Rossi *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2025) 2025:38
https://doi.org/10.1186/s13636-025-00426-1

EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH

Open Access

Advancing guitar emotion recognition through audio data augmentation to enhance smart musical instruments

Michele Rossi[1*], Giovanni Iacca[1] and Luca Turchet[1]

Abstract

Dataset augmentation techniques have been widely used to achieve state-of-the-art results in Music Information Retrieval tasks. However, their application in music emotion recognition (MER) remains underexplored. MER methods are particularly relevant to the design of smart musical instruments (SMIs), as emotionally aware SMIs have the potential to enrich musical interaction by providing feedback to musicians or dynamically adjusting their sound properties. In this study, we analyze the effect of 11 augmentation techniques on emotion classification in guitar recordings using a convolutional neural network. Our dataset consists of approximately 400 guitar recordings labeled with four emotions: aggressiveness, relaxation, happiness, and sadness. Results indicate that time shift, time stretch, and pitch shift provide the most significant improvements in classification accuracy. Further analysis combining these techniques under different settings yielded similar performance outcomes. A listening test confirmed that the applied augmentations did not significantly alter the perceived emotional content of the recordings. These findings support the development of emotionally aware SMIs by enhancing MER accuracy through data augmentation, ultimately enabling more expressive and interactive music-making experiences.

Keywords Dataset augmentation, Music emotion recognition, Convolutional neural networks, Smart musical instruments

1 Introduction

Smart musical instruments (SMIs) are an emerging class of musical devices that incorporate sensors, connectivity, artificial intelligence, and real-time feedback to enhance musical performance and creativity [1]. These instruments are envisioned to provide advanced functionalities such as gesture recognition, adaptive sound control, and interactive learning environments. One promising application of SMIs is music emotion recognition (MER), whose main goal is to map music to the emotions it conveys [2]. This would enable instruments to respond to the emotional content of music by providing feedback or dynamically adjusting sound properties [3]

However, developing accurate MER systems requires large and diverse labeled datasets, which are often time-consuming to collect and annotate. This is especially true for deep learning models with high capacity, i.e., a large number of parameters, where the issue of overfitting the training data is common [4].

Data augmentation techniques aim to mitigate this problem by artificially expanding the training samples of the considered dataset. This is achieved by applying transformations or modifications to the original samples. These techniques are commonly employed in various deep learning tasks, such as computer vision [5], natural language processing [6], and audio processing [7].

Besides increasing the number of available data samples, methods for data augmentation also have a

*Correspondence:
Michele Rossi
michele.rossi-2@unitn.it
[1] Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, 38123 Trento, Italy

secondary effect: they tend to increase the robustness of deep learning models by feeding distorted or modified versions of the original samples to the model during training. Therefore, even if some augmentation does not provide a significant increase in classification accuracy, it may still contribute to an increase in the model's robustness [8]. In the specific case of the audio domain, an effective example of such increased robustness is the ability of a model to properly classify noisy audio samples after implementing specific data augmentation during model training [8].

As described in the following section, some data augmentation techniques have been explored in the audio domain, mainly considering tasks related to voice detection [9, 10], environmental sound classification [11, 12], and music signals [13, 14]. Considering the latter, the majority of contributions have concerned the field of music genre classification [8].

While data augmentation techniques have been explored for emotion recognition in speech signals [15], to the best of our knowledge, they have not yet been systematically investigated in the context of MER. Lately, this field has received considerable attention from the research community [2, 16–20], as it can be beneficial in several practical applications such as music recommendation systems [21], music therapy [22], and media content creation [23], among others.

Although the MER field offers some publicly available datasets, these datasets typically do not contain enough samples to effectively train large deep learning models [19]. While some works have applied deep learning techniques for MER tasks [24–28], the problem of artificially increasing the dataset for MER has been largely overlooked thus far.

Our main aim with the present study is to fill this gap. To do so, we thoroughly explore 11 data augmentation techniques and assess which are the most convenient when dealing with MER tasks. Given that MER is a rather complex task, we decided to restrict our analysis to the classification of emotions in guitar recordings, using the dataset provided in [29]. As a first result, we offer insights related to the effectiveness of each data augmentation technique for the classification, providing the corresponding variation in the model performances. Secondly, we analyze the effect of applying multiple augmentation techniques in combination (selecting from among the best-performing ones). Moreover, we report the results obtained by the statistical tests that we performed for both the single and the multiple augmentation setup.

Finally, to verify whether the label (i.e., the emotion) of each audio file is preserved after the application of the data augmentation techniques, we conducted a listening test where we measured the variation of the perceived emotions associated with different music excerpts.

The remainder of the paper is structured as follows. Section 2 presents a brief overview of related work. Section 3 describes the implemented audio data augmentation techniques, the main pre-processing steps, and the deep learning model. Section 5 outlines the setup and results of the listening test. Section 6 provides a general discussion, including the main limitations. Finally, Section 7 offers the concluding remarks.

2 Related works

As mentioned earlier, our literature review did not yield any studies providing insights into the effect of data augmentation techniques for the MER field. Therefore, hereinafter we aim to offer a general overview of the main contributions related to data augmentation in audio, specifically within the realm of Music Information Retrieval (MIR).

Initially, data augmentation techniques have been primarily applied in the field of computer vision [30], yielding impressive results when dealing with large deep learning models [31]. Common data augmentation techniques for image data include rotation, flipping, translation, scaling, and brightness adjustment. While these methods are effective for natural images, they are generally not suitable for audio signals—even when audio is converted into spectrograms, which resemble images visually. This is because transformations such as rotation or flipping can distort the time-frequency structure of a spectrogram in ways that do not correspond to realistic variations in sound. For example, flipping a spectrogram vertically would invert the frequency axis, which has no meaningful interpretation in most audio contexts, and rotating it would scramble the relationship between time and frequency, severely degrading its utility for model training. Nonetheless, some studies have shown that such image-based augmentations can still improve audio classification performance in certain settings, despite their lack of semantic relevance [32]. Therefore, with the increasing interest in deep learning models, the audio research community had to develop their own specific transformations to artificially increase the datasets.

The literature provides different studies implementing dataset augmentation for speech recognition (for recent works, the reader is referred to [33–35]). These works typically incorporate specific augmentations suited for voice signals, such as vocal tract perturbation [36]. Other studies have implemented dataset augmentation techniques for the task of environmental sound detection [11, 37, 38]. In particular, the study reported in [11] implemented four different augmentation techniques, namely time stretching, pitch shifting,

dynamic range compression, and background noise addition. Results showed that pitch shifting had the greatest positive impact on accuracy, being the only one to provide a positive effect for all the considered classes.

Data augmentation techniques have been increasingly explored in the field of Music Information Retrieval (MIR), particularly over the past decade [19]. Most of these studies employ convolutional neural networks (CNNs) to evaluate the effectiveness of various augmentation methods across different MIR tasks [39–41].

In [39], the authors used a CNN to investigate the impact of seven data augmentation techniques—including four specific to audio signals—for the task of singing voice detection. These augmentations were applied directly to spectrograms, with pitch shifting identified as the most effective technique.

In [40], the authors used a CNN to compare four audio data augmentation techniques for the task of instrument detection: pitch shifting, time stretching, background noise addition, and dynamic range compression. Unlike the previous study, these techniques were applied in combination, resulting in four configurations: (1) pitch shift only, (2) pitch shift followed by time stretch, (3) pitch shift, time stretch, and noise addition, and (4) all techniques combined. The study reported a 2% increase in accuracy using pitch shift alone, with no statistically significant difference compared to the other augmentation combinations—all approaches resulted in similar accuracy improvements.

Similarly, the study in [41] evaluated the effect of four augmentation methods—noise addition, loudness variation, time stretching, and pitch shifting—using a CNN for the task of music genre classification. Consistent with previous studies [39, 40], pitch shifting (especially with a one-semitone increment or decrement) was found to be the most effective technique.

A more extensive investigation of data augmentation is presented in [8], which differs from the previous works by employing traditional machine learning techniques instead of deep learning models. This study analyzed the effect of 12 audio transformations and explored the role of segmentation in music genre classification. Time stretching was identified as the most effective augmentation. The authors also examined the impact of creating 1, 2, 4, and 14 augmented versions per sample. Interestingly, a single augmented version already yielded significant accuracy improvements, with additional versions offering diminishing returns. As previously mentioned, a limitation of this work is the exclusion of neural network-based models, focusing instead on classical machine learning approaches.

3 Methodology

In this section, we describe the data augmentation techniques we implemented, provide a concise overview of the dataset, elaborate on the pre-processing phase, and delve into the details of the deep learning model used in our experiments.

3.1 Data augmentations

In this section, we provide a brief description of each augmentation technique employed in our study. Prior to applying these techniques, all audio files were loudness-normalized using *pyloudnorm,* a Python package that calculates integrated loudness in accordance with the ITU-R BS.1770 standard. A comprehensive list with the associated parameters is reported in Table 1, which also indicates the library used for implementing each augmentation (if applicable). It is worth noting that the parameter range for each augmentation technique was empirically determined through informal listening tests conducted by the authors, with the aim of creating samples with enough variability without excessively distorting the original audio (e.g., from -3 to $+3$ semitones for the pitch shift augmentation technique). The effective value was then randomly sampled within the given interval. In the case of pitch shift, resampling, and time shift (with time shift measured in samples), we sampled integer numbers, while for all other cases, we sampled floating-point numbers. For these latter cases, the sampling is always uniform, except for high pass and low pass filters, where the sampling is logarithmic. In cases where a data augmentation technique presents more than one parameter, we specify the possible range for the main parameter and set all other parameters to a plausible fixed value. For example, in the case of the compressor, the ratio was set within the range [1, 4], while all other parameters were pre-tuned and kept fixed. Since, as we just mentioned,

Table 1 Data augmentation techniques and their corresponding parameter settings

Augmentation	Module/tool	Range of values
Compressor	pedalboard	Ratio: 1 to 4
High pass	pedalboard	Cutoff frequency: 200 to 2000 Hz
Low pass	pedalboard	Cutoff frequency: 500 to 5000 Hz
Noise addition	numpy	Percentage factor: 0 to 0.05
Pitch shift	librosa	Semitones: − 3 to 3
Random gain	Python	Gain: 0.8 to 1.2
Resampling	pedalboard	Sampling rate: 11025 to 22050 Hz
Reverb	pedalboard	Room size: 0 to 0.1
Saturation	pedalboard	Saturation level: 0 to 20
Time shift	Python	Time shift: 0 to 1.5 s
Time stretch	librosa	Stretch ratio: 0.97 to 1.03

the audio files were normalized in loudness prior to augmentation, we defined a plausible value for the compressor's threshold and modified only the ratio to generate augmented samples. Regarding the low-pass and high-pass filters, note that the chosen cutoff frequencies may affect the guitar's fundamental frequencies (F0), which typically range from about 80 Hz to 1.2 kHz, potentially resulting in a clearly audible change in the sound. This is not the case for resampling: when set to its lowest value (11,025 Hz), it only removes frequency content above approximately 5.5 kHz, as determined by the Nyquist limit.

Compressor. An audio compressor reduces the dynamic range of an audio signal by attenuating the amplitude of loud sounds and amplifying softer sounds. The *ratio* range was set from 1 (no compression) to 4. We utilized the compression module provided by the `pedalboard` library by Spotify [42].

High pass. A high-pass filter allows frequencies above a certain cutoff frequency to pass through while attenuating frequencies below that cutoff frequency. We used `pedalboard` and defined the cutoff frequency in a range from 200 to 2000 Hz. The sampling of the effective value was performed logarithmically to adhere to the curve of human perception of frequency.

Low pass. A low-pass filter allows frequencies below a certain cutoff frequency to pass through while attenuating frequencies above that cutoff frequency. We used `pedalboard` and defined a range between 500 and 5000 Hz for the cutoff frequency. Also in this case, the value was sampled logarithmically.

Noise addition. Noise addition refers to the process of adding various types of noise to the original audio samples. In our case, we implemented white noise addition using the `numpy` library. The noise percentage factor—which controls the relative amplitude of the added noise with respect to the standard deviation of the original signal—was sampled from the range [0, 0.05].

Pitch shift. Pitch shifting is a digital audio processing technique used to alter the pitch of a musical signal while preserving its duration, without affecting its tempo or timing. We used the `librosa` library [43] and defined a range between -3 and $+3$ semitones.

Random gain. Random gain refers to a technique where the gain (i.e., the amplitude level) of an audio signal is randomly adjusted within a specified range. We implemented this augmentation directly in Python, considering the range of gains in [0.8, 1.2].

Resampling. Resampling is a technique used to modify the sampling rate of an audio signal while maintaining the original content and duration. In our study, we randomly downsampled audio within the range 11025–22050 Hz and then upsampled it back to 22050 Hz to maintain a consistent output rate. This procedure was implemented using the pedalboard library, which at the time (early 2024) applied the *WindowedSinc* interpolation algorithm by default.

Reverb. Reverb is the reflection of sound that persists after the sound source has stopped. It can be artificially simulated in audio processing to create ambiance or spatial effects. In our case, we used the `pedalboard` library and set the range [0, 0.1] for the room size parameter.

Saturation. Saturation is a distortion effect that adds warmth and richness to audio signals by intentionally overloading the digitally simulated circuitry, often mimicking vintage analog equipment. We implemented it by using `pedalboard` and set the range of the saturation level from 0 to 20.

Time shift. Time shifting consists of shifting the temporal position of an audio signal along the time axis, resulting in a time offset of the samples relative to their original position. We applied a shift randomly selected from the range of 0 to 1.5 s, which corresponds

to [0, 33075] samples at a sampling rate of 22,050 Hz. This augmentation was implemented directly in Python.

Time stretch. Time stretching is a digital audio processing technique used to adjust the duration of an audio signal without affecting its pitch. It involves stretching or compressing the audio waveform in time. We used the `librosa` implementation, setting the values of the stretch ratio in the range [0.97, 1.03].

3.2 Dataset and audio pre-processing

In our experiments, we utilized the dataset presented in [29], which is currently not publicly available. Comprising 391 original short guitar pieces performed on acoustic and classical guitars, the dataset reflects unrestricted choices in technique, expression, style, genre, harmony, and tempo [29]. The audio excerpts vary in length, with a minimum of 12.4 s to a maximum of 75.5 s. The ground truth is composed of 4 possible labels: aggressive, relaxed, happy, and sad (in this order). The classification task is assumed to be song-level MER, as the label remains fixed for the entire duration of each music excerpt [19].

The dataset creators provided two possible label configurations: the first for the intended labels (i.e., the emotion provided by the musician who composed and performed the piece), and the second provided by a listening test. In this latter case, each audio track was listened to and evaluated by 16 listeners, who gave a score in a range of 7 values (from −3 to +3) for each of the 4 emotions. For example, a value of (3, −3, 1, 0) indicates that the listener found the musical piece to be very aggressive and not relaxed at all, while perceiving the happiness and sadness of the piece as negligible. We decided to utilize these listener-provided labels, as we were interested in the more general case of emotions recognized by a listener rather than the case of the emotions intended by a composer (notably, the study reported in [29] showed that what a composer intends at the emotional level may differ from what listeners actually perceive). Moreover, the availability of the 16 annotators allowed us to have more reliable results in terms of perceived emotion. In fact, as stated in the MER survey provided in [19], *"different people may have different emotional perceptions of the same music, even the same person is also inconsistent in different times and situations"*. Therefore, as the first step, we computed the average among the 16 evaluations of each musical piece and assigned the highest value as the label for the classification task, thus casting the problem into a multi-class classification problem. Notably, four compositions were discarded as they elicited more than one emotion with the maximum value (i.e., they were characterized by an ambivalent emotion [29]).

We would like to emphasize that a multi-class approach is not the only option; rather, it is a simplification intended to keep the problem manageable. Even when excluding ambiguous songs (i.e., those with two emotions having the exact same value), a musical piece may still exhibit one dominant emotion alongside another with a high score. Therefore, a multi-label approach—where a song can be associated with multiple emotions—could provide a more representative model. The main challenge with this approach lies in defining a reasonable threshold for the emotion scores, based on average listener ratings, to decide which emotions to include in the label encoding. For example, if a threshold of 2 on a scale from −3 to 3 is used, a song with a mean score of 1.9 would be labeled as absent (0), while a song with 2.1 would be labeled as present (1), potentially causing confusion during model training. Of course, alternative strategies such as a regression approach can be employed to handle labeling more effectively; however, we chose to keep the problem simple, as the primary focus of this study is to investigate the influence of data augmentation.

Following the application of the data augmentation techniques, as elaborated later, each audio file was segmented into 3-s segments, as recommended in [29]. Subsequently, the log mel spectrogram was extracted for each sample, employing a short-time Fourier transform with a frame size of 2048 samples, a hop length of 512 samples, and 128 mel bands, which are common values employed for MIR tasks (e.g., [44]).

3.3 Deep learning model

The CNN is a very common deep learning architecture for performing MER tasks [19]. Thus, we employed a CNN for our evaluation. Specifically, the model was implemented using the `Keras` API [45] based on the `tensorflow` backend [46].

The model architecture follows designs commonly used in MIR tasks [44, 47], but we implemented a more compact CNN with approximately 33 K parameters, considering the limited size of the dataset. The main hyperparameters were also selected based on insights gained from a previous MIR task involving electric guitar samples, where data availability was similarly constrained. It is important to note that the primary objective of this work was to assess the effectiveness of data augmentation techniques in the realm of MER, rather than

Table 2 Hyperparameters of the CNN model

Hyperparameter	Value
Convolutional layers	4 (with 16 filters each)
Convolutional filters size	3×3
Convolutional stride	2
Activation function	ReLU
Max-pooling window size	3×3
Max-pooling stride	2
Dropout rate (dense layer)	0.2
Optimizer	Adam
Learning rate	0.0001
Loss function	Sparse categorical cross-entropy
Batch size	8
Number of epochs	20
Evaluation metric	Accuracy

optimizing the model architecture to obtain the best possible accuracy.

The CNN architecture comprised four convolutional layers, each followed by max-pooling and batch normalization layers. The convolutional layers employed 16 3×3 filters with a rectified linear unit (ReLU) activation function. Max-pooling layers with a 3×3 window and a stride of 2 were utilized for downsampling feature maps and reducing computational complexity. Batch normalization layers aided in stabilizing and accelerating the training process. Additionally, a dropout rate of 0.2 was applied to the only dense layer of the network (excluding the output layer) to mitigate overfitting. The model was trained using the Adam optimizer [48] with a learning rate value of 0.0001. We opted for sparse categorical cross-entropy as the loss function. The batch size was set to 8, and the model was trained for 20 epochs. Accuracy served as the default metric for evaluating the model's performance. For a summary of the model's hyperparameters, the reader is referred to Table 2.

4 Experiments and results

In our code[1], we implemented a 3-fold stratified grouped cross-validation using the implementation provided by `scikit-learn` [49]. This approach ensures a balanced number of samples per class in each fold. It also ensured that pieces performed by the same musicians were not present in both the training and validation sets, thus eliminating the so-called "artist effect" [50].

During each iteration of cross-validation, the audio files were augmented according to the augmentation technique(s) being tested. For each technique, 10 runs were performed, resulting in a total of 30 training and validation cycles. The results obtained from the 10 runs were averaged, and the standard deviation was computed.

The prediction for the entire piece was accomplished using the soft voting technique. This entails summing the prediction values associated with each of the four classes for all segments of a specific piece. Subsequently, the class (i.e., the emotion) with the highest score was selected as the model's prediction.

4.1 Single augmentation

The initial experiment involved applying a single function during the training phase. For each sample in the current training set, two samples were generated, with the parameter of the augmentation (e.g., the stretch ratio for time stretch) being randomly sampled from the specified interval (for details, the reader is referred to Table 1). Subsequently, the model was trained using both the original sample and the augmented ones. Notice that, as previously mentioned, we created only 2 samples for each original piece, according to the study reported in [8], as elaborated in Section 2.

The results are presented in Fig. 1 and Table 3. Each augmentation technique is represented by its mean value across 10 runs, along with the associated standard deviation. Notably, the effectiveness of the augmentation technique is better assessed by examining the accuracy on the segment prediction. The accuracy on the entire piece can vary based on the proportion of correctly classified samples within a particular musical excerpt. Consequently, a piece is considered misclassified even if some segments are correctly classified. However, this outcome would also occur if all segments within that piece were classified incorrectly, leading to the same outcome at the piece level, even for very different performances at the segment level.

As evident from the figure, time shift, time stretch, and pitch shift emerged as the top-performing augmentation techniques in this experimental setup. They respectively achieved a 3.6%, 2.5%, and 2.4% increase in segment-level accuracy and a 4.8%, 3.3%, and 2.9% increase at the piece level compared to the case with no augmentation. None of the other 8 techniques yielded an accuracy increase greater than 1.2% at the segment level. Interestingly, although the segment-level increase was not significant for these 8 techniques (see Section 4.3), their performances at the piece level exhibited considerable variability, as evidenced by the higher standard deviation. This confirms our previous considerations on the scarce reliability of the results at the piece level.

[1] Available at: https://github.com/michelerossi1/data_augmentation_main_code.

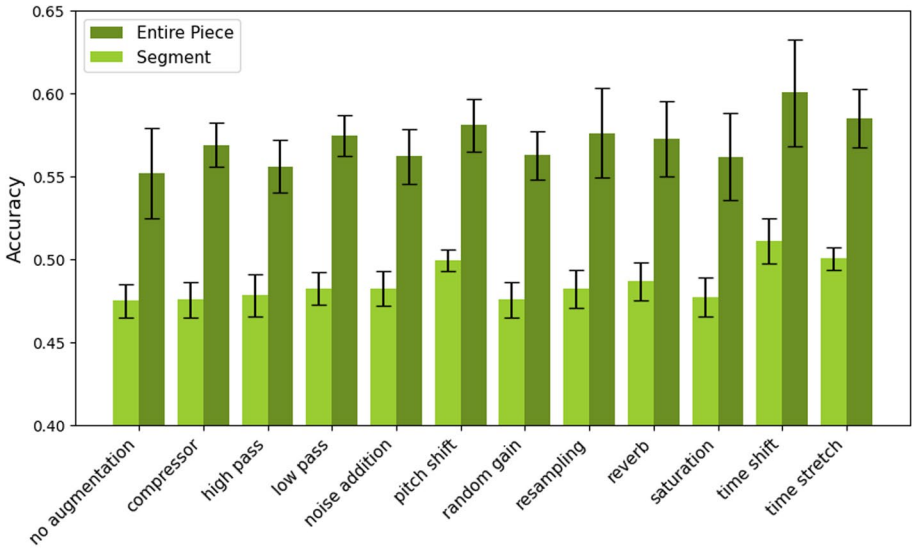

Fig. 1 Final accuracy value obtained after implementing the proposed data augmentation techniques. Both the segment-level accuracy and the accuracy for the entire piece are shown

Table 3 Final accuracy values achieved after applying the proposed data augmentation techniques

Augmentation	Segment Acc ± Std	Piece Acc ± Std	Δ Segment	Δ Piece
No augmentation	0.475 ± 0.010	0.552 ± 0.028	0.000	0.000
Compressor	0.476 ± 0.011	0.569 ± 0.014	+0.001	+0.017
High pass	0.478 ± 0.013	0.556 ± 0.016	+0.003	+0.004
Low pass	0.483 ± 0.010	0.575 ± 0.013	+0.007	+0.023
Noise addition	0.483 ± 0.010	0.562 ± 0.017	+0.007	+0.010
Pitch shift	0.499 ± 0.007	0.581 ± 0.016	+0.024	+0.029
Random gain	0.476 ± 0.011	0.563 ± 0.015	+0.001	+0.011
Resampling	0.482 ± 0.011	0.576 ± 0.027	+0.007	+0.024
Reverb	0.487 ± 0.011	0.573 ± 0.023	+0.012	+0.021
Saturation	0.477 ± 0.012	0.562 ± 0.026	+0.002	+0.010
Time shift	0.511 ± 0.014	0.600 ± 0.032	+0.036	+0.048
Time stretch	0.500 ± 0.007	0.585 ± 0.018	+0.025	+0.033

Both the segment-level accuracy and the accuracy for the entire piece are reported. The rightmost columns show the absolute increase in accuracy with respect to the baseline without augmentation

4.1.1 Experiment with a larger model

We also conducted an additional experiment aimed at assessing whether a model with a much larger number of parameters can still benefit from data augmentation. The motivation was to investigate whether the effectiveness of augmentation persists even under conditions with a substantially higher risk of overfitting. While this scenario is not directly relevant for SMI applications—where the model must remain compact—we decided to include these results here as we believe they can still provide valuable insights.

For this purpose, we implemented a model based on the well-known VGG-16 architecture, modifying it to have approximately 1 million parameters, compared to only 33,000 in our reference model. Specifically, we removed the last convolutional block, reduced the number of dense layers from two to one, decreased the number of neurons, and scaled down the convolutional filters in each block, starting with 16 filters in the first layer instead of 64.

The results show that time shift achieved the highest accuracy (50.3%), followed by reverb (50.0%) and time

stretch (49.2%), compared to 48.6% for the no-augmentation case. However, statistical analysis using the non-parametric Friedman test and the post-hoc Nemenyi test did not reveal any statistically significant differences between these best-performing augmentations and the no-augmentation baseline. This is likely due to greater variability and unpredictability in this larger-model configuration, which is inherently more prone to overfitting.

4.2 Multiple augmentation

In this section, we present results from three different strategies for applying multiple augmentations to the training set. These experiments use the three techniques that showed the best individual performance: time shift, time stretch, and pitch shift.

- **Sequential-fixed:** all three augmentations are applied sequentially to each original sample, in a fixed order.
- **Sequential-probabilistic:** each of the three augmentations is applied in sequence, but with a 50% probability of being active for each sample.
- **Single-random:** only one augmentation is applied to each sample, chosen at random with equal probability among the three techniques.

In all three settings, two augmented samples are generated from each original sample, as in the single-augmentation experiments. The *Sequential-Probabilistic* and *Single-Random* settings introduce increasing levels of stochasticity, leading to more varied augmented data. These configurations aim to evaluate whether greater sample diversity can further reduce overfitting.

Figure 2 and Table 4 present the results for this experimental setup. The first, second, and third settings exhibited accuracy increases of 2.4%, 2.5%, and 2.8% for segment prediction, respectively. For the classification of the entire piece, the accuracy increased by 2.5%, 3.1%, and 4.0%, respectively. On the other hand, we did not observe statistically significant differences among the three proposed settings, as described in the next section.

4.3 Statistical tests

In this section, we present the results of the statistical tests performed for all the configurations, including

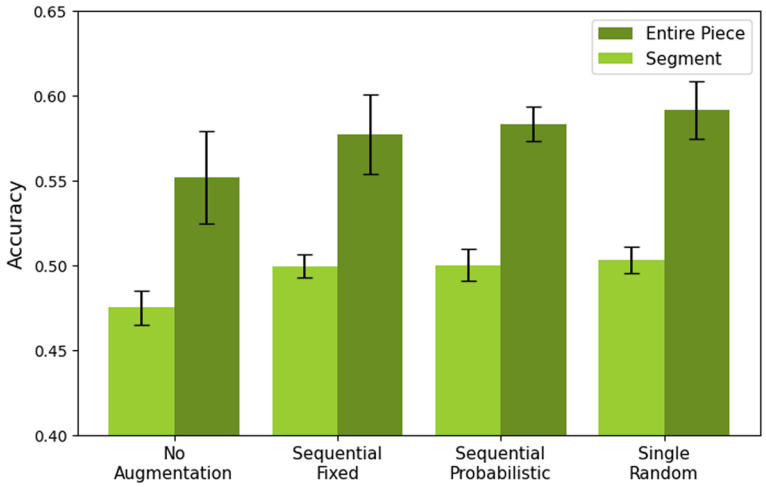

Fig. 2 Final accuracy values obtained following the implementation of the proposed three combinations of data augmentation techniques. Both the segment-level accuracy and the accuracy for the entire piece are shown

Table 4 Final accuracy values achieved after applying the proposed three combinations of data augmentation techniques

Augmentation	Segment Acc ± Std	Piece Acc ± Std	Δ Segment	Δ Piece
No augmentation	0.475 ± 0.010	0.552 ± 0.028	0.000	0.000
Sequential-fixed	0.499 ± 0.007	0.577 ± 0.024	+0.024	+0.025
Sequential-probabilistic	0.500 ± 0.009	0.583 ± 0.010	+0.025	+0.031
Single-random	0.503 ± 0.008	0.592 ± 0.017	+0.028	+0.040

Both the segment-level accuracy and the accuracy for the entire piece are reported. The rightmost columns show the absolute increase in accuracy compared to the baseline without augmentation

both single augmentations and multiple configurations. Using the Python package `autorank` [51], we obtained the following results for segment-level accuracy. As mentioned earlier, we focus indeed on segment accuracy as it presents more consistent results. Obviously, a higher value in the segment accuracy increases the probability of having a higher accuracy at the segment level (which is computed via soft voting, as stated before). Table 5 presents the results based on the following indices:

- **M (mean)**: mean value of each population (i.e., 10 executions);
- **SD (standard deviation)**: standard deviation within each population;
- **CI (confidence interval)**: range within which the true population parameter (i.e., the mean) is estimated with a certain level of confidence;
- **d (effect size)**: Cohen's *d* value, which is a standardized measure of the difference between two populations' means;
- **Magnitude**: interpretation of effect size magnitude as negligible, small, medium, or large, indicating practical significance.

Table 5 Statistical indices of single and multiple augmentation settings under evaluation

Augmentation	M	SD	CI	d	Magnitude
Compressor	0.476	0.011	[0.467, 0.484]	–	Negligible
No augmentation	0.475	0.011	[0.467, 0.483]	0.057	Negligible
Random gain	0.476	0.011	[0.467, 0.484]	0.004	Negligible
High pass	0.478	0.013	[0.470, 0.486]	−0.198	Negligible
Saturation	0.477	0.012	[0.469, 0.486]	−0.127	Negligible
Resampling	0.482	0.012	[0.474, 0.490]	−0.549	Medium
Noise addition	0.482	0.011	[0.474, 0.491]	−0.607	Medium
Low pass	0.483	0.010	[0.474, 0.491]	−0.627	Medium
Reverb	0.487	0.012	[0.478, 0.495]	−0.938	Large
Pitch shift	0.499	0.007	[0.491, 0.508]	−2.490	Large
Multiple (second)	0.500	0.010	[0.492, 0.508]	−2.263	Large
Multiple (first)	0.499	0.007	[0.491, 0.508]	−2.482	Large
Time stretch	0.500	0.007	[0.492, 0.509]	−2.595	Large
Multiple (third)	0.503	0.008	[0.495, 0.511]	−2.754	Large
Time shift	0.511	0.014	[0.503, 0.520]	−2.727	Large

In Fig. 3, we present the plot depicting the 95% confidence interval of the mean values, considering both single and multiple augmentations. From the plot, it is evident that time shift, time stretch, pitch shift, and the three multiple augmentation settings were the most effective. Notably, these cases show no overlap in the confidence interval with the no-augmentation setting.

The statistical analysis involved examining 15 populations (11 for the single augmentation settings, 4 for the multiple augmentation settings, and 1 for the case with no augmentation), each comprising 10 paired samples, with a significance level set at $\alpha = 0.05$. Initial tests failed to reject the null hypothesis for normality ($p = 0.067$) and homoscedasticity ($p = 0.596$), indicating normal distribution and homogeneity within the data. Subsequently, a repeated-measures ANOVA was employed due

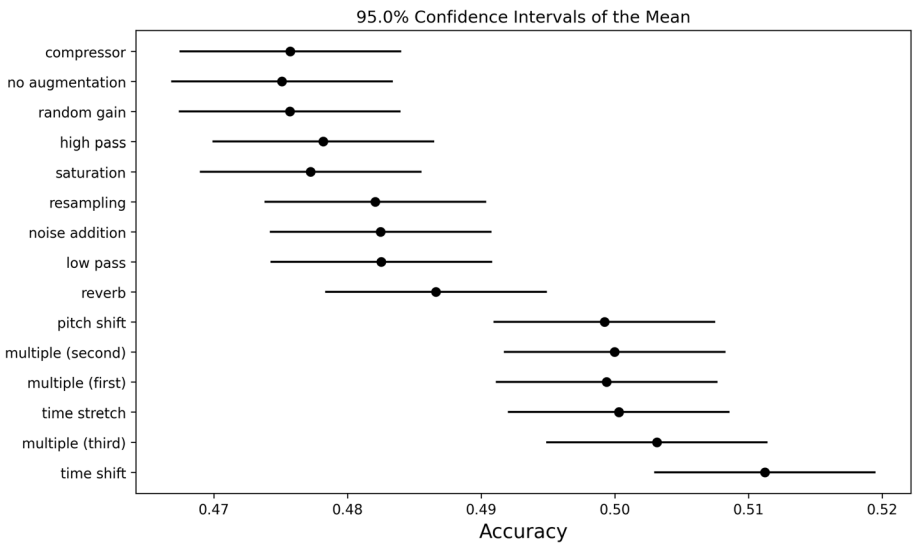

Fig. 3 Mean accuracy value for each augmentation technique with the associated 95% confidence intervals, i.e., the range of values within which we are 95% confident the true parameter lies. Two populations (i.e., augmentation techniques) are significantly different if their confidence intervals do not overlap

to the multiple populations, revealing statistically significant differences in mean values. Based on the post-hoc Tukey HSD test, it resulted that there were no significant differences within the following groups: (1) compressor, no augmentation, random gain, high pass, saturation, resampling, noise addition, and low pass; (2) resampling, noise addition, low pass, and reverb; (3) pitch shift, multiple (second), multiple (first), time stretch, and multiple (third); (4) multiple (third) and time shift. All other differences were statistically significant. Additionally, note that Cohen's d values were computed using the population with the highest rank value (the compressor, in this specific case) as the reference population.

5 Listening test

We conducted a listening test to assess whether the perceived emotional label of a sample was preserved after applying data augmentation techniques. The goal was to ensure that the primary emotion of a piece (among the four considered) remained dominant and was not overtaken by another emotion. The subsequent sections provide details about the experimental setup and the results obtained.

5.1 Listening test setup

Following the approach proposed in [29], we conducted a listening test with 16 participants, all of whom were musicians, as the cited study showed that musicians are better able to assign emotions to musical pieces due to their emotional perception more closely aligning with the composer's intended emotion. To ensure high-quality audio playback, participants were asked to use an external audio interface and high-quality headphones while taking the listening test via the web application presented later.

We selected eight musical pieces from the dataset described in Section 3.2, including two examples for each of the four emotions: aggressive, relaxed, happy, and sad. From each audio file, we generated 11 transformed versions (data augmentations) by applying the most extreme values within the range defined in our study. For pitch shift and time stretch, both the maximum and minimum values in the range were used (e.g., -3 and $+3$ semitones for pitch shift). Random gain and time shift were excluded from this phase: random gain only introduces minor volume changes, and time shift was primarily employed in our work to analyze different segments of the same excerpts, whereas here we focused on a piece-level analysis. This approach yielded a total of 96 audio samples (8 original pieces and 8×11 augmented versions).

To facilitate the test, we developed an open-source Python web application. For each music excerpt, participants were asked to rate the perceived levels of aggressiveness, relaxation, happiness, and sadness on a 7-point scale ranging from -3 to $+3$, as shown in the screenshot provided in Fig. 4.

The order of the audio samples was randomized for each participant. The test was divided into two parts to minimize attention fatigue. Participants could listen to each musical piece as many times as needed before providing their evaluation. On average, participants took 45 min to complete the test.

5.2 Listening test results

For each of the 8 musical pieces analyzed in the listening test, we calculated the mean and standard deviation for each of the 4 emotions, resulting in a total of 32 combinations. Figure 5 illustrates how the 4 emotions vary in one of the 8 pieces, which was initially classified as aggressive. Table 6 presents the mean and standard deviation values for the dominant emotion of each of the 8 pieces included in the test.

As observed in Fig. 5, in this specific case, the main emotion remains unchanged when data augmentation techniques are applied. However, slight variations can be noticed in some of the emotion values.

To verify whether the value of the main emotion changed significantly when data augmentation was applied compared to the perception of the original recording, we conducted a Friedman test, with the Nemenyi post-hoc test, for the four emotions across each of the eight pieces. Results showed that the median value of the main emotion for each piece did not change significantly across the different data augmentation techniques.

Additionally, we performed the same statistical test for the non-primary emotions of each piece, which correspond to the three emotions other than the true label for the piece (e.g., relaxation, happiness, and sadness for a piece classified as aggressive), resulting in a total of 24 combinations—three emotions for each of the eight pieces. In this case, we found a statistically significant difference in only 2 out of 24 combinations. The first combination was for a happy piece, where the emotion "relaxation" exhibited slight changes across the 12 populations (i.e., the original piece and 11 augmented versions), resulting in two distinct groups of augmentations with no statistical difference within each group. These two groups were nearly identical, differing only in the inclusion of low-pass and high-pass filtering. The second case was for a sad piece, where the emotion "aggressiveness" showed statistical significance. Similarly, the statistical test identified two groups, with all augmentations being shared except for noise addition, pitch shift, and high-pass filtering.

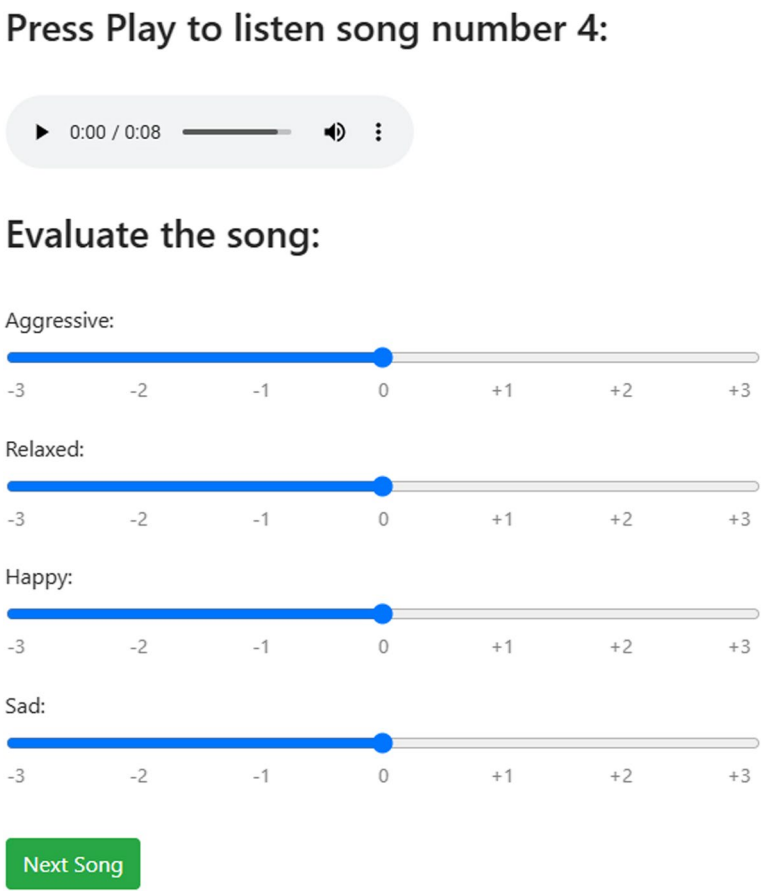

Fig. 4 Screenshot of the Python web application we developed for the listening test. Users are asked to evaluate each piece by rating the intensity of four emotions on a scale from −3 to +3, based on their perception

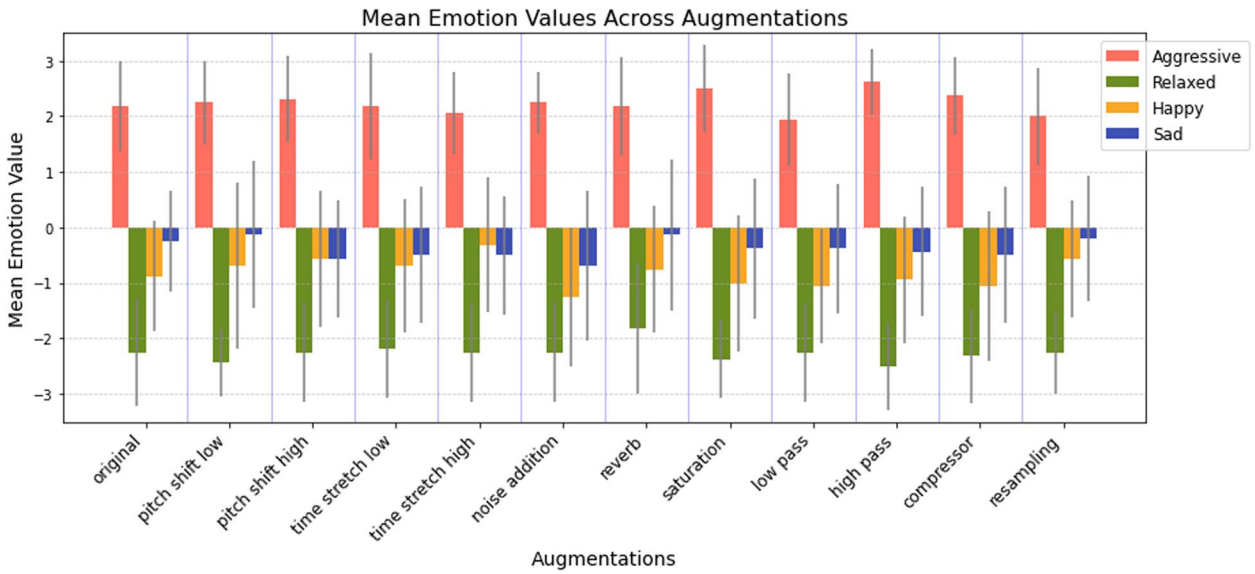

Fig. 5 Mean and standard deviation calculated across 16 users, for each data augmentation method, for the four emotions analyzed: aggressiveness, relaxation, happiness, and sadness

Table 6 Mean and standard deviation, for each data augmentation method, of the primary emotion across the 8 songs included in the listening test

Augmentation Technique	Piece 1 (aggressive)	Piece 2 (aggressive)	Piece 3 (relaxed)	Piece 4 (relaxed)	Piece 5 (happy)	Piece 6 (happy)	Piece 7 (sad)	Piece 8 (sad)
Original	2.19 ± 0.81	2.06 ± 0.83	1.75 ± 1.52	1.69 ± 0.98	1.81 ± 0.95	2.56 ± 0.70	1.25 ± 1.20	1.88 ± 0.93
Pitch shift low	2.25 ± 0.75	2.06 ± 1.03	2.12 ± 1.11	1.44 ± 1.37	1.75 ± 1.25	2.56 ± 0.61	1.44 ± 1.12	2.00 ± 0.71
Pitch shift high	2.31 ± 0.77	2.19 ± 0.81	1.81 ± 1.24	1.44 ± 1.46	1.88 ± 0.86	2.50 ± 0.87	1.50 ± 0.87	1.94 ± 0.75
Time stretch low	2.19 ± 0.95	1.81 ± 0.95	1.94 ± 1.30	1.69 ± 1.10	2.06 ± 0.90	2.50 ± 0.71	1.12 ± 1.32	2.12 ± 0.70
Time stretch high	2.06 ± 0.75	2.00 ± 0.94	1.62 ± 1.36	1.75 ± 1.03	1.94 ± 0.75	2.44 ± 0.70	0.94 ± 1.09	1.81 ± 0.73
Noise addition	2.25 ± 0.56	2.00 ± 0.79	1.44 ± 1.69	1.12 ± 1.41	1.75 ± 0.66	2.25 ± 0.90	1.31 ± 1.40	1.81 ± 0.73
Reverb	2.19 ± 0.88	2.06 ± 0.90	1.94 ± 1.34	1.69 ± 0.92	1.94 ± 0.75	2.31 ± 0.77	0.81 ± 1.38	1.94 ± 0.90
Saturation	2.50 ± 0.79	2.00 ± 0.71	1.00 ± 1.27	1.69 ± 0.85	1.81 ± 0.73	2.06 ± 0.90	1.38 ± 1.05	1.88 ± 0.60
Low pass	1.94 ± 0.83	2.00 ± 1.06	2.06 ± 0.90	2.00 ± 1.12	1.50 ± 1.12	2.12 ± 0.86	1.12 ± 1.32	2.25 ± 0.66
High pass	2.62 ± 0.60	2.19 ± 0.95	1.69 ± 1.65	1.38 ± 1.45	1.81 ± 1.01	2.62 ± 0.60	1.00 ± 1.54	1.75 ± 0.75
Compressor	2.38 ± 0.70	2.25 ± 0.66	1.50 ± 1.41	1.69 ± 0.85	1.81 ± 0.88	2.56 ± 0.61	1.06 ± 1.60	1.94 ± 0.75
Resampling	2.00 ± 0.87	1.88 ± 0.93	1.50 ± 1.32	1.62 ± 1.41	1.62 ± 1.17	2.25 ± 0.75	1.25 ± 1.09	1.94 ± 0.75

From this listening test, it is possible to conclude that even when applying data augmentation techniques with the strongest values within the range considered in our study (see Table 1), the perceived emotion generally does not change significantly.

As it was mentioned earlier, it is important to consider that the primary concern in the context of our experiment — using data augmentation to artificially expand the dataset — is to ensure that the main emotion of a piece (among the four considered) is not surpassed by another emotion. This is crucial because our problem is framed as a multi-class classification task, where each piece is assigned a single label (i.e., emotion).

This consideration also implies that if a piece's classification is ambiguous *per se* (i.e., without data augmentation), minor fluctuations introduced by data augmentation techniques could become problematic. Such fluctuations might increase ambiguity in defining the piece's main emotion. However, this problem is more related to the initial ambiguity of some musical pieces than it is to the data augmentation itself, and it can be solved by using only pieces that are strongly defined for a specific emotion.

Based on the results of this listening test, we can generally conclude that the data augmentation techniques can be applied within the range presented in Table 1 without significant issues related to changes in the label. Therefore, in the following discussion, we will take this observation as a given and focus on analyzing the results and considerations related to our original problem.

6 Discussion

The statistical analysis conducted on the differences in mean accuracy scores underscored the substantial benefits of employing certain augmentation techniques for MER tasks when utilizing a deep learning model. The statistical analysis revealed that compressor, random gain, high pass, and saturation produced negligible effects and were not statistically significant. Resampling, noise addition, and low pass resulted in moderate increases in accuracy (with medium effect sizes), but these changes were also not statistically significant, as their confidence intervals overlapped with that of the baseline. In contrast, reverb, pitch shift, time stretch, and time shift produced statistically significant improvements, all associated with large effect sizes (See Table 5 for details). Notably, the specific method used to combine the three augmentations in the multiple augmentation settings did not yield considerable improvements compared to the single augmentation settings. Moreover, by inspecting Fig. 3, it becomes apparent that the combination of multiple augmentations did not significantly improve the accuracy obtained by the best-performing augmentations applied individually. Instead, a single technique (time shift) provided the overall best accuracy at the segment level.

The fact that time shift emerged as the best augmentation technique can be analyzed from the following perspective. Considering the range of shifts applied (from 0 to 1.5 s), this technique is akin to obtaining overlapping segments from the original piece, thereby increasing the number of reliable samples for the dataset. For instance, if the shift value were consistently set to 1.5 s, this would generate a training set where the

segments are derived from the entire pieces with a 50% overlap across consecutive samples (noting that the augmentation stage precedes segmentation). Consequently, further analysis should be conducted to determine if this technique yields high performance even when segments are extracted with fixed overlaps (e.g., 50% or 75%) by design.

In line with findings from previous studies on dataset augmentation for MIR tasks, such as genre classification, singing voice detection, and instrument recognition, we noted that pitch shift significantly increased the overall model accuracy (e.g., [41]). Additionally, the high performance achieved by the time stretch technique was also in line with the findings previously reported in [8].

While our analysis is based on a single model, future work could explore the effect of augmentation techniques across different architectures—such as deeper CNNs, residual networks, or transformer-based models—to obtain more general conclusions for MER tasks. Implementing models with varying numbers of parameters could provide valuable insights. Such experiments would also help validate what was stated in [11], where the authors emphasized the importance of utilizing both augmentation techniques and high-capacity deep learning models to maximize system performance.

Another limitation of our research is that we restricted the creation of augmented samples to only two samples per original sample. While this decision was in line with the findings of [8], which we discussed earlier, it is important to recall that that study primarily employed traditional machine learning techniques, rather than deep learning as in our case.

Furthermore, our research primarily focused on overall accuracy, without delving into the specific effects of each augmentation technique on individual classes. This aspect could be valuable, particularly in the context of creating class-conditional data augmentations, as suggested in [11].

Another aspect worth exploring in future work is the specific tools or implementations used for each data augmentation technique. For instance, in the case of reverb, various types of responses (e.g., room, plate, hall) could be analyzed. Similarly, different kinds of background noise can be added in the case of noise addition. Therefore, a more in-depth analysis considering the best-performing techniques identified in the previous stage could provide valuable insights.

Finally, the application of these augmentations to larger pre-trained models using transfer learning or exploring other more recent paradigms, such as transformers, represents a promising direction for future research. These approaches could potentially enhance model performance and robustness in MER tasks.

7 Conclusions

This paper presented an analysis of the effectiveness of 11 data augmentation techniques for the task of music emotion recognition using a convolutional neural network model. We explored the impact of dataset augmentation when a single transformation was applied, as well as when multiple transformations were applied simultaneously.

In the first scenario, where only one augmentation technique was applied, we found that time shift, time stretch, and pitch shift were the most effective techniques in increasing the classification accuracy. In the second scenario (multiple augmentations), we investigated three different settings for applying multiple augmentation techniques during training. These settings involved (1) applying all augmentations to each augmented sample in cascade, (2) assigning a 50% probability for each augmentation to be applied, and (3) selecting one of the three augmentation techniques randomly for each augmented sample. We did not observe significant differences in performance among these three techniques, although all of them resulted in an accuracy increase of at least 2.4% compared to the case without augmentation.

To ensure that the main label is perceptually preserved when applying the data augmentation techniques, we conducted a listening test. In this test, we verified that the primary emotion associated with each piece does not change significantly with the application of data augmentation techniques.

In conclusion, we demonstrated that appropriate audio data augmentation can lead to statistically significant improvements in the classification accuracy of deep learning models in the task of music emotion recognition. This improvement holds great promise for the development of emotionally aware SMIs, which can respond to the emotional content of music in real-time, providing musicians with more intuitive and expressive tools. We hope this work will inspire further contributions, promoting more research in both MER and its applications in SMIs.

Abbreviations

MIR	Music information retrieval
MER	Music emotion recognition
SMIs	Smart musical instruments
CNN	Convolutional neural network

Acknowledgements

The authors sincerely thank all participants of the listening test for their time and valuable contributions to this study.

Authors' contributions

The concept for this work was developed and approved by all three authors. MR wrote the code for the paper, conducted the listening tests, and drafted the majority of the manuscript. GI and LT provided valuable guidance throughout the process. They also reviewed the entire paper, offering revisions and clarifications for the final version of the manuscript.

Funding
Not applicable

Data availability
The main code used for creating the augmentation and to evaluate them is available at https://github.com/michelerossi1/data_augmentation_main_code, while the code used to implement the listening test web app can be found here: https://github.com/michelerossi1/listening_test_webAPP.

Declarations

Competing interests
LT is a guest editor for the current special issue: Signal Processing for the Internet of Sounds. All other authors declare that they have no competing interests.

Received: 5 February 2025 Accepted: 24 September 2025
Published online: 17 November 2025

References

1. L. Turchet, Smart musical instruments: vision, design principles, and future directions. IEEE Access **7**, 8944–8963 (2019)
2. X. Yang, Y. Dong, J. Li, Review of data features-based music emotion recognition methods. Multimed. Syst. **24**, 365–389 (2018)
3. L. Turchet, D. Stefani, J. Pauwels, Musician-ai partnership mediated by emotionally-aware smart musical instruments. Int. J. Hum.-Comput. Stud. **191**, 103340 (2024)
4. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**(7553), 436–444 (2015)
5. C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
6. S.Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy. A survey of data augmentation approaches for NLP. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, Online, pp. 968–988 (2021)
7. S. Wei, S. Zou, F. Liao et al., in *Journal of physics: Conference series*. A comparison on data augmentation methods based on deep learning for audio classification, vol. 1453 (IOP Publishing, Bristol, 2020), p. 012085
8. R. Mignot, G. Peeters, An analysis of the effect of data augmentation methods: experiments for a musical genre classification task. Trans. Int Soc. Music. Inf. Retr. **2**(1), 97–110 (2019)
9. T. Fukuda, R. Fernandez, A. Rosenberg, S. Thomas, B. Ramabhadran, A. Sorin, G. Kurata, in *Interspeech*. Data augmentation improves recognition of foreign accented speech (ISCA, Hyderabad, 2018), September, pp. 2409–2413
10. X. Cui, V. Goel, B. Kingsbury, Data augmentation for deep neural network acoustic modeling. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(9), 1469–1477 (2015)
11. J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process. Lett. **24**(3), 279–283 (2017)
12. N. Davis, K. Suresh, in *IEEE Recent Advances in Intelligent Computational Systems)*. Environmental sound classification using deep convolutional neural networks and data augmentation (IEEE, New York, 2018), pp. 41–45
13. W. Bian, J. Wang, B. Zhuang, J. Yang, S. Wang, J. Xiao, in *Pacific Rim International Conference on Artificial Intelligence*. Audio-based music classification with DenseNet and data augmentation (Springer, Cham, 2019), pp. 56–65
14. A. Ramires, X. Serra. Data augmentation for instrument classification robust to audio effects. Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19), Birmingham, UK, pp. 1–9 (2019)
15. R. Shankar, A.H. Kenfack, A. Somayazulu, A. Venkataraman. A comparative study of data augmentation techniques for deep learning based emotion recognition. arXiv preprint arXiv:2211.05047 (2022). https://doi.org/10.48550/arXiv.2211.05047
16. R. Panda, R. Malheiro, R.P. Paiva, Novel audio features for music emotion recognition. IEEE Trans. Affect. Comput. **11**(4), 614–626 (2018)
17. R. Panda, R. Malheiro, R.P. Paiva, Audio features for music emotion recognition: a survey. IEEE Trans. Affect. Comput. **14**(1), 68–88 (2020)
18. Y.E. Kim, E.M. Schmidt, R. Migneco, B.G. Morton, P. Richardson, J. Scott et al., in *International Society for Music Information Retrieval Conference*. Music emotion recognition: A state of the art review, vol. 86 (ISMIR, Utrecth, 2010), pp. 937–952
19. D. Han, Y. Kong, J. Han, G. Wang, A survey of music emotion recognition. Front. Comput. Sci. **16**(6), 166335 (2022)
20. X. Jiang, Y. Zhang, G. Lin, L. Yu, Music emotion recognition based on deep learning: A review. IEEE Access 12, 157716–157745 (2024)
21. S. Deng, D. Wang, X. Li, G. Xu, Exploring user emotion in microblogs for music recommendation. Expert Syst. Appl. **42**(23), 9284–9293 (2015)
22. L. Bunt, M. Pavlicevic, in *Music and emotion: Theory and research*. Music and emotion: Perspectives from music therapy (Oxford University Press, Oxford, 2001)
23. H. Jalonen, in *Federated Conference on Computer Science and Information Systems*. Social media and emotions in organisational knowledge creation (IEEE, New York, 2014), pp. 1371–1379
24. M. Huang, W. Rong, T. Arjannikov, N. Jiang, Z. Xiong, in *Artificial Neural Networks and Machine Learning*. Bi-modal deep Boltzmann machine based musical emotion classification (Springer, Cham, 2016), pp. 199–207
25. X. Liu, Q. Chen, X. Wu, Y. Liu, Y. Liu. CNN based music emotion classification. arXiv preprint arXiv:1704.05665 (2017). https://doi.org/10.48550/arXiv.1704.05665
26. R. Sarkar, S. Choudhury, S. Dutta, A. Roy, S.K. Saha, Recognition of emotion in music based on deep convolutional neural network. Multimed. Tools Appl. **79**(1), 765–783 (2020)
27. P.T. Yang, S.M. Kuang, C.C. Wu, J.L. Hsu, in *International Conference on Human-Computer Interaction*. Predicting music emotion by using convolutional neural network (Springer, Cham, 2020), pp. 266–275
28. P.L. Louro, H. Redinho, R. Malheiro, R.P. Paiva, R. Panda, A comparison study of deep learning methodologies for music emotion recognition. Sensors **24**(7), 2201 (2024)
29. L. Turchet, J. Pauwels, Music emotion recognition: intention of composers-performers versus perception of musicians, non-musicians, and listening machines. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 305–316 (2021)
30. S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, F. Shen. Image data augmentation for deep learning: A survey. arXiv preprint arXiv:2204.08610 (2022). https://doi.org/10.48550/arXiv.2204.08610
31. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12, pp. 1097–1105.
32. S. Grollmisch, E. Cano, Improving semi-supervised learning for audio classification with fixmatch. Electronics **10**(15), (2021). https://doi.org/10.3390/electronics10151807
33. D.S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E.D. Cubuk, Q.V. Le. SpecAugment: a simple data augmentation method for automatic speech recognition. Proceedings of Interspeech 2019, pp. 2613–2617 (2019). https://doi.org/10.21437/Interspeech.2019-2680
34. J. Wang, S. Kim, Y. Lee, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Speech augmentation using wavenet in speech recognition (IEEE, New York, 2019), pp. 6770–6774
35. E. Tsunoo, K. Shibata, C. Narisetty, Y. Kashiwagi, S. Watanabe. Data augmentation methods for end-to-end speech recognition on distant-talk scenarios. Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH 2021), pp. 301–305 (2021). https://doi.org/10.21437/Interspeech.2021-958
36. N. Jaitly, G.E. Hinton, in *ICML Workshop on Deep Learning for Audio, Speech and Language*. Vocal tract length perturbation (vtlp) improves speech recognition, vol. 117 (ICML, Atlanta, 2013), p. 21
37. K.J. Piczak, in *IEEE International Workshop on Machine Learning for Signal Processing*. Environmental sound classification with convolutional neural networks (IEEE, New York, 2015), pp. 1–6
38. G. Parascandolo, H. Huttunen, T. Virtanen, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Recurrent neural networks for polyphonic sound event detection in real life recordings (IEEE, New York, 2016), pp. 6440–6444

39. J. Schlüter, T. Grill, in *International Society for Music Information Retrieval Conference*. Exploring data augmentation for improved singing voice detection with neural networks (ISMIR, Malaga, 2015)
40. B. McFee, E.J. Humphrey, J.P. Bello, in *International Society for Music Information Retrieval Conference*. A software framework for musical data augmentation (ISMIR, Malaga, 2015)
41. R.L. Aguiar, Y.M. Costa, C.N. Silla, in *International Joint Conference on Neural Networks*. Exploring data augmentation to improve music genre classification with convnets (IEEE, New York, 2018), pp. 1–8
42. P. Sobot. Pedalboard. (2021). https://doi.org/10.5281/zenodo.7817838
43. B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, O. Nieto, in *Python in Science Conference*. librosa: Audio and music signal analysis in Python (SciPy.org, Austin, 2015), pp. 18–24
44. M. Comunità, D. Stowell, J.D. Reiss. Guitar effects recognition and parameter estimation with convolutional neural networks. Journal of the Audio Engineering Society 69(7/8), 594–604 (2021). https://doi.org/10.17743/jaes.2021.0019
45. F. Chollet et al. Keras. (2015). https://github.com/fchollet/keras
46. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen et al., TensorFlow: Large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org
47. J. Pons, O. Nieto, M. Prockup, E.M. Schmidt, A.F. Ehmann, X. Serra. End-to-end learning for music audio tagging at scale. Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018), Paris, France, pp. 1–9 (2018). International Society for Music Information Retrieval, Canada. ISBN: 978-2-9540351-2-3
48. D.P. Kingma, J. Ba. Adam: a method for stochastic optimization. Proceedings of the 3rd International Conference for Learning Representations (ICLR) (2015). https://arxiv.org/abs/1412.6980
49. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
50. A. Flexer, A closer look on artist filters for musical genre classification. World **19**(122), 16–17 (2007)
51. S. Herbold, Autorank: a python package for automated ranking of classifiers. J. Open Source Softw. **5**(48), 2173 (2020)

Publisher's Note