Exploring Data Augmentation to Improve Music Genre Classification with ConvNets

Rafael L. Aguiar
Postgraduate Program
in Informatics (PPGIa)
Pontifical Catholic University
of Paraná (PUCPR)
Curitiba, Brazil
Email: aguiar.pr@gmail.com

Yandre M. G. Costa
Graduate Program in
Computer Science (PCC)
State University
of Maringá (UEM)
Maringá, Brazil
Email: yandre@din.uem.br

Carlos N. Silla Jr.
Postgraduate Program
in Informatics (PPGIa)
Pontifical Catholic University
of Paraná (PUCPR)
Curitiba, Brazil
Email: carlos.silla@pucpr.br

***Abstract*—In this work we address the automatic music genre classification as a pattern recognition task. The content of the music pieces were handled in the visual domain, using spectrograms created from the audio signal. This kind of image has been successfully used in this task since 2011 by extracting handcrafted features based on texture, since it is the main visual attribute found in spectrograms. In this work, the patterns were described by representation learning obtained with the use of convolutional neural network (CNN). CNN is a deep learning architecture and it has been widely used in the pattern recognition literature. Overfitting is a recurrent problem when a classification task is addressed by using CNN, it may occur due to the lack of training samples and/or due to the high dimensionality of the space. To increase the generalization capability we propose to explore data augmentation techniques. In this work, we have carefully selected strategies of data augmentation that are suitable for this kind of application, which are: adding noise, pitch shifting, loudness variation and time stretching. Experiments were conducted on the Latin Music Database (LMD), and the best obtained accuracy overcame the state of the art considering approaches based only in CNN.**

Keywords*—Data augmentation, Music information retrieval, Automatic music genre classification, Spectrograms, Deep learning, Convolutional Neural Networs.

I. Introduction

Music Information Retrieval (MIR) is an interdisciplinary area of study which has attracted attention from the research community since the beginning of the 2000s. There are several examples of problems that can be assessed in this field of research, like music mood classification [1], [2], musical instrument classification [3], recommender systems for music [4], [5], and so on. Among these problems, we can highlight the Music Genre Classification. It was originally introduced as a pattern recognition task by Tzanetakis and Cook [6], in 2002, and since then it has been considered a great challenge to the machine learning research community.

Music genre is probably the most common way people use to categorize music [7]. Although there is no formal definition for musical genre, it is noticeable that most of the music pieces commonly associated to the same genre keeps similar characteristics to each other regarding to rhythm, timbre, and pitch, for example. Most of the works on music genre classification developed until recently were based on hand-made features. In this sense, many researchers devoted significant efforts aiming to investigate different ways to describe physical properties of the audio signal to use as input for classification algorithms [6], [8], [9].

From 2011, Costa et al. [10], [11] started to investigate a new way to describe the audio content for music genre classification purposes. The authors introduced an approach in which the audio signal is converted to a time-frequency image (i.e. spectrogram), and then it is explored in the visual domain by using well-known image texture descriptors (GLCM – Grey Level Co-occurrence Matrices, LBP – Local Binary Patterns, LPQ – Local Phase Quantization, Gabor Filters, among others) taken from the image processing literature. The authors achieved good results using this strategy, and more recently they started to investigate the performance of classifiers created using representation learning, based on Convolutional neural networks (CNN or also ConvNets), and the combination of these classifiers with others, based on hand-made features [12].

CNN is a potent deep learning method introduced by LeCun [13]. It was created in 1998, but it has become popular only in the last 6 years, thanks to the accessibility of Graphic Processing Units (GPUs), which have a massively parallel architecture specifically designed for handling multiple tasks simultaneously. To the best of our knowledge, CNN has been used in MIR tasks since 2012 [14]–[18]. Regardless to peculiarities of these works, it is noticeable that as deeper is the learning architecture, as larger must be the dataset used in the experiments. In this work we investigate whether or not the use of data augmentation improves the performance of CNN for music genre classification. For this reason, we have evaluated four different strategies to generate additional spectrogram samples for each music piece, they are: noise addition, pitch shifting, time stretching and loudness variance.

Our computational experiments were performed on the Latin Music Database [19]. The analysis of our results shows that, overall, using the data augmentation approach can improve the classification results. But this will also depend on the fusion rule used.

The remainder of this paper is organized as follows: Section II presents some concepts used in this work, like CNNs and data augmentation techniques. Section III describes our methodology and the dataset we use to evaluate the experi-

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

ments. The outcome of the experiments are in Section IV and, finally, in Section V we make the finals remarks.

II. Theoretical background

In this Section we present a brief explanation about concepts related to our work. Section II-A briefly describes deep learning and CNNs. Section II-B lists up some data augmentation techniques that are normally found in CNNs based in natural images and why they are not suitable for spectrograms. In Section II-C we describe some data augmentation techniques used in CNNs based in audio signal.

A. Deep learning and CNNs

Deep learning architectures are artificial intelligence systems composed of multiple layers of non-linear transformations. Because of their several levels of abstractions they are able to learn complex representations. The learning is hierarchical: it is possible to learn high-level features by learning the the low-level ones. Deep learning systems have being achieving the state-of-art on several problems, such as speech recognition and object detection [20].

Deep learning systems are harder to train than shallow ones. Moreover, they need more samples to provide a good classification model. Well-labeled large datasets are not always available or they might be hard to acquire and to operate. In this work we investigate the impacts of the use of a data augmentation strategy starting from a subset of the LMD composed of only 900 samples (more information about the restricted number of samples in Section III).

The most common deep learning architecture is the CNN, which consists of a neural network that has at least one convolutional layer on the top (there are usually more than one and they are often interpolated with pool layers). The first work that introduced the idea of combining the convolutional operation with neural networks is dated 1990 [21], the CNN was formalized in 1998 [13] and its potential was broadly propagated in 2012 [22].

CNNs that use high-resolution images as input need much more computational power or time, as the convolutional operation must slide through the whole image. Furthermore, they have even more trainable parameters, which would require more training data to create a well generalizable model. To deal with these issues we implement the same approach implemented by Costa et al. [12], where the spectrograms were split into patches and the patches were used as the input of the CNN (more details about it in Section III-C).

B. Classical data augmentation techniques

Steinkraus et al. [24] address document analysis in their work and state that the most important practice in CNN is to have as much as possible data for training. Their data augmentation techniques were elastic distortions (translations, rotations and skewing) and bi-linear interpolation applied to images.

Krizhevsky et al. [22] deals with object recognition. They used image translations and horizontal reflections in order to artificially augment the original dataset. They also use some kind of patch approach, they got random patches sized 224×224 from images that originally are 256×256 and, still, they manipulate the RGB colors channels in order to increase the training data.

Keras[1], a deep learning library, also provides implementations of some data augmentation techniques, like rotations and reflections. When using these techniques on Keras there is no need to generate artificial images before the training phase. The data augmentation is done on the fly, while one mini-batch is running, the data for the next one is being generated.

However it should be noted that some data augmentation techniques commonly used in other computer vision applications, such as image rotation for example, do not work with spectrograms as they can distort important information. This is to be expected as the spectrograms are based on the Short Time Fourier Transform (STFT) and they are representations of the audio signal on the visual domain. Therefore, rotating the images would artificially create images that no longer maintained the information across the time domain (X–axis). Thus, it is important that data augmentation approaches for audio applications are proposed and evaluated taking into account their specificities.

C. Data augmentation techniques applied in audio signal

In order to improve a CNN-based system of singing voice detection, Schlüter and Grill [23] used a series of data augmentation techniques, namely: adding different kind of noises, pitch shifting, time stretching, variation of loudness and random frequencies filters. Their highest accuracy was achieved applying data augmentation techniques in both the training and test sets.

The system that won the BirdCLEF 2016 Challenge[2] was developed on the bases of speech recognition and deep learning. Sprengel et al. [24] used time and pitch shifting, adding noise and combining same class samples to enlarge the dataset. Their data augmentation techniques were used only in the training set.

Takahashi et al. [25] proposed a novel data augmentation method to introduce more variance in a non-linear way for the problem of acoustic event detection. They mixed sounds of the same classes for training in randomly time and perturb the sound by boosting and attenuating a particular frequency band. The application of data augmentation raised their accuracy in both CNNs developed in that work.

In this work we follow the same procedure that obtained the best results in Schlüter and Grill [23], i.e. we will apply the data augmentation to both the training and test sets. The data augmentation techniques that will be used are: adding noise, pitch shifting, time stretching and loudness variance.

III. Methodology

In this Section we present the methodology, tools and libraries used in this work. The overview of the experimental protocol is illustrated in Figure 1. The first step is converting the digital audio signal into spectrogram images. Then the images are segmented into patches — vertical slices of

[1] https://keras.io/

[2] http://www.imageclef.org/lifeclef/2016/bird

spectrograms. Patches are the input of the CNN both in training and testing phases.

The output of the CNN is composed of classes predictions for each patch. As our protocol consists of classifying several patches from each sample, we must combine decisions obtained for all the patches from the same sample to generate the final outcome of that sample. The sum rule presented the best accuracy in all cases, as discussed in Section III-E. More details about the process are described along this Section.

In order to evaluate the proposed methodology, experiments were performed on the LMD, a music dataset composed of 3,227 samples of 501 artists assigned to one of 10 possible Latin genres. Due to cultural similarities found between the Latin countries from where the genres were taken, it is very hard in many cases to distinguish between them. Thus, genre classification using LMD can be considered a challenging task. The 10 musical genres found in this dataset are the following: "Axé", "Bachata", "Bolero", "Forró", "Gaúcha", "Merengue", "Pagode", "Salsa", "Sertanejo", and "Tango" [19].

In 2007, Flexer [26] introduced the artist filter concept. Since then, it is considered an important constraint to be taken into account in works that aim to perform music genre classification. The artist filter constraint states that songs from the same artist cannot be present in both the training and test sets simultaneously. The artist filter was proposed in order to avoid the development of classifiers that are able to perform author recognition instead of genre recognition.

To fit the LMD dataset into a cross-validation protocol considering the artist filter restriction, we had to select a subset of it. The subset is composed of 900 samples distributed into 3 folds of 300 samples each. All folds are composed of 30 samples of each class. As a consequence of the cross-validation protocol, the results in this paper are the average of the 3 folds and its standard deviation.

The remainder of this section is organized in subsections, as follows: Subsection III-A defines how the sampling of the audio signal was done, Subsection III-B describes the spectrogram generation and some pre-processing. The data augmentation techniques applied are described in Subsection III-C, details of the CNN employed are found in Subsection III-D and the methods for combining all the results of each patch into the result of a sample are presented in Subsection III-E.

A. Audio signal sampling

In MIR research it is common to use only part of the audio signal instead of the whole signal from the music. This strategy is used in order to avoid noise in live recordings, use more significant portions of the music and even balance the length of the signal. For example, in [27] the authors extracted 30 seconds from each music piece (3 parts of 10 seconds each). While in [12], [28] 60 seconds of each sound track were used, sequentially.

In this work, we decided to use the whole music signal except the first and the last 20 seconds, as we intend to obtain more data to use in the CNN in order to improve the generalization capability. The decision on the exclusion of a 20 seconds segment, both in the beginning and in the end of the music piece, was done aiming to attempt to remove non-expressive portions of the music, avoiding noises that typically occur in these parts, along with fade-in and fade-out effects. It should be noted that all experiments use this same underlying strategy and then we apply the specific data augmentation strategies.

B. Spectrograms generation

In this work we explore the audio signal in the visual domain, as it is often done in related works [12], [27]. To generate spectrograms the software SoX[3] was used.

For each audio signal a spectrogram is generated with a fixed height of 513 pixels, this value was empirically defined considering experiments already developed using this dataset and previously published [11], [12]. The wide of the spectrogram images generated varies proportionally to the duration of the music, since the time is related to the X–axis.

Once the spectrograms are created, we follow the strategy used by Costa et al. [12], we apply a downsampling to the spectrograms generating images of 256 pixels height and a varying axis that depends on the duration of the audio signal, as explained in the previous paragraph.

After the resizing, each patch extracted is sized 256×16, the same used in [12]. We experimented variations of this value and extraction with overlapping, but they did not improve the results. The total number of patches per spectogram will depend on the signal duration. Figure 2 illustrates the process of splitting spectrograms into patches. When the last patch of a spectrogram has less than 16 pixels width it is discarded.

It should be noted, that the processing steps up to this point is what is used as the baseline approach in this work.

C. Data augmentation techniques

1) Noise addition: Adding noise into a signal is a simple way to implement data augmentation. In this work it was performed through duplicating the patch and imposing a probability p to each pixel of the copy being *turned off*. The spectrogram is a gray scale image and by *turned off* we mean setting the value to 0. We use $p = 20\%$, the same used in the Dropout noise addition method implemented in [23]. With this technique we duplicate the number of samples of the dataset, because we have used all the patches from the baseline spectrogram and also the patches from the spectrogram that was generated after applying the noise addition data augmentation procedure. An example of this method is presented in Figure 3(f).

2) Loudness: Another augmentation approach that was used in this work is the generation of spectrograms with different loudness range values. In [23] the authors applied random factors. In this work we generate spectrograms for the whole dataset with -10 and +10 *dB*. We also use SoX in this step and the number of samples increase by a factor of 3, therefore we will extract the patches from three spectrograms, the baseline plus the +10 loudness and -10 loudness spectrograms. Examples of this technique are presented in Figures 3(b) and 3(c).

[3]http://sox.sourceforge.net/sox.html

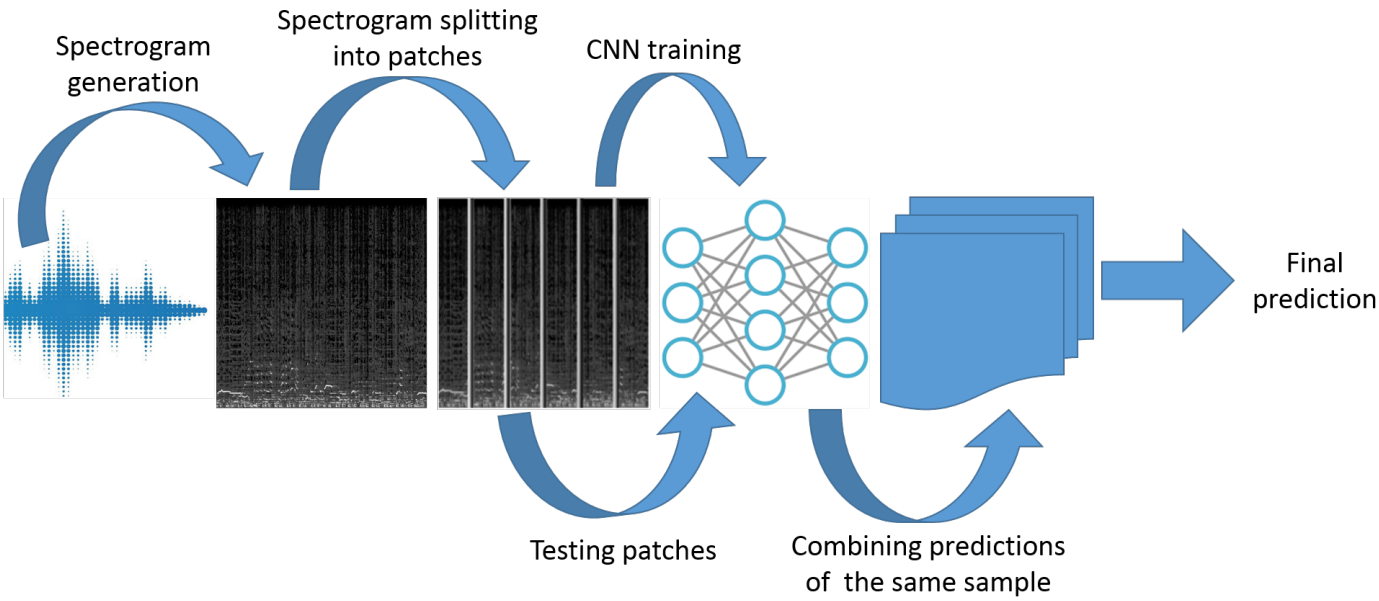


Fig. 1: Methodology overview.

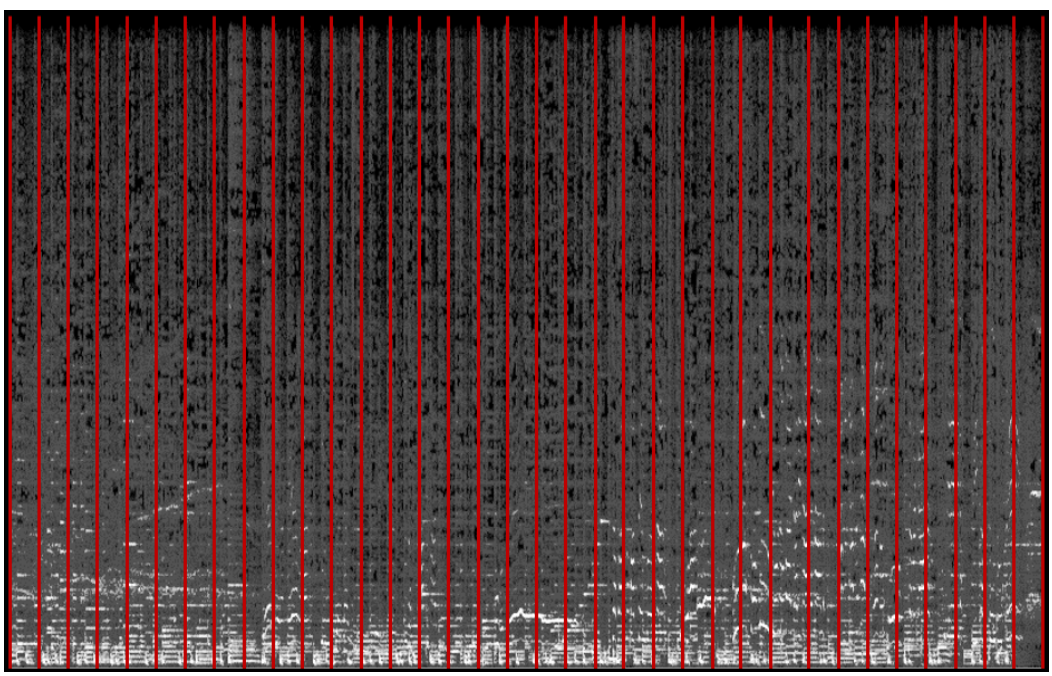
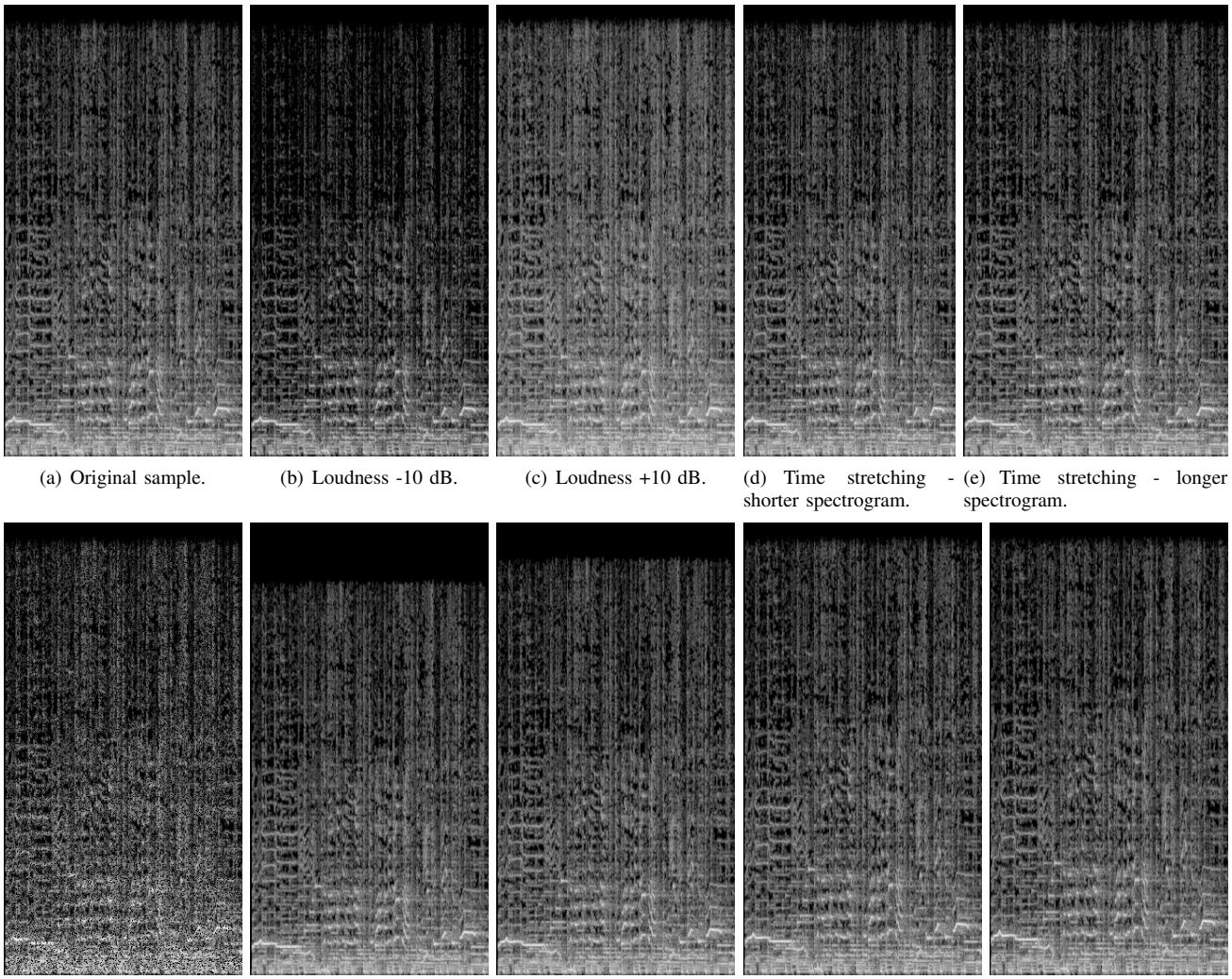


Fig. 2: A spectrogram split into several patches.

3) Time Stretching: Spectrogram images are generated through the STFT, and one if its parameters is the step of application of the Fast Fourier Transform (FFT) window. Increasing or decreasing this value lead to spectrograms with different sizes along the time dimension and, therefore, different images but with similar textures. In ours experiments we choose to generate spectrograms that are about 11% longer and 11% shorter than the default configuration. The number of patches in this experiment were about three times higher than the number of patches from the baseline experiment. Examples of time stretching spectrograms are presented in the Figures 3(d) and 3(e), both spectrograms are related to the same sample of audio used to generate the Figure 3(a).

4) Pitch Shifting: When thinking about music it is natural that a given song can be played in different tones. The idea behind pitch shifting is to increase the number of samples by transposing the tone of the song. In this work we have experimented two different approaches. In the first approach we generate two new spectograms by shifting the pitch of the songs by a semitone. This is done by using a higher



(a) Original sample. (b) Loudness -10 dB. (c) Loudness +10 dB. (d) Time stretching - shorter spectrogram. (e) Time stretching - longer spectrogram.

(f) Noised sample. (g) Pitch Shifting - one tone lower. (h) Pitch Shifting - half tone lower. (i) Pitch Shifting - half tone higher. (j) Pitch Shifting - one tone higher.

Fig. 3: Example of explicit data augmentation techniques.

semitone as well as a lower semitone, hence we have two new spectograms. The second approach has the same reasoning, but instead of using a semitone, we use a tone. Also generating two new spectograms. Examples of this method are presented in Figures 3(g), 3(h), 3(i) and 3(j).

D. Convolutional neural network

One important aspect of using any convolutional neural network is the configuration of the network. Figure 4 presents the CNN used for the experiments presented in this work. Our CNN is similar to the one applied in [12].

The first layer is a convolutional one and its input is a patch sized 256×16, the second layer is a max pooling that downsamples the feature maps to 128×8, they are followed by another convolutional and max pooling layers, the output of this part of the network are 64 feature maps sized 64×4. The kernels used in the convolutional layers are sized 5×5. There are 2 fully-connected layers, one with 500 neurons and the last one with 10.

All the activation functions are Rectified Linear Units (ReLU) [29] excepted in the last layer, where we use Softmax, which is suitable for categorical classification. We also apply dropout [30] in the first fully connected layer during training with a probability $p = 0.5$. Our CNN was implemented with Keras and its configuration settings are summarized in Table I.

E. Combining prediction scores

In pattern recognition is common to use more than one classifier in the same system [12], [31]. This is done based on the idea of complementarity between two or more classifiers. In [12] the authors used two levels of fusion, the first one to combine CNN predictions taken from the patches, and the second one to combine predictions from Support Vector Machines (SVM) with the result of the first level.

In our work the CNN also provides predictions for each patch. Looking for the final prediction of each sample, we need to combine all predictions of the patches obtained from that

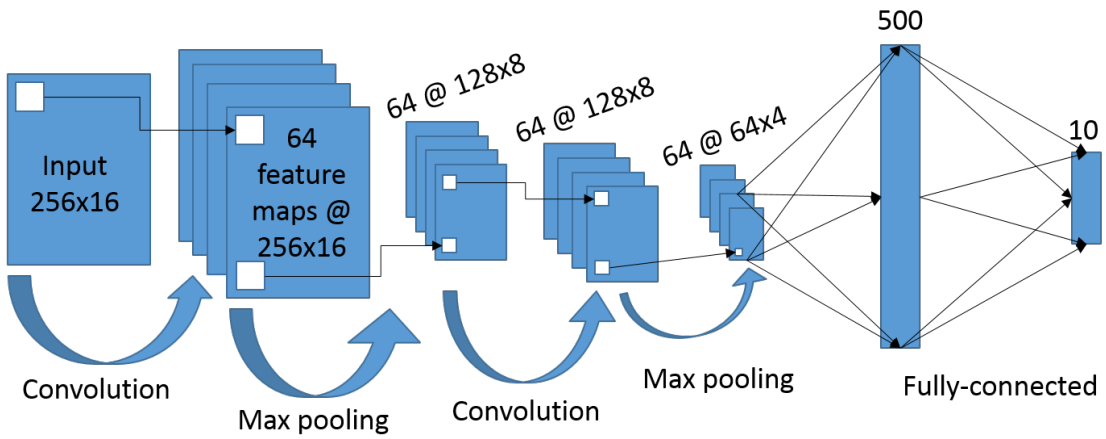


Fig. 4: An illustration of the CNN architecture.

TABLE I: Information about the CNN architecture.

Layers	Number of trainable parameters	Dimensions of the output
1º $2D$ convolution	$1,664$	$64 \times 256 \times 16$
1º $2D$ max pooling	0	$64 \times 128 \times 8$
2º $2D$ convolution	$102,464$	$64 \times 128 \times 8$
2º $2D$ max pooling	0	$64 \times 64 \times 4$
Fully connected	$8,192,500$	500
Output	$5,010$	10
Total of trainable parameters	$8,301,638$	-

sample. We have performed it by using four different fusion rules proposed in [32]: minimum, maximum, sum and product.

The sum rule is presented in Equation 1. Where x is a sample; c is the number of classes and n is the number of patches related to the same sample. $P(\omega_k|y_i(x))$ is the probability of the sample x to belong to the class k in the i-*th* patch.

$$sum(x) = \arg\max_{k=1}^{c} \sum_{i=1}^{n} P(\omega_k|y_i(x)) \quad (1)$$

The others rules, minimum, maximum and product, are similar to Equation 1, the difference is that the $\sum$ is replaced by $\min$, $\max$ and $\prod$, respectively.

IV. Experimental results

In this Section we are interested in answering the following questions with controlled experiments: (a) What is the impact of the different data augmentation strategies for the problem of music genre classification using a convolutional neural network? (b) When compared with some of the state of the art methods, what are the lessons learned in these experiments?

Table II presents the computational results for our experiments using the different data augmentation methods presented in Section III-C and the combining rules presented in Section III-E. The analysis of the results of Table II shows that, except for noise addition, all data augmentation methods improve the classification results. The best obtained accuracy of 89.45% was obtained with one tone pitch shifting.

In all experiments the sum rule presents the best accuracies. On the other hand, the lowest ones were generate by the product rule. In general, one can note that as higher is the number of samples, as lower is the accuracy when the product rule is used. It can be explained by the fact that the product rule makes the result fall apart if only one of the patches produces bad scores.

In order to verify whether or not there were statistical significance difference between the results, we have applied the Wilcoxon sum rank test (with $\alpha = 0.1$). The result of the test shows that the approach using Pitch shifting with one tone is statistically significant better than all the other approaches with the exception of the other pitch shifting strategy with half tone.

It should be noted that our results corroborate with some of the related work in the literature. In [24] some strategies for data augmentation were evaluated for the problem of Bird Species Identification. In their experimental results they found out that noise addition did not improve their computational results. For the problem of singing voice detection, Schlüter and Grill [23] have found that the use of pitch shifting, as a data augmentation strategy, provided the best results when compared to other strategies.

TABLE II: Accuracy (%) and standard deviation (σ) obtained on the experiments.

Experiment description	Augmentation factor	Acc. by min rule	Acc. by max rule	Acc. by sum rule	Acc. by prod. rule
Baseline	n	77.78% σ 1.03	50.11% σ 1.37	86.33% σ 1.20	65.44% σ 0.87
Noise addition	$2 \times n$	79.44% σ 3.10	61.22% σ 6.53	83.78% σ 5.23	33.89% σ 1.23
Loudness	$3 \times n$	81.00% σ 2.36	63.33% σ 7.92	86.44% σ 2.11	32.44% σ 0.83
Pitch shifting half tone (+1/2 tone and -1/2 tone)	$3 \times n$	84.78% σ 2.86	72.78% σ 6.06	88.33% σ 2.45	36.89% σ 0.31
Pitch shifting one tone (+1 tone and -1 tone)	$3 \times n$	83.56% σ 2.73	64.00% σ 3.07	**89.45% σ 1.97**	32.22% σ 0.96
Time stretching	$\approx 3 \times n$	81.78% σ 2.46	54.56% σ 3.20	86.78% σ 2.06	31.00% σ 1.67

V. Conclusion

This work started from the hypothesis that the use of data augmentation may improve the performance of music genre classification based on ConvNets. The experimental results with four different data augmentation in the Latin music database has shown promising results.

The best result obtained in the Latin Music Database (89.45%) was obtained by using the one tone shifting data augmentation strategy and the sum rule. The baseline result without using data augmentation was of 86.33%. It should be noted that all of the four data obtained better results than the baseline for the max, min and sum rules. While with the product rule the results were worse than the baseline.

In future research, we intend to explore other techniques of audio data augmentation, such as pitch spiral [33] and Gaussian noise [23]. We also plan to combine different types of hand-crafted features with non hand-crafted features, such as the ones used in this work.

Another research direction is to explore the use of data augmentation to different audio problems such as birds species classification and acoustic events recognition.

Acknowledgments

We would like to thank the State University of Maringá and the Pontifical Catholic University of Paraná for the structure provided. This research has been partially supported by the Brazilian Research-support agencies Coordination for the Improvement of Higher Level Personnel (CAPES), The National Council for Scientific and Technological Development (CNPq) and Araucária Foundation.

References

[1] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal music mood classification using audio and lyrics," in *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*. IEEE, 2008, pp. 688–693.

[2] X. Hu, J. S. Downie, and A. F. Ehmann, "Lyric text mining in music mood classification," *American music*, vol. 183, no. 5,049, pp. 2–209, 2009.

[3] G. Yu and J.-J. Slotine, "Audio classification from time-frequency texture," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 1677–1680.

[4] M. A. Domingues, F. Gouyon, A. M. Jorge, J. P. Leal, J. Vinagre, L. Lemos, and M. Sordo, "Combining usage and content in an online recommendation system for music in the long tail," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 1, pp. 3–13, 2013.

[5] A. Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2643–2651.

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

[7] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *Journal of new music research*, vol. 32, no. 1, pp. 83–93, 2003.

[8] T. Lidy and A. Rauber, "Evaluation of feature extractors and psychoacoustic transformations for music genre classification," in *ISMIR*, 2005, pp. 34–41.

[9] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 282–289.

[10] Y. M. G. Costa, L. S. Oliveira, A. L. Koericb, and F. Gouyon, "Music genre recognition using spectrograms," in *2011 18th International Conference on Systems, Signals and Image Processing*, June 2011, pp. 1–4.

[11] Y. M. G. Costa, L. Oliveira, A. Koerich, F. Gouyon, and J. Martins, "Music genre classification using LBP textural features," *Signal Processing*, vol. 92, no. 11, pp. 2723 – 2737, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168412001478

[12] Y. M. G. Costa, L. S. Oliveira, and C. N. S. Jr., "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 28 – 38, 2017.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[14] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2. IEEE, 2012, pp. 357–362.

[15] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics." in *ISMIR*, 2012, pp. 403–408.

[16] T. Nakashika, C. Garcia, and T. Takiguchi, "Local-feature-map integration using convolutional neural networks for music genre classification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[17] J. Schlüter and S. Böck, "Musical onset detection with convolutional neural networks," in *6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic*, 2013.

[18] G. Gwardys and D. Grzywczak, "Deep image features in music information retrieval," *International Journal of Electronics and Telecommunications*, vol. 60, no. 4, pp. 321–326, 2014.

[19] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner, "The latin music database," in *ISMIR 2008, 9th International Conference on Music Information Retrieval*, 2008, pp. 451–456.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[21] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 396–404. [Online]. Available: http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[23] J. Schlüter and T. Grill, "Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[24] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, "Audio based bird species identification using deep learning techniques," in *CLEF*, 2016.

[25] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Interspeech 2016*, 2016, pp. 2982–2986.

[26] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23-27, 2007*, 2007, pp. 341–344.

[27] Y. M. G. Costa, L. Oliveira, A. Koerich, and F. Gouyon, "Music genre recognition based on visual features with dynamic ensemble of classifiers selection," in *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, July 2013, pp. 55–58.

[28] J. George and L. Shamir, "Computer analysis of similarities between albums in popular music," *Pattern Recognition Letters*, vol. 45, pp. 78 – 84, 2014.

[29] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[31] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, "Combining visual and acoustic features for music genre classification," *Expert Systems with Applications*, vol. 45, pp. 108 – 117, 2016.

[32] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar 1998.

[33] V. Lostanlen and C. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 612–618.