-tuning

Input: $\{p_1, p_2, p_3, ..., p_n, \ldots, p_N\}$
Output: g'
1: Load pre-trained Deformer (θ)
2: for i = 1 to I do:
3: Forward pass: g' = Deformer (p_n, θ)
4: Compute $L = cross\_entropy(g', g)$
5: Update $\theta = \theta - \eta^*\nabla L(\theta)$
6: End for

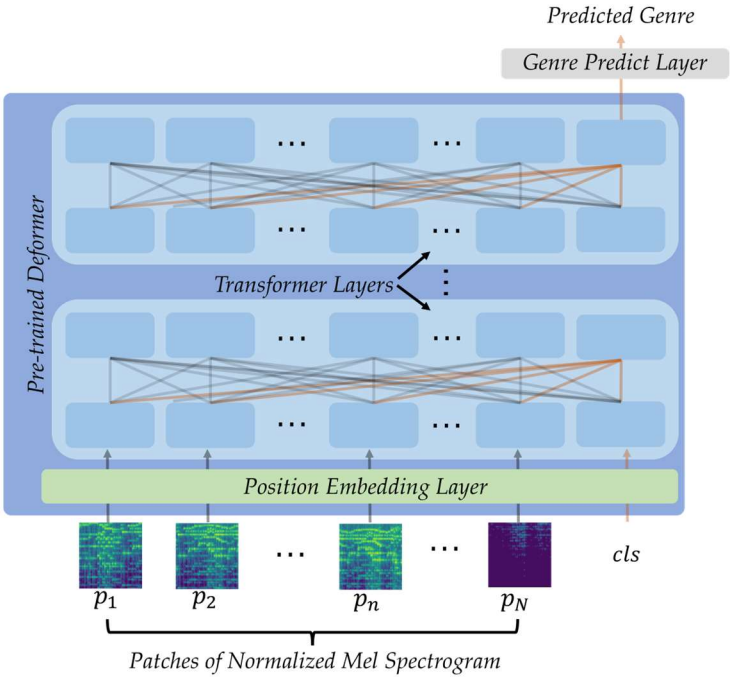

Figure 6. Fine-tuning stage of Deformer.

4. Experiments and Results

Three experiments, namely, prior decoder training, Deformer pre-training, and Deformer fine-tuning, were performed to thoroughly evaluate the effectiveness of the proposed Deformer-based method in terms of audio music genre classification. First, in the prior decoder training experiment, an autoencoder was trained to convert low-dimensional hidden states into patches of normalized Mel spectrograms. In the Deformer pre-training experiment, Deformer was pre-trained to understand musical deep representations by restoring the original Mel spectrograms from noisy Mel spectrograms. Finally, in the Deformer fine-tuning experiment, the pre-trained Deformer classified the audio music genre through supervised learning. To evaluate the performance and effectiveness of the proposed method, two baseline models are introduced for comparison. The first model [13] utilizes a residual neural network–bidirectional gated recurrent unit (ResNet-BiGRU), while the second relies on S3T [16]. These models serve as benchmarks, helping to underscore the advantages of the proposed technique for music genre classification.

4.1. Experimental Environment

Table 2 summarizes all the hyperparameters used in the three experiments. The autoencoder comprises an encoder and a decoder; the encoder consists of three convolutional layers utilizing the same kernels but with different channels. The Deformer hyperparameters include 196 patches, a patch size of 16 × 16, a hidden size of 768, four intermediate multiplications, 12 hidden layers, and 12 attention heads.

During the training of the three models, the resized Mel Spec $(W' \times H')$ size was set to 224 × 224, which can be adjusted according to the hardware of the experimental environment. As mentioned before, noise injection was operated in pre-training. The noise injection ratio β was determined by experimental results, given that the highest classification performance was obtained when β was set to 0.75. Similarly, 0.75 was also used as the mask parameter in [22], which similarly achieved good results. As the input to the prior decoder is a patch, it allows for a higher batch size compared to others. The parameters of the AdamW optimizer were nearly similar. When setting the learning rate, it was considered that pre-training requires warmup. Unlike other approaches that use a fixed learning rate, pre-training employed a dynamically changing learning rate based on the WarmupDecayLR scheduler.

Table 2. Hyperparameters used in the autoencoder, pre-training, and fine-tuning experiments.

Hyperparameters	Autoencoder (Training)	Pre-Training	Fine-Tuning
Kernel Size	3×3	-	-
Channels of Encoder	32, 64, 128	-	-
Channels of Decoder	128, 64, 32	-	-
Number of Patches	-	196	196
Patch Size	-	16×16	16×16
Hidden Size	-	768	768
Intermediate Multiplication	-	4	4
Number of Hidden Layers	-	12	12
Number of Attention Heads	-	12	12
Size of resized Mel Spec. ($W\prime \times H'$)	224×224	224×224	224×224
Noise Ratio (β)	-	0.75	-
Batch Size	64	16	16
Optimizer	AdamW	AdamW	AdamW
AdamW Betas	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)
Weight Decay	0.01	0.01	0.01
Learning Rate (η)	1×10^{-4}	-	1×10^{-6}
Scheduler	-	WarmupDecayLR	-
Minimum Warmup Learning Rate	-	1×10^{-6}	-
Maximum Warmup Learning Rate	-	1×10^{-4}	-
Warmup Steps	-	800	-
Total Training Steps (I)	8000	8000	20,000

The experiments were conducted on a system running Windows 10 with 2 Xeon(R) Silver 4310 CPUs, 4 NVIDIA GeForce RTX 3090 GPUs, and 128 GB of DDR4 RAM. The proposed method was developed in Python 3.10.12 and implemented using the PyTorch 2.0.0 platform, complemented by the DeepSpeed acceleration engine for enhanced performance.

In addition to conducting these experiments, a comparative assessment was performed with two baseline models. The first baseline model [13] employed a hybrid approach, combining ResNet18 and Bi-GRU. ResNet18 utilizes residual connections, comprising 18 weighted layers, including an initial convolutional layer, a max-pooling layer, 4 convolutional blocks (each with 2 convolutional layers), an average pooling layer, and a fully connected layer. Bi-GRU is a recurrent neural network designed for processing sequential data, consisting of a GRU layer and a fully connected layer.

The second baseline model [16] adopted S3T, leveraging the Swin Transformer as a feature extractor in the time–frequency domain of music. It integrates a momentum-based MoCo paradigm for enhanced performance. The feature extractor follows the Swin-T configuration, using the compact version of the Swin Transformer with a hidden channel number of 96. Each block comprises 2, 2, 6, and 2 layers, ensuring increased efficiency.

4.2. Experimental Data

Two distinct datasets, each divided into an 80% training set and a 20% test set, were employed in the genre classification experiment involving audio music data. The MAESTRO dataset [23] was used for feature extraction via an autoencoder and for pre-training Deformer. This dataset encompasses a broad spectrum of musical instruments and styles,

with contributions from both professionals and amateur musicians. Mel spectrograms derived from raw audio files were used as inputs. The training set was used for model optimization using techniques such as gradient descent, and the test set was designated for performance evaluation using metrics such as MSE. The GTZAN [24] music dataset was exclusively used to fine-tune Deformer. Renowned in genre classification, this dataset consists of one thousand 30 s audio segments across ten distinct genres, including blues, classical, and hip hop. The audio clips were transformed into Mel spectrograms to serve as inputs for the model. The training process involved iterative Deformer updates based on loss minimization, and the test phase assessed the genre-classification capabilities of Deformer in terms of the precision and recall metrics.

4.3. Experimental Results

The results from the autoencoder training, pre-training, and fine-tuning experiments were analyzed. The initial results indicated a rapid loss function convergence, validating the effectiveness of decoder training. Further findings from the pre-training and fine-tuning processes revealed that Deformer exhibited superior performance in music data processing, outperforming the baseline models in multiple key performance metrics.

4.3.1. Prior Decoder Training Results

Figure 7 shows the prior decoder training experiment results, which involved 8000 steps. The MSE loss decreased rapidly from 0.14 to 0.02. Subsequently, the loss continued to decrease at a slower pace, eventually converging to approximately 0.001.

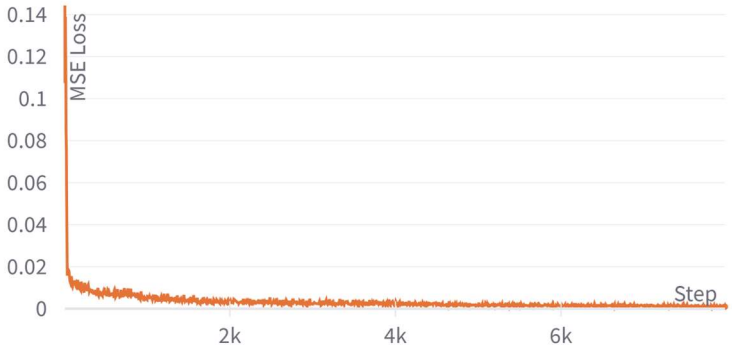

Figure 7. MSE loss in the prior decoder training experiment.

Figure 8 shows the test results of the prior decoder experiment and the reference for comparison. Figure 8a shows the Mel spectrograms assembled from the patch output reconstructed by the decoder, while Figure 8b shows the original Mel spectrograms used for comparison with the reconstructed version; subtle local differences can be observed in the areas marked with red boxes. Interestingly, the Mel spectrograms assembled from the reconstructed patches in Figure 8a were almost indistinguishable from the original Mel spectrograms in Figure 8b, demonstrating that the prior decoder could effectively reconstruct the Mel spectrograms from the low-dimensional hidden states.

4.3.2. Pre-Training Results

Figure 9 shows the two distinct phases in the loss curve during the model training process. Initially, the loss value rapidly decreased from a higher level to approximately 0.15, after which the rate of decline significantly decreased and eventually stabilized at approximately 0.01 after approximately 8000 steps.

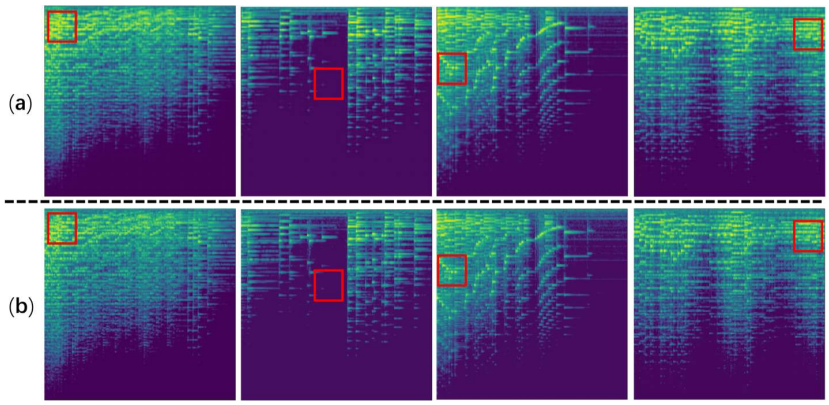

Figure 8. Output of the decoder. (**a**) Reconstructed Mel spectrograms and (**b**) original Mel spectrograms. The red boxes indicate subtle differences between (**a**,**b**).

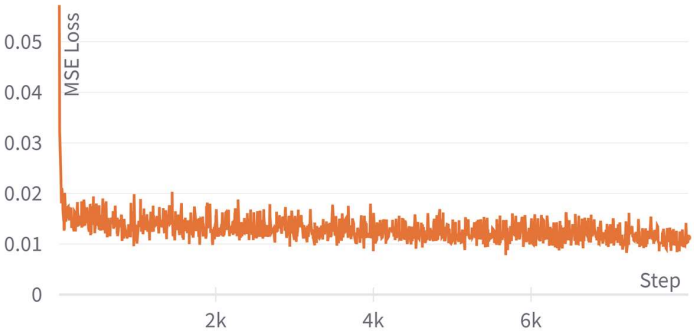

Figure 9. MSE loss in the pre-training experiment.

Figure 10 shows the pre-training stage of Deformer using Mel spectrograms constructed from the patches. Figure 10a shows the Mel spectrograms assembled from patches injected with noise, which served as the inputs to the model during the pre-training stage. Figure 10b shows the Mel spectrograms assembled from the denoised patches, which are the outputs of Deformer. Finally, Figure 10c shows the original noise-free Mel spectrograms. The principal features and trends shown in Figure 10c are successfully captured, as shown in Figure 10b, albeit with some loss of detail, demonstrating the capabilities of Deformer in terms of noise reduction and learning meaningful representations of music data. These observations further emphasize the effectiveness of the pre-training stage as well as the preparedness of the pre-trained Deformer for the next fine-tuning stage.

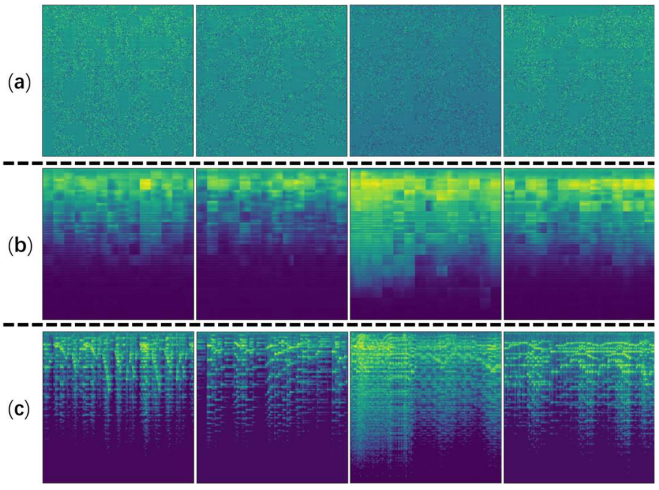

Figure 10. Pre-training stage of Deformer. (**a**) Mel spectrograms with injected noise; (**b**) denoised Mel spectrograms; and (**c**) original Mel spectrograms.

4.3.3. Fine-Tuning Results

Figure 11 shows the loss changes of Deformer during fine-tuning. The orange line (pre-trained) demonstrates a rapid decline in loss during fine-tuning, indicating a high level of learning efficiency. The blue line (without pre-training) exhibits a slower loss decrease. At 20,000 steps, the fine-tuning process based on pre-training demonstrated a significant performance advantage compared to that without pre-training.

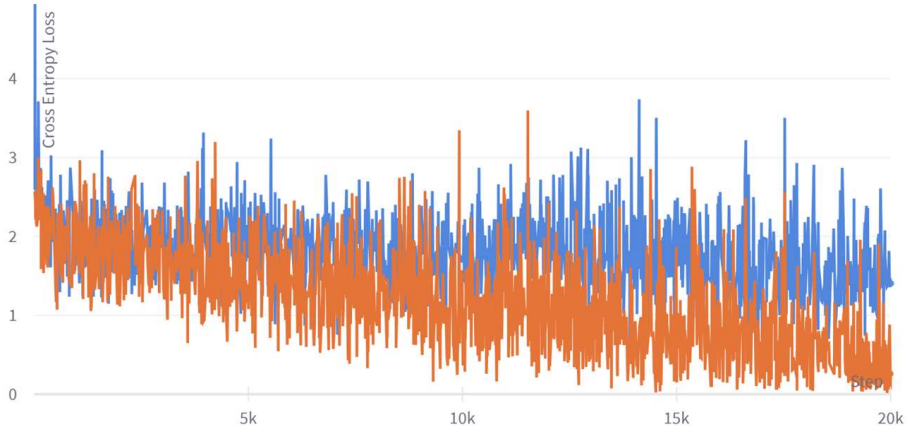

Figure 11. Cross entropy loss in fine-tuning of Deformer and ablation experiment.

To assess the classification efficacy of Deformer across different music genres, the confusion matrix depicted in Figure 12 is provided. The confusion matrix presents true-positive, true-negative, false-positive, and false-negative results, providing a clear classification performance evaluation. The fact that the predicted results are clearly distributed along the diagonal of the confusion matrix indicates that most of the predictions are correct. It can be seen that Deformer exhibited exceptional performance in the "classical" and "pop" categories, achieving impeccable accuracy with zero misclassifications within these genres. This outcome highlights its acute understanding of the unique attributes associated with these music genres. However, it exhibited inaccuracies within the "rock" and "blues" genres. Specifically, a few samples falling under the "rock" category were incorrectly classified as "blues" and "metal". Likewise, a subset of "blues" samples was inaccurately classified as "jazz" and "metal". These misclassifications suggested potential limitations of Deformer, particularly when differentiating between genres having nuanced or overlapping traits. Analyzing the confusion matrix is vital as it paves the way for prospective refinements and emphasizes the need to improve the discriminatory capabilities of the model when classifying music belonging to closely related genres such as "rock" and "blues".

Table 3 presents the accuracy, precision, recall, and F1 scores for the proposed pre-trained Deformer, Deformer without pre-training used for the ablation experiment, and ResNet-BiGRU and S3T as two baseline models. To complete the comparison, two additional results [25,26] are given, which demonstrated high accuracy in audio music genre classification. The pre-trained Deformer reached a classification accuracy of 84.5%, which is 3.4% higher than that of ResNet-BiGRU (81%), 3.3% higher than that of S3T (81.1%), 0.6% higher than that of M2D [25] (83.9%), and 4.8% higher than that of the Jukebox model pre-trained with CALM (79.7%) [26]. The pre-trained Deformer significantly outperformed its non-pre-trained counterpart in terms of accuracy, precision, recall, and F1 score, with the latter only achieving an accuracy and recall of 0.37, a precision of 0.3334, and an F1 score of 0.3464. This comparison highlights the importance of pre-training in enhancing the performance of Deformer for music classification. It is worth noting that all data presented in Table 3 were obtained through testing on the GTZAN dataset.

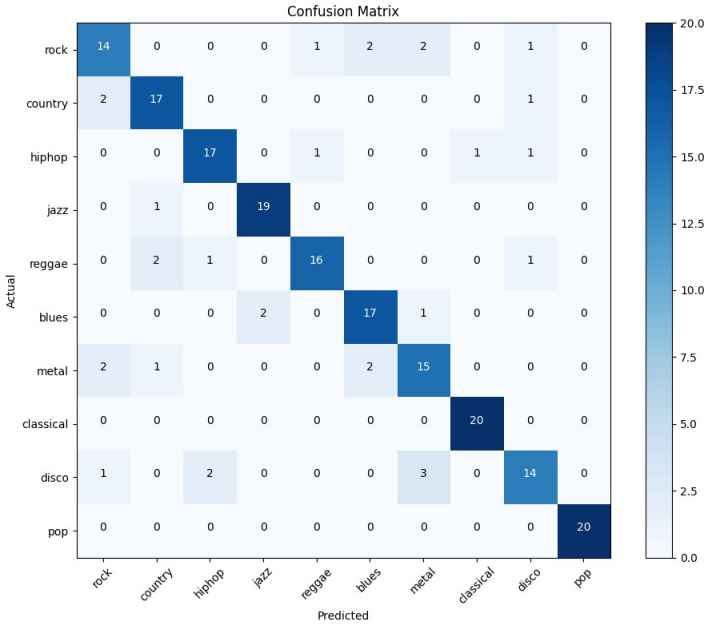

Figure 12. Confusion matrix of audio music genre classification performed by Deformer.

Table 3. Performance comparison: audio music classification models.

Model	Accuracy	Precision	Recall	F1 Score
ResNet-BiGRU	0.81	0.82	0.81	0.81
S3T	0.811	-	-	-
M2D	0.839	-	-	-
Juckbox	0.797	-	-	-
Deformer (w/o pre-train)	0.37	0.3334	0.37	0.3464
Deformer (pre-trained)	0.845	0.844	0.845	0.844

5. Conclusions

A pre-trained model, Deformer, was introduced to address the specific challenges associated with existing Swin transformer-based approaches in music genre classification within the context of MIR. These challenges include the computational burden associated with managing large dynamic dictionaries, the finicky nature of the contrastive loss function with respect to hyperparameter choices, and the low level of model interpretability commonly observed in MoCo-based approaches. Utilizing a two-stage process of pre-training and fine-tuning, the proposed model leveraged unlabeled audio data during the pre-training stage. The experimental results underscore the significance of incorporating Deformer in the realm of deep learning architectures for audio music classification. The proposed method achieved an accuracy of 84%, outperforming the ResNet-BiGRU-based (81%) and S3T-based (81.1%) models. This highlights the substantial contribution of Deformer to superior performance in audio classification, marking a noteworthy advancement over traditional approaches.

Regarding its limitations, the proposed model was not assessed on larger or more diverse datasets, creating gaps in information regarding its generalizability. Future research directions could involve restructuring the architecture of the model to enable it to better handle genres that have subtle similarities, such as “rock” and “blues”. The focus should be on enhancing the ability of the model to distinguish between closely aligned genres. Further improvements can be made to evaluate the performance of the model across a more diverse set of music genres and use cases. By pursuing these avenues, this research would not only add to the growing literature in the domain of music genre classification but also set a strong performance standard in subsequent investigations.

Author Contributions: Conceptualization, J.W., S.L. and Y.S.; methodology, J.W., S.L. and Y.S.; software, J.W. and S.L.; validation, J.W., S.L., and Y.S.; formal analysis, J.W., S.L. and Y.S.; data curation, J.W. and S.L.; writing—original draft preparation, J.W.; writing—review and editing, J.W., S.L. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2021R1F1A1063466). This work was supported by the Dongguk University Research Fund of 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found here https://magenta.tensorflow.org/datasets/maestro and http://marsyas.info/download/data_sets.

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

References

1. Qiu, L.; Li, S.; Sung, Y. DBTMPE: Deep Bidirectional Transformers-Based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics* **2021**, *9*, 530. [CrossRef]
2. Prabhakar, S.K.; Lee, S.-W. Holistic Approaches to Music Genre Classification Using Efficient Transfer and Deep Learning Techniques. *Expert Syst. Appl.* **2023**, *211*, 118636. [CrossRef]
3. Jin, P.; Si, Z.; Wan, H.; Xiong, X. Emotion Classification Algorithm for Audiovisual Scenes Based on Low-Frequency Signals. *Appl. Sci.* **2023**, *13*, 7122. [CrossRef]
4. Thao, H.T.P.; Roig, G.; Herremans, D. EmoMV: Affective Music-Video Correspondence Learning Datasets for Classification and Retrieval. *Inf. Fusion* **2023**, *91*, 64–79. [CrossRef]
5. Kong, Q.; Choi, K.; Wang, Y. Large-Scale MIDI-Based Composer Classification. *arXiv* **2020**, arXiv:2010.14805.
6. Nasrullah, Z.; Zhao, Y. Music Artist Classification with Convolutional Recurrent Neural Networks. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019.
7. 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; IEEE: Piscataway, NJ, USA, 2017.
8. Li, T.; Chan, A.B.; Chun, A. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. *Genre* **2010**, *10*, 1x1.
9. Lee, J.; Park, J.; Kim, K.; Nam, J. SampleCNN: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification. *Appl. Sci.* **2018**, *8*, 150. [CrossRef]
10. Choi, K.; Fazekas, G.; Sandler, M. Automatic Tagging Using Deep Convolutional Neural Networks. *arXiv* **2016**, arXiv:1606.00298.
11. 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; IEEE: Piscataway, NJ, USA, 2017.
12. 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]
13. Zhang, J. Music Genre Classification with ResNet and Bi-GRU Using Visual Spectrograms. *arXiv* **2023**, arXiv:2307.10773.
14. Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; Liu, T.-Y. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. *arXiv* **2021**, arXiv:2106.05630.
15. Chou, Y.H.; Chen, I.; Chang, C.J.; Ching, J.; Yang, Y.H. MidiBERT-Piano: Large-Scale Pre-Training for Symbolic Music Understanding. *arXiv* **2021**, arXiv:2107.05223.
16. Zhao, H.; Zhang, C.; Zhu, B.; Ma, Z.; Zhang, K. S3T: Self-Supervised Pre-Training with Swin Transformer for Music Classification. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; IEEE: Piscataway, NJ, USA, 2022.
17. Zhang, Q.; Xiao, J.; Tian, C.; Chun-Wei Lin, J.; Zhang, S. A Robust Deformed Convolutional Neural Network (CNN) for Image Denoising. *CAAI Trans. Intell. Technol.* **2023**, *8*, 331–342. [CrossRef]
18. Xue, T.; Ma, P. TC-Net: Transformer Combined with CNN for Image Denoising. *Appl. Intell.* **2023**, *53*, 6753–6762. [CrossRef]
19. Deshpande, H.; Singh, R. Classification of Music Signals in the Visual Domain. In Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, 6–8 December 2001.
20. Costa, Y.M.G.; Oliveira, L.S.; Koerich, A.L.; Gouyon, F.; Martins, J.G. Music Genre Classification Using LBP Textural Features. *Signal Process.* **2012**, *92*, 2723–2737. [CrossRef]
21. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.

22. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. *arXiv* **2021**, arXiv:2111.06377.
23. Hawthorne, C.; Stasyuk, A.; Roberts, A.; Simon, I.; Huang, C.-Z.A.; Dieleman, S.; Elsen, E.; Engel, J.; Eck, D. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. *arXiv* **2018**, arXiv:1810.12247.
24. Sturm, B.L. The GTZAN Dataset: Its Contents, Its Faults, Their Effects on Evaluation, and Its Future Use. *arXiv* **2013**, arXiv:1306.1461.
25. Niizumi, D.; Takeuchi, D.; Ohishi, Y.; Harada, N.; Kashino, K. Masked Modeling Duo: Learning Representations by Encouraging Both Networks to Model the Input. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rodos, Greece, 4–10 June 2023; IEEE: Piscataway, NJ, USA, 2023.
26. Castellon, R.; Donahue, C.; Liang, P. Codified Audio Language Modeling Learns Useful Representations for Music Information Retrieval. *arXiv* **2021**, arXiv:2107.05677.