An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task

Rémi Mignot* and Geoffroy Peeters†

Supervised machine learning relies on the accessibility of large datasets of annotated data. This is essential since small datasets generally lead to overfitting when training high-dimensional machine-learning models. Since the manual annotation of such large datasets is a long, tedious and expensive process, another possibility is to artificially increase the size of the dataset. This is known as data augmentation. In this paper we provide an in-depth analysis of two data augmentation methods: sound transformations and sound segmentation. The first transforms a music track to a set of new music tracks by applying processes such as pitch-shifting, time-stretching or filtering. The second one splits a long sound signal into a set of shorter time segments. We study the effect of these two techniques (and the parameters of those) for a genre classification task using public datasets. The main contribution of this work is to detail by experimentation the benefit of these methods, used alone or together, during training and/or testing. We also demonstrate their use in improving the robustness of potentially unknown sound degradations. By analyzing these results, good practice recommendations are provided.

Keywords: Data Augmentation; Supervised training; Datasets; Musical genre classification

1. Introduction

A common task in Music Information Retrieval (MIR) is the prediction of metadata based on the music signal content itself, e.g. in audio classification, musical structure segmentation, tempo prediction, fundamental frequency estimation. Whereas some of the methods are based on known properties which can be directly evaluated using dedicated algorithms, other techniques need a number of annotated examples to enable automatic learning of the discriminant characteristics which help to solve the given problem: this is called supervised training.

One of the problems of these approaches, for practical reasons, is that the creation of such a dataset is long and tedious. For example, many audio classification tasks ideally need some hundreds of annotated songs per class to achieve a good estimation.

The use of too small or unrepresentative datasets usually leads to overfitting with prediction methods which have a high level of complexity. In this case, the trained models may focus on sound properties which discriminate the few given examples, but which are irrelevant in a general way.

This phenomenon may also appear if all the given examples present a common characteristic which is not representative of real-world examples. For example, this can be met if all the song files of the training set are encoded using the same format (e.g. MP3, and same bitrate and codec). In this case, the prediction of a sound file encoded with a different format may fail.

When only few examples can be manually annotated, we investigate the use of Data Augmentation which allows to artificially increase the size of the training dataset. This approach has already been used in many different fields, including Image Recognition and Music Information Retrieval, and it generally provides significant improvements.

In this paper, we provide an in-depth analysis of two data augmentation methods: sound transformations and sound segmentation. The first one relies on the computation of different types of transformations applied to the sounds, and the second one splits long sound signals into several shorter time segments. Based on experiments on genre classification tasks, the main contribution of this work is to detail the effect of those methods. Analysing the results for some different parameter settings, good practice recommendations are then provided. Additionally, the robustness to potentially unknown sound degradations is studied.

This paper is organised as follows: Section 2 presents the known task of genre classification (which is the task we use for testing data augmentation). The application of data augmentation to sound signals is detailed in Sec. 3.
Section 4 presents the transformation process which is used in the following experiments. The implementation of the experiments is then presented in Sec. 5 (datasets, audio features, classification algorithms). The experiments are detailed in Sec. 6 and the results are discussed. Finally, Section 7 concludes this paper.

2. Audio Classification in Music Genre

Audio classification is one of the most common tasks in MIR, and it consists of predicting the class of an input song. For musical applications, the classification can be made by: emotion (Feng et al., 2003), dance type (Marchand and Peeters, 2014), or musical genre (Tzanetakis and Cook, 2002). The large majority of the approaches is based on supervised methods which require an annotated training set.

As detailed by Fu et al. (2011), the usual approaches begin with a frame-by-frame extraction of timbral descriptors such as: MFCCs (Davis and Mermelstein, 1980), spectral moments (centroid, spread, skewness or kurtosis), spectral flatness, noisiness (Peeters et al., 2011). Mid-term descriptors can be then derived from those by applying mathematical operators over a few seconds. See for example the Rhythm Patterns proposed by Lidy et al. (2007). Finally, a summary can represent the whole signal by one vector, e.g.: averages of descriptor components in time, variances, covariances, medians.

To reduce the complexity or to focus the sound descriptions to a relevant sub-space, a dimensionality reduction is usually done. For example, Principal Component Analysis (PCA; Hotelling, 1933), when applied to normalised components, usually catches pertinent information, and the supervised Linear Discriminant Analysis (LDA; Duda et al., 2001) selects the few directions which maximise the Fisher information.

Finally, standard supervised classification algorithms are usually used to predict the class of a given vector, for example: Gaussian Mixture Models (GMM; Duda et al., 2001), Support Vector Machines (SVM; Boser et al., 1992), or k-Nearest Neighbour (k-NN; Cover and Hart, 1967).

During training, the dataset is used to provide a collection of descriptors representing all classes in order to train the models. During testing, the same descriptors and the trained models are used to predict the classes of the test signals.

The above description corresponds to the simplest approaches, but other approaches exist. For example, the results of a first stage of classifiers can be fused to a final classifier (cf. e.g. Ness et al., 2009). Also, high-level descriptors relating to the beat, pitch and chord can be added to low-level features. Finally, note that Artificial Neural Networks (shallow or deep) can also be used for classification. In recent years, Deep Learning has also been proposed to simultaneously learn relevant features of spectrograms, and to classify the elements, using Convolutional or Recurrent Neural Network (CNN or RNN; cf. e.g. Bishop, 1995; Lee et al., 2009b).

The purpose of this paper is to demonstrate the benefit of data augmentation and not to achieve the highest possible recognition rate for musical genre classification. Consequently, for the sake of simplicity, generality and reproducibility, the selected methods follow the simplest approach described above, together with well-known techniques from the literature. See Sec. 5.2 for more details.

3. Data Augmentation for Audio

The main idea of data augmentation is to artificially increase the size of the training set using any possible techniques which preserve the classes. This is especially useful with small annotated datasets and complex trainable algorithms, to avoid overfitting.

3.1 An overview of Data Augmentation

In image recognition, a common method is to transform the original training images by some deformations which do not change the class. This idea has been used early in research by Yaeger et al. (1997) and LeCun et al. (1998) for handwritten character recognition tasks where the training characters were modified by skew, rotation, and scaling; and later by Simard et al. (2003) with a more complex elastic distortion. More recently, Krizhevsky et al. (2012) artificially augmented a training set for image classification using: translation, horizontal reflection, and colour alterations.

For speech recognition, the early work of Chang and Lippmann (1995) used speaker modification techniques (Quatieri and McAulay, 1992) to increase the size of the training set for a speaker independent task of keyword spotting. More recent works (such as Jain and Hinton, 2013; Kanda et al., 2013; Ragni et al., 2014; Cui et al., 2015) also used data augmentation for Automatic Speech Recognition. These works mainly take advantage of special transformations for speech such as vocal tract length perturbation.

In Music Information Retrieval, data augmentation has been also widely used for some years: for genre classification (Li and Chan, 2011), for chord recognition (Lee and Slaney, 2008; Humphrey and Bello, 2012), for polyphonic music transcription (Kirchhoff et al., 2012) and in singing voice detection (Schlüter and Grill, 2015). Moreover dedicated software is proposed by McFee et al. (2015) which is tested for instrument recognition, and the Audio Degradation Toolbox has been developed and tested for the robustness of different tasks in MIR (Mauch and Ewert, 2013).

For example, Li and Chan (2011) noticed that the MFCC features are influenced by the fundamental frequencies of instruments. It has also been observed that in the dataset GTZAN of Tzanetakis and Cook (2002), there is a strong correlation between the genre (classes) and the musical keys of the songs. Then, the trained models may implicitly focus on the dominant key of the genres, which should be avoided. Finally, using transpositions for the training examples, the algorithm becomes invariant to musical key, and the classification is improved.

Finally, we note that even if the term ‘Data Augmentation’ is mainly associated with data transformations, in this paper we consider it with a more general point of view. In that sense, time splitting/segmentation of sounds is also seen as a data augmentation technique, cf. Sec. 3.3.
3.2 Sound transformations

As presented above, the first way to obtain new examples from the original ones is to transform them. In the work of Schlüter and Grill (2015), the sound modifications are done directly on the modulus of the spectrogram, considered as an image, before the use of CNN. For a more general use for audio applications, in this paper, the sound transformations lead to new audio signals in the time domain, which can then be processed by any algorithm, like the original ones.

We implemented a lot of different elementary transformations, such as: filtering, equalising, noise addition, scale changes (pitch shifting and time stretching), distortions, quantization, dynamic compression, format encoding/decoding (e.g. MP3, GSM), reverberation. Moreover, each transformed version of the original can be processed by a succession of some different elementary transformations, see Sec. 4.

In some cases, at least one transformation may change the class. For example, with a vocal gender classification, a pitch shifting may change the apparent gender. In these cases, either the annotated class could be changed for the creation of the augmented training set, or the confusing transformations prohibited. For the sake of simplicity, we only use class-preserving transformations.

3.3 Sound segmentation

For copyright reasons, many public datasets are made of short sound excerpts, and most of the proposed approaches found in the literature only compute one summarising vector of descriptors per input element (cf. e.g. Fu et al., 2011). But from a practical point of view, it is no more difficult to annotate full tracks than excerpts, if the annotations do not vary over time. Consequently, if the complete duration of each song is available in the dataset, or at least a long enough excerpt, to duplicate the number of annotated examples a simple idea is to split the input signal into several shorter segments in time.

Providing that the model used is not sensitive to the duration and that the classes do not vary over time, the segmentation is class-preserving and this transformation of time can easily increase the total number of examples for the machine learning methods used.

Indeed, a song is usually made of different and successive parts in time, for example: introduction, verse, chorus and bridge, for popular music. Thus, using one descriptor vector per part, the representation of each class is wider and remains accurate. Moreover, for songs with strong differences between parts, a descriptor averaging would be meaningless in many cases.

In our implementation, the segmentation is not adaptive: we use a fixed window duration and a fixed step size. Note that this idea has been suggested by Peeters (2007).

3.4 Augmentation for training and testing

In the previous parts, data augmentation was mainly presented as a method to increase the training dataset size. As said, the main objective is to avoid overfitting for trainable algorithms, such as: PCA, LDA, GMM, SVM, k-NN.

Actually, many of the data augmentation methods presented above can also be applied during testing (Schlüter and Grill, 2015). If not, a single descriptor vector is computed from the whole original input and classified. If transformations or segmentations are used during testing, all the produced elements (transformed sounds/segments) are separately classified, and a final aggregation rule makes a unique decision based on the results of all the elements. For example, Peeters (2007) proposes two methods for the aggregation of segments: the first is based on a majority voting, named "cumulated histogram", and a second uses the mean of class probability estimates, named "cumulated probability"; we use in this paper the "cumulated probability". Figure 1 provides an overview of the classification process.

3.5 Robustness

Finally, the use of transformed/degraded sounds during training may improve the robustness to data alterations. Indeed, to predict the class of a degraded sound, it seems...
preferable to include degraded sounds into the training set, rather than only clean signals. This point is evaluated in Sec. 6.5.

Moreover, a particular case of overfitting may occur if the training examples have a common characteristic, such as encoding format (e.g. MP3). Then, re-encoding the training songs with various formats should generalise the trained models.

### 4. Transformation process

Our transformation processing was initially inspired by the Audio Degradation Toolbox of Mauch and Ewert (2013), but with additional sound transformations, for example: pitch shifting, time stretching, MP3 compression, filtering, saturation, noise addition, reverberation (cf. the complete list in Sec. 4.3). This section presents the ideas of transformation chain and of transformation strength. Then it explains the process used to obtain a high number of different transformations, based on a random draw of the transformation parameters listed below.

#### 4.1 Chain of transformations

As illustrated in Sec. 6.7, we must be aware that using a small number of different transformation types, the trained models may be specialised to them, and new overfitting problems can occur. To avoid this issue, elementary transformations are chained in series, forming a more complex transformation. Using different arrangements and different parameter settings, the total number of different sound transformations becomes significantly greater.

#### 4.2 Pseudo-random parameter drawing

Moreover, instead of manually setting the parameters individually, a special control procedure has been created: all the parameter values are randomly drawn with the constraint that the final transformation strength must match a given global strength estimate \( \Gamma \) for the transformation effect. It is designed to resemble a perceptual measure of the effect, based on subjective and informal tests realised by the authors. This constraint has been implemented using empirical mappings between the transformation parameters and the individual transformation strength \( \gamma \); see Table 1 for the meaning of the values of \( \Gamma \) and \( \gamma \).

The random choice of strengths can be summarised as follows: first, a target expectation \( \Gamma^* \) of the transformation strength is set for the whole chain, and also a parameter \( \beta_k \) for each elementary transformation \( k \) of the chain. Then, with \( X \) a uniform random variable between 0 and 1, the \( K \) elementary transformation strengths \( \gamma_k \) are randomly drawn by \( \gamma_k = \rho X^\beta_k \), with \( \rho \) such that \( \mathbb{E}[\|y\|] = \Gamma^* \mathbb{E}[\|\cdot\|] \) and \( \|\cdot\| \) respectively denote the expectation and the \( \ell_p \) norm.

**Notes:** The value of \( \rho \) is precomputed using a Monte Carlo procedure. The chosen value of \( P \) acts on the sparsity of the vector \( \gamma \), which increases with \( P \). Then, when \( P \) is higher, less elementary transformations are ‘activated’ in the chain. The parameters \( \beta_k \) individually control the ‘activation’ frequency: a transformation with a high \( \beta_k \) is less frequent than other transformations with lower values. The distribution of \( \Gamma = \|\gamma\| \) is close to a Gaussian law, but the obtained values are manually limited to a given range \( I_\gamma \), by redoing the draw until \( \Gamma \in I_\gamma \). In Sec. 6, we choose: \( P = 3, \Gamma^* = 1, I_\gamma = [0.6, 1.5] \), unless indicated.

#### 4.3 Elementary transformations of the chain

In this sub-section, we give some information about the transformation implementations and the used mappings (from the individual strengths \( \gamma \) to the parameters). Note that all these mappings have been empirically set by listening to the obtained transformations, in a way that \( \gamma = 1 \) provides clearly audible changes, but respecting the preservation of the class (genre, cf. Sec. 5), and higher \( \gamma \) values correspond to sounds that are likely to be uncomfortable. The presented order is the one used for the chain, and it follows a real-world scenario (scale changes and compressions for digital radio broadcasting, followed by filtering, saturations, noises and reverberation for non-ideal amplifiers, loudspeaker emission and room effects). Note that all processes are performed at a sampling rate of 22050 Hz.

The question of how to find the optimal transformation is a difficult task which is not dealt with here. The chosen configurations are ad-hoc, and better choices could be made depending on the task. In Sections 6.6 and 6.7, different transformation configurations are tested and compared.

**Pitch shifting and time stretching**, with preservation of formants and transients. The process is based on the work of Röbel (2003); Röbel and Rodet (2005), and is implemented by the software SuperVP.

Given a strength \( \gamma \), the mapping is as follows: the pitch shifting factor \( s \) in semitones and the time stretching scaling \( t \) in percent are randomly chosen on an ellipse of equation: \( (s/a)^2 + (s/a)^2 = \gamma^2 \), with \( a_4 = 4 \) and \( a_5 = 5 \) the respective maximal changes. For this elementary transformation, \( \beta_k = 1, \) cf. above.

**MP3 compression.** The command `lame` is used for encoding-decoding, by setting the bitrate. The values \( \gamma \in (0.35, 1, 1.5, 2) \) give the bitrates \( (56, 40, 32, 24) \) kbps respectively; and a cubic interpolation is performed for the intermediate values of \( \gamma \) (rounded to the nearest bitrate accepted by `lame`). For a less frequent use of this transformation: \( \beta_k = 3 \).

**Spectral quantization and bin dropout,** with signal reconstruction. First, the spectral quantization simply simulates the effect of a lossy compression. For example,
\( \gamma \in \{0.5, 1, 1.5, 2\} \) simulate equivalent bitrates \( \{107, 96, 40, 32\} \) kbps. Second, the dropout degradation sets some randomly chosen bins of the spectrogram to 0. For example, \( \gamma \in \{0.5, 1, 1.5, 2\} \) zeros \( \{10, 25, 40, 50\} \)% of bins. As previously, a cubic interpolation is performed for all the intermediate values of \( \gamma \); and \( \beta_1 = 3 \).

**Multi-band compression/dilatation of dynamics.** Here the release time is fixed to 100 ms, the threshold to \(-60\text{dB}\), the pre-normalization to \(-10\text{dB}\), and 5 bands are used up to the Nyquist frequency (11025 Hz).

First, a random choice is done: 80% compression, 20% dilatation. Then, with \( X \) a uniform random variable on the range \([0.75, 1.25]\), the chosen ratio is \( r = 2^X \) for compressions or \( r = 2^{-X/3} \) for dilatations. Also, the attack time is \( \tau_a = 10^{-X/4+1} \) in [ms], and \( \beta_1 = 1 \).

**Graphic equalisers.** With a spectral computation, an equaliser has been implemented. In our experiments, the frequency response is a periodic function on a logarithmic scale.

The amplitude of the response is given by \( g_{db} = \pm 4\gamma \) in \([\text{dB}]\), and the period is randomly drawn between 1 and 2 octaves independently from \( \gamma \). Also an offset in frequency is drawn. \( \beta_1 = 1 \).

Remark: a filter is implemented by designing its frequency response modulus; and using its minimum-phase solution (Oppenheim and Schafer, 2009). The spectrogram is computed with a Hann window of size 150 ms and a hop size of 37.5 ms. Then the signal is reconstructed from the filtered spectrum (Zölzer, 2011).

**Low-pass slope filter.** To smoothly delete contents in higher frequencies, a linear low-pass slope filter is applied to the signal. The decreasing slope is directly given by \( a = 5\gamma \) in \([\text{dB/decade}]\). Additionally, the starting frequency of the slope is randomly chosen between 20 and 150 Hz. Finally, the reconstruction uses the same principle as the graphic equaliser, and \( \beta_1 = 3 \).

**Sample quantisation and saturation.** First, the quantisation consists of rounding the sample values. With \( \gamma \in \{0.5, 1, 1.5, 2\} \), the quantization step is set to give a signal-to-noise ratio of \([34, 25, 20, 16]\) dB respectively. Second, the saturation consists of a sine characteristic on the sample values, with a pre-normalisation which depends on \( \gamma \); with \( \gamma \in \{0.5, 1, 1.5, 2\} \), the signal is normalised to \([-8, -2, 0.5, 2]\) dB beforehand. Note that the signal is renormalised to its original norm afterwards; and as previously, a cubic interpolation is performed for all the intermediate values of \( \gamma \); \( \beta_1 = 2 \).

**Varying gain.** To simulate an irregular tremolo, a varying gain is applied to the signal. The gain randomly varies between \( \pm 4\gamma \) in \([\text{dB}]\), with an average frequency of \( 4\gamma \) in \([\text{Hz}]\). \( \beta_1 = 2 \).

**Synthetic noise** addition to the input signal. We use a pink noise \((-10\text{db/decade})\), with a cutoff frequency at 300 Hz, and the signal-to-noise ratio of the mix depends on \( \gamma \); e.g. with \( \gamma \in \{0.5, 1, 1.5, 2\} \), \( \sigma_{db} \in \{30, 22, 18, 15\} \) dB; and a cubic interpolation is performed for the intermediate values of \( \gamma \); \( \beta_1 = 3 \).

**Convolution Reverb** with pre-recorded or pre-computed responses. Among several stored responses (for rooms, studios, churches, halls, car parks, springs, plates, etc.) one is randomly chosen, and the pre-delay is also drawn between 15 and 25 ms. Finally, the reverberation gain (setting the ratio: dry/wet effect), depends on \( \gamma \); e.g. with \( \gamma \in \{0.5, 1, 1.5, 2\} \), \( \sigma_{db} \in \{13, 6, 0, -2\} \) dB; with a cubic interpolation for the intermediate values of \( \gamma \); \( \beta_1 = 1 \).

**Background noise** addition to the signal. Among several recorded noise scenes (restaurants, streets, fairs, markets, halls, public transportation, etc.) one is randomly chosen. Then the signal-to-noise ratio of the mix depends on \( \gamma \); e.g. with \( \gamma \in \{0.5, 1, 1.5, 2\} \), \( \sigma_{db} \in \{10, 25, 40, 50\} \)%; with a cubic interpolation for the intermediate values of \( \gamma \); \( \beta_1 = 1 \).

**Time shift.** Finally an imperceptible time delay is applied, with 0 insertion before for positive values and after for negative values. Actually it can be noticed that the computed descriptors may change with a slight shift in time in some cases. Thus, a signal translation in time is performed to improve invariance of results. The shift is given by \( \pm 50\gamma \) in [ms]. \( \beta_1 = 1 \).

5. Preparation of Experiments

5.1 Datasets

The public dataset ISMIR Genre 2004 is a set of 1442 full song tracks (with an average duration of 4 minutes) split into two subsets: 713 files for training, and 729 files for testing. All the songs are partitioned into 6 unbalanced genre classes which are (the numbers give the number of songs for training/testing): ‘world’ \((106/122)\), ‘electric’ \((115/114)\), ‘classical’ \((320/320)\), ‘metal_punk’ \((45/45)\), ‘rock_pop’ \((101/102)\), ‘jazz_blues’ \((26/26)\). Most of the experiments of Sec. 6 use ISMIR-2004 because it is a well-known and public dataset.

We also consider the small subset of the FMA dataset, from the Free Music Archive (Defferrard et al., 2017). The small subset is made of 8 well balanced classes, each of them made of approximately 1000 songs: ‘Hip-Hop’, ‘Pop’, ‘Folk’, ‘Experimental’, ‘Rock’, ‘International’, ‘Electronic’, ‘Instrumental’. Note that originally the files of this small subset are 30 second excerpts, but to test the segmentation with full duration, we extracted the 8000 corresponding full tracks from the full FMA dataset. This dataset is used in Sec. 6.4, which needs more full tracks than ISMIR-2004 has.

Finally, the public dataset 1517-Artists is tested in Sec. 6.9 (Seyerlehner et al., 2010a). It is initially composed of 3180 full tracks annotated with 19 genres. Nevertheless, since it is used in a cross-dataset context with ISMIR-2004, only the genres ‘World’, ‘Electronic&Dance’, ‘Classical’, ‘Alternative&Punk’, ‘Rock&Pop’, ‘Jazz’ and ‘Blues’ are extracted, and the last two are merged to make the class ‘Jazz&Blues’. Then, the new dataset is composed of 1173
unbalanced files which are split into two subsets for training and testing, with the same class repartition. Note that the songs of an artist are gathered in the same split, avoiding the artist effect (Flexer, 2007).

5.2 Descriptors and classifiers
In this paper, for the sake of simplicity and generality, we have chosen reproducible methods well-known in the literature and based on the schema of Sec. 2. Note that many of the cited works above have applied data augmentation to Neural Network techniques (e.g. Schlüter and Grill, 2015), and its general benefit has been demonstrated already for these methods. We expect that using more complex classification methods, the results will become comparatively better and follow the same general behaviours.

The implemented method is based on the Block-Level features of Seyerlehner et al. (2010b), with the modifications of Seyerlehner and Schedi (2014). These descriptors are made of 9476 variables coming from different representations named: Spectral Pattern, Delta Spectral Pattern, Variance Delta Spectral Pattern, Log-Fluctuation Pattern, Spectral Contrast Pattern, and Correlation Pattern. Note that for a given song/segment, the summarisation is done by a percentile function, see Figure 1. After a individual normalisation of each component, a PCA projection is done to reduce the dimensionality to 200.

Finally, the classification is made by an SVM, using a One-Vs-One paradigm. In order to reduce computation time, the cost parameter is fixed to 1, but during the experiments of Secs. 6.1–6.3, and 6.5–6.7, the \( \sigma \) parameter of the RBF kernel is optimised. This optimisation is done using a grid search and a 5-fold cross validation on the ISMIR-2004 training dataset. Then, using the best \( \sigma \), the models trained on the whole training dataset are tested on the ISMIR-2004 test dataset. For the other experiments, we fixed \( \sigma = 1 \).

In order to examine whether the results are independent of the classification method, most of the experiments of Sec. 6 are also performed with a different method, presented in Appendix A.

5.3 Testing the effect of transformation parameters
Because the transformation parameters are randomly drawn, in most of the experiments which deal with sound transformations, we test the deviation of the accuracy with regard to the drawn transformations. In these cases, training and testing are repeated 25 times with the same transformations but with different drawn parameters, cf. the previous section. The given values are then the mean of accuracy, and its standard deviation \( s \). Also the 95% confidence interval of the mean accuracy is approximated by \( 1.96 s/\sqrt{25} \) (e.g. Barlow, 1989). Note that without transformation, the experiments are deterministic, thus only one iteration is needed, and \( s = 0 \).

6. Results of Experiments
This section describes experiments that show the benefit of the proposed data augmentation. Note that the aim is not to get the best results of the literature, but to compare classification accuracies when data augmentation is used or not. Even if the tendencies are not fully surprising, it is helpful to quantify them, and good practice recommendations are highlighted.

The chosen classification method is explained in Sec. 5.2, but note that the results of the same experiment using a different classifier are given in Appendix A; it can be seen that they follow the same behaviour.

6.1 Segmentation results
In this first experiment, the data augmentation by segmentation is evaluated. For both training and testing, 4 configurations are tested:

- No segmentation, i.e. the descriptors are computed on the whole song only.
- Segments of 80 seconds (s), with a step size of 40 s. Note that with an average song duration of 4 minutes, the number of vectors increases by a factor of \( 5 = (4 	imes 60 – (80 – 40))/40 \).
- Segments of 30 s, with a step size of 15 s. In this case the number of vectors is multiplied by 15.
- Segments of 15 s, with a step size of 7.5 s, providing a multiplication factor of 31.

Using all these configurations for training and testing, we get 16 tested combinations. For example, in an evaluation, the training step is done by segmenting the signal into 30 s segments whereas the testing step predicts 80 s segments, and aggregates the results as explained in Sec. 3.4 (mean of probabilities).

Table 2 shows the results in terms of accuracies for genre classification of the ISMIR-2004 dataset. The first entry of the array (rows) is for the configuration of the training, and the second entry (columns) is for the testing step.

In this experiment, the \( \sigma \) parameter of the SVM is optimised independently for each cell of the table, as detailed in Sec. 5.2. Even if the optimisation only uses the ISMIR-2004 training dataset, the vectors used during the training and testing of the cross validation are from the respective evaluated configurations which can be different.

Increasing the number of training or testing examples by using shorter segments generally improves accuracy, but for too short segments (15 s) it decreases again. Finally, with a baseline of 81.0% (no segmentation) the best result is 89.3% and is obtained with the segments of 30 s for training and testing. In consequence, the experiment

<table>
<thead>
<tr>
<th>train\test</th>
<th>no seg.</th>
<th>80 s</th>
<th>30 s</th>
<th>15 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>no seg.</td>
<td>81.0</td>
<td>84.3</td>
<td>84.9</td>
<td>81.2</td>
</tr>
<tr>
<td>80 s</td>
<td>83.7</td>
<td>86.5</td>
<td>86.8</td>
<td>85.3</td>
</tr>
<tr>
<td>30 s</td>
<td>85.6</td>
<td>86.5</td>
<td>89.3</td>
<td>86.1</td>
</tr>
<tr>
<td>15 s</td>
<td>85.3</td>
<td>85.3</td>
<td>87.2</td>
<td>87.3%</td>
</tr>
</tbody>
</table>
shows the benefit of data augmentation by segmentation using relatively short segments.

Note that the use of shorter segments (e.g. 15 s) does not provide any benefit, even if the training set size proportionally increases. Moreover, according to the descriptors used, shorter durations may not provide meaningful representations, which is the case with the Block Level features of Seyerlehner et al. (2010b).

6.2 Transformation results

The second experiment evaluates the benefits of sound transformations alone. Here, 5 configurations are tested:

- **No transformation**: only original songs are used.
- **Original + 1 transf.**: the original songs are used together with 1 transformed version of each (using the chain of elementary transformations).
- **Original + 2 transf.**: the original songs are used together with 2 transformed versions of each.
- **Original + 4 transf.**: the original songs are used together with 4 transformed versions of each.
- **Original + 14 transf.**: the original songs are used together with 14 transformed versions of each.

All these transformations are obtained by the previously detailed implementation (see Sec. 4), i.e. with a chain of 12 elementary transformations in series, and with a random draw of parameters controlled by a global transformation strength \( \Gamma^* = 1 \) (cf. Table 1).

Note that with 4 additional transformed songs (or 14 respectively), the increasing factor is 5 (or 15 resp.), as with the previous experiment with segments of 80 s (or 30 s resp.). This was done in order to compare the two methods (segmentation and transformation) with equal numbers of descriptor vectors.

As previously, the \( \sigma \) parameter of the SVM is optimised independently for each cell of the table with different augmentation configurations for training and testing. Also, to evaluate the deviation due the transformations, as noted in Sec. 5.3, 25 repetitions are performed for each experiment.

First, analysing the results of Table 3, it can be seen that when only the original songs are used during the testing step (first column), including at least one transformed version for each training song provides a significant benefit. Nevertheless, we observe that the accuracy does not significantly change when the number of transformed sounds increases. Thus the addition of many transformed sounds does not seem useful.

Second, when the training only uses original sounds, the first row shows that the results decrease when transformed sounds are added during testing. This can be explained by a problem of robustness which is is studied in Sec. 6.5.

The best result is obtained with 14 transformed songs for both training and testing with an accuracy mean of \( 85.8\% \). Nevertheless, for many practical applications, the computation at test time needs to be as fast as possible, but the processing of transformations and corresponding descriptors in this case is quite long. So, for a faster online prediction process, a good option is to use transformations only for the offline model training, with a small number of transformed sounds.

6.3 Combination of segmentation and transformation

Table 4 compares the results obtained with different combinations of segmentation and transformation for training and testing. The objective is to test if the use of both segmentation and transformation is advantageous or not. Note that for the sake of simplicity, these methods are configured using two configurations: \( 80 \text{ s segments} \) and \( \text{original+4 transf.} \). To compare segmentation and transformation independently from the number of descriptor

![Table 4: Testing combinations of segmentation and transformation. Accuracy mean (%) for ISMIR-2004.](image)

<table>
<thead>
<tr>
<th>train\test</th>
<th>S T</th>
<th>S T</th>
</tr>
</thead>
<tbody>
<tr>
<td>S T</td>
<td>81.0</td>
<td>77.2</td>
</tr>
<tr>
<td>S T</td>
<td>85.0</td>
<td>85.4</td>
</tr>
<tr>
<td>S T</td>
<td>83.7</td>
<td>81.8</td>
</tr>
<tr>
<td>S T</td>
<td>85.8</td>
<td>86.3</td>
</tr>
</tbody>
</table>

Table 3: Evaluation of transformations. Accuracy mean (%) for ISMIR-2004, using a transformation strength \( \Gamma^* = 1 \).

The small numbers in parentheses are the standard deviations (in percentage points, pp) computed with 25 repetitions. Note that the 95% confidence interval of the accuracy mean is less than 0.37pp.
vectors, these configurations have been selected because they provide the same factor 5 (see Sections 6.1 and 6.2).

Note that there is the choice of the order: segmentation before or after transformation. Either the segmentation is performed separately on each transformed version of the full input track, or several transformations are separately processed on each segment of the original track. The second option is used in this experiment, cf. Figure 1. As previously, the $\sigma$ parameter is optimised independently for each cell, and 25 repetitions are performed for each experiment with sound transformations.

In this experiment, the best result is obtained when both segmentation and transformations are used during the training and the testing: 87.1%. Therefore, the use of both methods together makes sense.

Nevertheless, since the segmentation is only a copy of shorter signals, it is fast and can be considered during training and testing. But the processing of some transformations may be very time consuming, and avoiding it at least during the testing step can be reasonable to save computation time. This provides a decrease of 0.6 or 0.8 percentage points of the accuracy mean compared to the best result of our experiment (87.1% → 86.5 or 86.3%).

Finally, comparing Tables 2, 3 and 4, we observe that segmentation is more effective than transformation, even with less examples. However, as seen in Sec. 6.5, the use of transformation during the training significantly improves the robustness.

### 6.4 Natural vs artificial augmentation

One important question is: “is it preferable to increase the training set size by data augmentation or by annotating more examples”? That is the purpose of the experiment of this section.

The small FMA dataset contains approximately 8000 full tracks, partitioned into 8 classes. Here, we evaluate the classification using 10-fold cross validation, with an artist filter (each artist is associated exclusively to a fold, with all respective original and transformed segments or tracks). For all the folds, the training sets (originally composed of 7200 files (= 8000/10 × 9)) are down-sampled to 1000 in a first case (simulating a small dataset) and to 5000 files in a second case (simulating a larger dataset). In both cases, we test: no augmentation, segmentation with 80 s segments, and transformations with 4 additional transformed sounds.

Note that using these configurations and with an average song duration of 4 minutes, the segmentation increases the number of vectors by a factor of 5, which is the same factor for the transformations, and also between the small training set and the big training set. Consequently, the small training sets with segmentation or transformation have approximately 5000 vectors, as with the big training set without augmentation. In this experiment, to save computation time, the $\sigma$ parameter of the SVM is not optimised, it is fixed to $\sigma = 1$, and no repetitions are performed for the transformations.

Table 5 quantifies an expected result: it is better to have more manually annotated examples than to augment the training set. With no augmentation, the use of 5000 files during the training provides an accuracy of 54.9% whereas the smaller data set with segmentation or transformation, also producing 5000 vectors, gives 48.6% or 48.5% respectively. Thus, more annotations produce a wider variety of examples which improves the learned models. Nevertheless, when it is not possible to manually annotate more songs, data augmentation remains a good alternative, with significant improvements.

Note that the relatively low accuracies obtained in Table 5 are due to the FMA dataset itself. As explained by the authors, many annotations may be inconsistent, which makes the classification more difficult. To compare our implementation of Block Level + SVM, we also tested the classification task of FMA-medium with the same context as the experiments of Defferrard et al. (2017), i.e. without data augmentation. We obtained an accuracy of 63.3% which is almost the same as the best score of the cited paper: 63%.

### 6.5 Robustness to degradation

The purpose of this experiment is to test the robustness to sound degradations, i.e. the ability of the method to recognise the class when the given input signal to predict is transformed, or corrupted by strong degradations. To test this, as a first case, training is performed with segmentation (80 s) and 4 transformation configurations: original, 1, 2 and 4 transformed versions of each segment, with the same process as before (1 = 1). Then the signals to predict are degraded and classified several times, with an increasing global degradation strength $\Gamma$, from 0 to 2. In this experiment, the $\sigma$ parameter is optimised once for each row of Table 6, with 25 repetitions for each experiment.

#### Table 5: Natural vs artificial data augmentation. Accuracy mean (%) for FMA (10-fold cross validation).

<table>
<thead>
<tr>
<th></th>
<th>Small (1000)</th>
<th>Big (5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No augmentation</td>
<td>45.8</td>
<td>54.9</td>
</tr>
<tr>
<td>Segmentation</td>
<td>48.6</td>
<td>55.2</td>
</tr>
<tr>
<td>Transformation</td>
<td>48.5</td>
<td>54.7</td>
</tr>
</tbody>
</table>

#### Table 6: Robustness to degradation, shown by mean prediction accuracy (%) for ISMIR-2004. Rows represent amount of data augmentation; columns represent transformation strength $\Gamma$. The standard deviations are less than 0.88pp and the 95% confidence intervals of the mean are less than 0.35pp.

<table>
<thead>
<tr>
<th>$\Gamma$ for testing</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>86.5</td>
<td>73.2</td>
<td>74.2</td>
<td>71.7</td>
<td>68.9</td>
</tr>
<tr>
<td>+1 transf.</td>
<td>86.3</td>
<td>85.2</td>
<td>84.5</td>
<td>82.7</td>
<td>81.1</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>86.4</td>
<td>86.0</td>
<td>85.6</td>
<td>83.9</td>
<td>82.1</td>
</tr>
<tr>
<td>+4 transf.</td>
<td>86.3</td>
<td>86.8</td>
<td>86.0</td>
<td>84.7</td>
<td>83.0</td>
</tr>
</tbody>
</table>
Comparing the rows of Table 6, we first see that the accuracy roughly decreases when the degradation of the signal to predict is stronger. But whereas the decrease is very strong without transformation during the training (with a loss of 17.6 percent points between $\Gamma = 0$ and $\Gamma = 2$), it is weak when some transformed versions are used in the training set (with a loss of 3.3 percent points in the same context with 4 additional transformations).

Consequently, even if the benefit of sound transformation is not always significant, this experiment demonstrates an improved robustness to transformation or degradation of the signal to classify.

### 6.6 Individual and chained transformations

Figure 2 compares the classification accuracy for different transformations used during training: individual or chains. The tests are performed solely with the original testset of ISMIR-2004, without segmentation or transformation. This experiment imitates the one of Schlüter and Grill (2015), with the transformations: dropout with 5%, 10% and 20% of zeroed bins; Gaussian noise with SNRs of 26dB, 20dB and 10dB; pitch shifting and time stretching with $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, and $\pm 50\%$; frequency filter with $\pm 5\text{dB}$, $\pm 10\text{dB}$, and $\pm 20\text{dB}$. Then other transformations are tested: MP3 encoding/decoding with 128 kbps, 80 kbps, and 40 kbps; background noise addition with 26dB, 20dB, and 13dB; equaliser and varying-gain with $\pm 5\text{dB}$, $\pm 10\text{dB}$, and $\pm 20\text{dB}$; and reverberation with a mix coefficient of $\pm 20\%$, $\pm 30\%$, and frequency filter $\pm 10\text{dB}$, the second is the chain used previously in this paper, cf. Sec. 4 with random parameters, and the last is a mixed combination of the two previous chains. Note that the $\sigma$ parameter is optimised as explained in Sec. 5.2, and 25 repetitions are done for each tested transformation.

The figure reveals that all transformations, individual or chained, provide a better classification on original signals. Time stretching is the most effective and Gaussian noise the least effective transformation. These results are quite different from the results of Schlüter and Grill (2015), but first the task is different, and second the transformations are performed on the signal itself and not on the spectrogram. The three chains do not seem to provide better results than the time stretching, but their real benefit is shown in the next section.

### 6.7 Testing of transformation overfitting

This section tests the classification of transformed signals when the models are trained with different transformations (individual or chains). The aim is to demonstrate that the training with one individual transformation produces overfitting to the transformation of the training. Using 13 different transformations, Figure 3 compares the classification of original and transformed signals when the training is performed with original or transformed datasets.

Note that the $\sigma$ parameter is optimised once for each row of Figure 3, and 25 repetitions are performed for each cell. For sake of simplicity, only the mean accuracy is displayed.

**Figure 2:** Individual and chained transformations. Each horizontal colored bar and black segment represents the mean accuracy and its standard deviation computed with 25 repetitions of each experiment. The vertical dashed line represents the accuracy without transformation, and the dotted line represents the mean accuracy for the transformation chain used in this paper.

**Figure 3:** Testing of transformation overfitting. The rows represent the transformations used during training, and the columns represent the transformations of the test signals (ISMIR-2004).
On the one hand, analysing the results of the green square (for individual transformations), the most noticeable result is that for each line and each column, the cell on the diagonal contains the best or the second best score. This observation shows that a transformed signal is best classified with models based on the same transformation, which reveals the possible overfitting using individual transformations.

On the other hand, using a chain of some transformations during the training (three last rows), the classification of transformed signals is improved in most of the cases, avoiding the previously mentioned overfitting problem. Nevertheless, it is not easy to find a chain that improves robustness in all cases: for example the chain of Schlüter and Grill (2015) is not effective for additional noises, and the chain of this paper is weaker with pitch shifting.

An interesting observation is that the mixed chain (combining training examples transformed by the chain of Schlüter and the chain used in this paper) takes almost the best of each. However, it is not always the case, as we saw in other experiments not reported here.

### 6.8 Sensitivity of models to transformations

In this paper, we define “sensitivity” as the property of a model to degrade its prediction accuracy when the inputs to classify are modified by a given transformation. Because the first row of Figure 3 shows the mean accuracy of the classification of transformed sounds without data augmentation, it is an indication of the sensitivity of the model to the different tested transformations.

Based on the mean accuracies displayed in the first row of Figure 3, we see that time stretching seems to be a transformation to which the model is less sensitive. In this case, we could assume that the time stretching would be useless for data augmentation. However, looking at Figure 2 we observe that this transformation provides almost the greatest benefit when it was used during training.

Conversely, Figure 2 shows that the addition of Gaussian noise during the training provides only little benefit when classifying clean sounds. But, observing the column (D) of Figure 3, the model seems to be very sensitive to Gaussian noise (upper cell), and using it during the training obviously improves the robustness of the classification.

From these observations, we can draw the following hypotheses: first, data augmentation can be used to increase the accuracy when we only have a small training dataset, and transformations to which the model is not sensitive are useful (time stretching for example). Second, data augmentation improves the robustness when classifying degraded sounds, and in this case it is useful to use transformations to which the model is sensitive (Gaussian noise for example).

### 6.9 Testing transformations for cross-dataset issues

In many cases, the musical tracks collected to make a dataset share common characteristics, such as mixing style or audio compression format. After training a model with the training set of a given dataset, we usually obtain best results using a testing set from the same dataset rather than from another one, possibly with different characteristics. In this present section, we study a cross-dataset scenario, and by using audio transformation during training, we test whether it results in an improvement in robustness.

The tested datasets are: ISMIR-2004 (already split into training and testing sets), and 1517-Artists with the class modifications detailed in Sec. 5.1 which aims to match the taxonomy of ISMIR-2004. Recall that 1517-Artists has been split like ISMIR-2004 into two subsets for training and testing. The transformations (if used) are performed with the same chain as in Sections 6.2, 6.3 and 6.4, and segmentation uses segments of 80 s for training and testing. Finally, the classification method is the same: Block-Level features, normalisation, PCA and SVM classifier.

The results are presented in Table 7. First, when the testing set and training set come from the same dataset, the results of the modified 1517-Artists show lower accuracy than with ISMIR-2004, which suggests a weaker consistency of its annotations. Even if the use of transformations does not change the accuracy of the tests of ISMIR-2004 (in all the cases), the transformations improve the predictions of the 1517-Artists testset in a cross-dataset context (trained with ISMIR-2004), with a gain of almost 6pp. These results are not as convincing as we expected, but at least they emphasize the improved robustness resulting from data augmentation.

### 7. Conclusion

In this paper we provide an in-depth analysis of two data augmentation methods: sound transformation and sound segmentation. Testing them for a genre classification task, some experiments have shown their ability to significantly improve the results for small datasets when it is not possible to annotate more examples manually.

First, among the tested segmentation configurations of Sec. 6.1, it was seen that it is preferable to use segments of 30 seconds both during training and testing, rather than a longer duration. But as noted, depending on the descriptors used, segments shorter than that may not provide meaningful representations, and they did not provide improvements with the method tested.

Second, the evaluation results of Sec. 6.2 showed the benefit of applying transformations to the training examples. Also, it was observed that there is no use in

### Table 7: Classification accuracy (%), showing the effect of transformations for cross-dataset issues. The SVM parameters $C$ and $\sigma$ are fixed to 1.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing set (only original)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISMIR-2004</td>
<td>ISMIR-2004</td>
</tr>
<tr>
<td>original</td>
<td>85.0</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>85.3</td>
</tr>
<tr>
<td>1517-Artists</td>
<td>ISMIR-2004</td>
</tr>
<tr>
<td>original</td>
<td>57.0</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>56.9</td>
</tr>
</tbody>
</table>
employing more than a few transformed versions of each example. Then, in Sec. 6.5, the robustness of models trained with transformations was shown experimentally.

In Section 6.7, the results of individual transformations (without a chain) show an overfitting problem which focuses the models to the transformation type used during the training. Using different transformations in series (a chain) does not improve classification of original sounds (compared to results using individual transformations), but it significantly improves the overall robustness, i.e. when applying the model to transformed or degraded sounds.

The question of which is the best chain of transformations is not solved in this paper. We use here a complex chain of 12 elementary transformations with randomly selected parameters (see Sec. 4.3), which provides different results than the chain of Schlüter and Grill (2015). Even if the tendencies show an overall benefit to use transformation chains, the selection of the best chain is not an easy task. First, the number of possible choices (parameter settings, order, etc.) is not limited, and second it can be expected that the optimal chain strongly depends on the MIR task, and that the improvements may not be significant when testing original sounds.

For the segmentation method, note that for some tasks where the annotations may vary in time, such as a sung/instrumental classification, the segmentation may change the class, e.g. by extracting instrumental segments from a sung musical piece. However, it has been observed in similar tasks (not presented here) that the aggregation can correct this, thanks to the well annotated segments, and a fine setting of the cost parameter of the SVM avoids the corruption of the model by incorrectly annotated examples. Nevertheless, the reader must be aware of this possible issue in order to apply suitable adaptations. Note that Mandel and Ellis (2008) and Schlüter (2016) dealt with the similar problem of Multiple Instance Learning where Naive approaches are considered.

A. Experimental Results with Another Classification Method

This section shows the results of some experiments of Sec. 6 using a different method. This has been done to validate the reproducibility of the some observed behaviours, independently from the method used.

Here is a summary of the method: a collection of 177 standard descriptors is computed frame-by-frame: MFCCs (13), δ-MFCCs (13), auto-correlation coefficients (12), chroma features (12), spectral flatness measure (4), spectral crest function (4), loudness, and various spectral information (centroid, spread, kurtosis, skewness, decrease, slope, rolloff, variance, and tristimulus; standard and perceptually weighted, with 6 different scales). All these descriptors are summarised by computing the means and the standard deviations for the whole duration of the segment or song (Davis and Mermelstein, 1980; Peeters et al., 2011). Additionally, 108 coefficients of the modulation spectrum are concatenated to the former ones to form the whole descriptor vector (Lee et al., 2009a).

Finally, feature selection is performed with IRMFSP (Peeters and Rodet, 2003), before applying an LDA projection and a GMM classifier (with 2 Gaussians and full covariance matrices). Moreover, because the GMM modeling depends on its initialisations which are randomly chosen here, as done in Sec. 5.3 for the drawing of transformation parameters, for some of the following experiments 25 repetitions have been computed with different initialisations, in addition to the transformations when they are used.

Table 8 presents the evaluation with transformations of the new model \( \text{StdDesc+ModSpec+GMM} \). Compared to the results of Table 3, it shows similar behaviours. For example the best result is also obtained with the highest number of transformations in both the training step and the testing step. Nevertheless, the results of the first column (transformations during the training step and not during the testing step) do not show any significant improvement compared to the reference value of 83%.

Note that the mean of the baseline (without data augmentation) is higher with this method than using the method Block Level+SVM. It is now between 83% and 83.9% (cf. Tables 8–10) instead of 81%. This can be explained by the lower complexity of the method which results in a better behaviour with small datasets. But when using data augmentation, the previous method usually provides better results.

As in Sec. 6.1, Table 9 globally presents better results when using a relatively high number of segments during the two steps, excepted with too short segments (15 s). Nevertheless, using model \( \text{StdDesc+ModSpec+GMM} \), the segmentation during the testing step provides worse results than the reference. Consequently with this model,

**Table 8:** Evaluation of transformations. Accuracy mean (%) for ISMIR-2004, \( \Gamma^* = 1 \), using: \( \text{Std- Desc+ModSpec+GMM} \).

<table>
<thead>
<tr>
<th>train\test</th>
<th>original</th>
<th>+1 transf.</th>
<th>+2 transf.</th>
<th>+4 transf.</th>
<th>+14 transf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>83.0 (0.4)</td>
<td>79.8 (0.8)</td>
<td>79.1 (0.4)</td>
<td>78.9 (1.2)</td>
<td>78.5 (1.1)</td>
</tr>
<tr>
<td>+1 transf.</td>
<td>82.7 (0.9)</td>
<td>82.8 (0.9)</td>
<td>83.1 (0.8)</td>
<td>83.4 (0.9)</td>
<td>83.6 (0.9)</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>83.3 (1.0)</td>
<td>83.3 (0.8)</td>
<td>83.6 (0.8)</td>
<td>84.3 (1.1)</td>
<td>84.5 (0.8)</td>
</tr>
<tr>
<td>+4 transf.</td>
<td>83.1 (0.5)</td>
<td>83.5 (0.8)</td>
<td>84.0 (0.8)</td>
<td>84.4 (0.7)</td>
<td>84.8 (0.4)</td>
</tr>
<tr>
<td>+14 transf.</td>
<td>83.7 (0.9)</td>
<td>84.1 (0.7)</td>
<td>84.7 (0.7)</td>
<td>85.0 (0.5)</td>
<td>85.3 (0.4)</td>
</tr>
</tbody>
</table>

The small numbers given between parentheses are the standard deviations (pp) computed with 25 repetitions. Note that the 95% confidence interval of the accuracy mean is less than 0.49 pp.
it is important to use the segmentation at least during the training step.

Compared to Table 4, Table 10 presents global similarities excepted for the first column (no data augmentation during the testing step).

Table 11 shows approximately the same behaviour as Table 5, without significant difference. The only difference is the baseline as noted previously.

As in Sec. 6.5, Table 12 presents the benefit of transformation in terms of robustness. But the first column does not show any improvement when classifying original (i.e. non-transformed) signals.

Compared to Table 7, Table 13 presents an interesting, but unexplained, difference: in a cross-dataset scenario, the transformation of the training dataset provides a significant benefit when classifying 1517-Artists tracks using models trained on ISMIR-2004 with the Block Level+SVM method (Sec. 6.9), but on the contrary with the StdDesc+ModSpec+GMM method, the benefits are present when classifying ISMIR-2004 tracks using models trained on 1517-Artists.

Table 9: Evaluation of segmentation. Accuracy (%) for ISMIR-2004, using: StdDesc+ModSpec+GMM. The small numbers given between parentheses are the standard deviations (pp) computed with 25 repetitions. Note that the 95% confidence interval of the accuracy mean is less than 0.41pp.

<table>
<thead>
<tr>
<th>train\test</th>
<th>no seg.</th>
<th>80 s</th>
<th>30 s</th>
<th>15 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>no seg.</td>
<td>83.9(0.6)</td>
<td>83.8(0.6)</td>
<td>83.6(0.6)</td>
<td>82.0(0.6)</td>
</tr>
<tr>
<td>80 s</td>
<td>85.2(0.6)</td>
<td>86.4(0.6)</td>
<td>86.3(0.6)</td>
<td>85.8(0.6)</td>
</tr>
<tr>
<td>30 s</td>
<td>85.3(0.3)</td>
<td>85.9(0.3)</td>
<td>86.9(0.3)</td>
<td>86.9(0.3)</td>
</tr>
<tr>
<td>15 s</td>
<td>84.8(0.3)</td>
<td>85.6(0.2)</td>
<td>85.8(0.4)</td>
<td>85.9(0.2)</td>
</tr>
</tbody>
</table>

Table 10: Testing combinations of segmentation and transformation. Accuracy (%) for ISMIR-2004, using: StdDesc+ModSpec+GMM. cf. Table 4 for an explanation. Note that the 95% confidence interval of the accuracy mean is less than 0.43pp.

<table>
<thead>
<tr>
<th>train\test</th>
<th>S T</th>
<th>S T</th>
<th>S T</th>
<th>S T</th>
</tr>
</thead>
<tbody>
<tr>
<td>S T</td>
<td>83.1(0.7)</td>
<td>78.6(1.1)</td>
<td>83.2(0.3)</td>
<td>80.3(0.8)</td>
</tr>
<tr>
<td>S T</td>
<td>83.1(0.7)</td>
<td>84.4(0.6)</td>
<td>83.5(0.7)</td>
<td>85.1(0.7)</td>
</tr>
<tr>
<td>S T</td>
<td>85.3(0.3)</td>
<td>83.2(0.7)</td>
<td>85.8(0.6)</td>
<td>84.4(0.5)</td>
</tr>
<tr>
<td>S T</td>
<td>83.8(0.6)</td>
<td>84.3(0.6)</td>
<td>84.3(0.7)</td>
<td>85.0(0.6)</td>
</tr>
</tbody>
</table>

Table 11: Natural vs artificial data augmentation. Accuracy (%) for FMA using StdDesc+ModSpec+GMM. cf. Table 5 for an explanation.

<table>
<thead>
<tr>
<th></th>
<th>Small (1000)</th>
<th>Big (5000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No augmentation</td>
<td>41.7</td>
<td>54.0</td>
</tr>
<tr>
<td>Segmentation</td>
<td>48.2</td>
<td>54.0</td>
</tr>
<tr>
<td>Transformations</td>
<td>48.0</td>
<td>53.4</td>
</tr>
</tbody>
</table>

Table 12: Robustness to degradation, shown by mean prediction accuracy (%) for ISMIR-2004, using StdDesc+ModSpec+GMM. Rows represent amount of data augmentation; columns represent transformation strength $\Gamma^*$. The standard deviations are less than 1.05pp and the 95% confidence intervals of the mean are less than 0.41pp.

<table>
<thead>
<tr>
<th>$\Gamma^*$ for testing $\rightarrow$</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>85.8</td>
<td>79.9</td>
<td>78.1</td>
<td>75.8</td>
<td>72.2</td>
</tr>
<tr>
<td>+1 transf.</td>
<td>84.9</td>
<td>84.2</td>
<td>83.7</td>
<td>82.5</td>
<td>80.6</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>84.4</td>
<td>84.3</td>
<td>83.8</td>
<td>83.1</td>
<td>81.4</td>
</tr>
<tr>
<td>+4 transf.</td>
<td>84.2</td>
<td>84.2</td>
<td>84.4</td>
<td>83.5</td>
<td>81.6</td>
</tr>
</tbody>
</table>

Table 13: Classification accuracy (%), showing the effect of transformations for cross-dataset issues, using StdDesc+ModSpec+GMM.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing set (only original)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISMIR 2004</td>
<td>ISMIR 2004</td>
</tr>
<tr>
<td>original</td>
<td>46.7</td>
</tr>
<tr>
<td>+2 transf.</td>
<td>56.7</td>
</tr>
</tbody>
</table>

Notes
2. The frequency filter is here implemented using a band-pass shelving filter (Orfanidis, 2005).

Acknowledgements
This work was partly supported by European Union’s Horizon 2020 research and innovation program under grant agreement No 761634 (Future Pulse project).

Competing Interests
The authors have no competing interests to declare.

References


