Augmentation Methods on Monophonic Audio for Instrument Classification in Polyphonic Music

Agelos Kratimenos*
School of ECE, NTUA, Athens, Greece
ageloskrat@yahoo.gr

Kleanthis Avramidis*
School of ECE, NTUA, Athens, Greece
kle.avramidis@gmail.com

Christos Garoufis
School of ECE, NTUA, Athens, Greece
cgaroufis@mail.ntua.gr

Athanasia Zlatintsi
School of ECE, NTUA, Athens, Greece
nzlat@cs.ntua.gr

Petros Maragos
School of ECE, NTUA, Athens, Greece
maragos@cs.ntua.gr

Abstract—Instrument classification is one of the fields in Music Information Retrieval (MIR) that has attracted a lot of research interest. However, the majority of that is dealing with monophonic music, while efforts on polyphonic material mainly focus on predominant instrument recognition. In this paper, we propose an approach for instrument classification in polyphonic music from predominantly monophonic data that involves performing data augmentation by mixing different audio segments. A variety of data augmentation techniques focusing on different sonic aspects, such as overlaying audio segments of the same genre, as well as pitch and tempo-based synchronization, are explored. We utilize Convolutional Neural Networks for the classification task, comparing shallow to deep network architectures. We further investigate the usage of a combination of the above classifiers, each trained on a single augmented dataset. An ensemble of VGG-like classifiers, trained on non-augmented, pitch-synchronized, tempo-synchronized and genre-similar excerpts, respectively, yields the best results, achieving slightly above 80% in terms of label ranking average precision (LRAP) in the IRMAS test set.

Index Terms—instrument classification, audio mixing, data augmentation, deep learning, ensemble learning

I. Introduction

With the term "music" we refer to interesting combinations of one or more instrument sounds and sometimes vocals. Humans are able, due to their natural ability, to identify these sounds when they listen to music. Of course, this ability gets harder depending on the number of instruments, the performance style and also when perceptually similar instruments play simultaneously. It is easy to infer that for computers and computing algorithms this task gets even more complex, which has led to increased research activity in various fields of MIR.

Instrument Classification (IC) on monophonic data has already been successful. On the other hand, the equivalent polyphonic task is much harder, because of the existence of similar instruments and the superposition of time-frequency features, such as pitch and timbre, from different instruments. However, this task is being thoroughly researched, as it can lead to significant advances and applications. The ability to determine which instruments are playing at each time can provide useful insight into musical structure and therefore
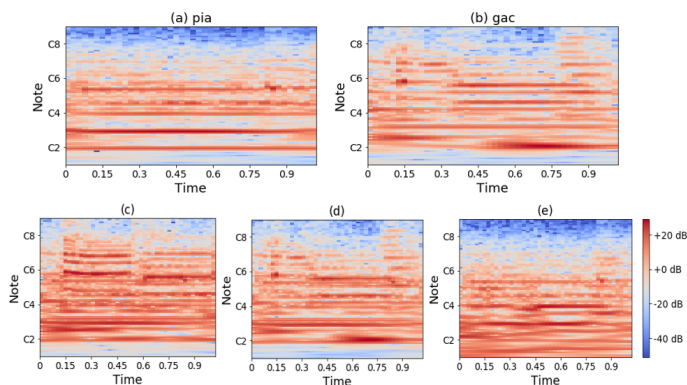

Fig. 1. CQT spectrograms of 2 segments derived from piano and guitar. (a) Piano, (b) Acoustic Guitar, (c) Random Mix, (d) Tempo Mix, (e) Pitch Mix

can act in assistance to a variety of fundamental MIR tasks, such as music browsing [1], auto-tagging [2], music automatic transcription [3] and source separation [4].

An important research constraint for instrument classification is the lack of easily accessible data, since it is quite hard to create a large dataset of polyphonic audio that is correctly annotated. On the other hand, monophonic data can be easily accessed, collected, and automatically annotated. Inspired by this, in this paper we investigate how to efficiently augment predominantly monophonic data in order to train an instrument classifier for polyphonic music, thus focusing on mixing methods. To the best of our knowledge, this is the first study that focuses on investigating methodologies for mixing monophonic audio in order to create a dataset suitable for polyphonic music tasks. As mixing methods, we investigate superimposition between perceptually similar audio segments, as well as superimposition of audio segments after alignment on the temporal or frequency scale. A secondary contribution of this work is a comparative study between the architecture proposed in [14] for the task of IC and deeper, VGG-like [23] architectures, such as the one used in [19], indicating that a deeper architecture is necessary to integrate the required features for successful instrument classification.

The rest of this paper is organized as follows: Section II provides a review of related work in the recent literature. The

---

*The first two authors contributed equally.

proposed augmentation methods, used to convert our initial dataset into a polyphonic one, are analyzed in Section III. Section IV deals with the experimental setup, describing the preprocessing and the model architectures we exploit, while also introducing the evaluation metrics utilized. In Section V we present the results of our experiments and evaluate the augmentation strategies. Finally, in Section VI we conclude our remarks and propose plausible future directions.

II. RELATED WORK

A. Instrument Classification

Traditionally, research work using monophonic audio focused mainly on recorded isolated notes, using for instance cepstral coefficients and temporal features [5] or phase information with MFCCs [6]. For an overview on note-level instrument recognition, readers are referred to [7].

Until recently, the majority of works on instrument classification used datasets of solo recordings or excerpt-level annotations (e.g. IRMAS [10]). In order to apply instrument classification results to multi-timbral audio segments, a number of research studies utilized synthesized polyphonic audio from recorded single tones [8] or MIDI files [13], [30]. In [9] hand-crafted audio features along with dimensionality reduction techniques were used in multi-instrument segments, while in [33] and [34], spectral regions that contain isolated information about the presence of certain instruments in polyphonic signals are detected and extracted.

Since labeling the presence of instruments for each frame requires a lot of effort and time, few projects deal with segment-level analysis [32] and datasets with frame-level instrument labels have only emerged over the recent few years [11], [12]. Among state-of-the-art efforts, Hung et al. [13], [30] exploit musical traits like timbre and pitch to make frame-level predictions using MusicNet [12]. In [14], Gururani et al. experiment with various temporal resolutions in the classification task, while in [24], the usage of an attention layer was shown to improve classification results when applied to a set of features extracted from a pre-trained VGG net.

B. Data Augmentation Techniques on Speech and Music

Data augmentation has generally been reported to prevent overfitting issues, something that has resulted in numerous related experiments in music recognition recently. Frequently used data augmentation techniques in MIR and speech recognition include addition of noise [25], [27], [28], simulation of spatial effects [25], pitch shifting [26], [27], time stretching [26], [27], and amplitude shifting [27][1], while in [16] an augmentation policy that includes feature warping and masking on spectrograms has provided state-of-the-art results. The work that is closest to ours, with regards to the type of augmentation performed, is that of Lee et al. [29], where vocal and instrumental audio segments are mixed together by virtue of their similarity with regards to tempo and key.

---

[1]Open-source packages that support a number of these transformations include muda [15] and rubberband (https://breakfastquay.com/ rubberband)
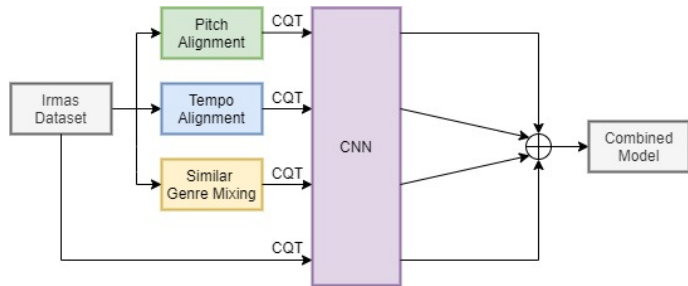

Fig. 2. Proposed pipeline. The best model occurs from an aggregation of the individual models produced by each mixing method.

III. METHODOLOGY

A. Initial Dataset

The dataset used in order to train our models is the IRMAS dataset [10], which is divided into a training and a testing set. We use the training set, which consists of 6705 3-sec audio snippets at 44.1kHz, annotated with one of the 11 available predominant instrument labels: piano (pia), acoustic guitar (gac), electric guitar (gel), violin (vio), cello (cel), saxophone (sax), clarinet (cla), trumpet (tru), human voice (voi), organ (org) and flute (flu). We choose to cut each track into 1-sec segments, since this temporal resolution yielded the optimal results in [19]. Our experiments show that the imbalance between certain classes is not an important factor during classification, so we do not consider further balancing.

B. Augmentation Methods

We apply a variety of mixing methods to our dataset, in order to make it polyphonic and of increased size, thus capable of yielding better estimations. We consider a naive random mixing method, mixing audio segments of the same music genre, thus exploiting audio similarity akin to [29], and mixing audio segments after synchronization over their pitch or tempo. These features are chosen since they encapsulate most of the meaningful information of an audio track. Moreover, we expect that by synchronizing audio tracks over certain features, a classifier would be able to isolate better other meaningful attributes from an audio mixture. Here, we experiment with mixing track pairs from each class, resulting in 55 additional classes. The mixes are implemented using PyDub.

1) Random Mixing: For each combination of instruments we define the class with the fewer audio segments, shuffle the samples of both sets and simply overlay them one by one. As a result, we produce a new dataset of mixed tracks without any repetition. Thus, every new dataset will have as many tracks as the smaller of the 2 initial ones.

2) Pitch-Sync Mixing: The intuition behind the specific augmentation strategy is that, by pitch alignment of the over-lapped segments, we force our network to focus on specific spectral and timbral characteristics. In order to align two audio segments, we first use the CREPE pitch prediction model [20], which estimates the pitch of an audio signal using 10 msec windows. After estimating the main frequencies f_1 and f_2 of the two segments, we calculate the required frequency

TABLE I
INITIAL AND PROPOSED MODEL ARCHITECTURE. DEPTH:
$d_i = 64, 128, 256, 640$, MAX POOLING: $p_i = (2, 2), (2, 2), (3, 3), (3, 3)$

	Initial 1-Conv Model [14]	Proposed 2-Conv Model
×4	Conv2D $(3 \times 3, d_i)$	2×Conv2D $(3 \times 3, d_i)$
	Batch Normalization	
	ELU	LeakyReLU $(a = 0.3)$
	Max Pooling (p_i)	
	Dropout (0.2)	
	Dense (1024)	
	ELU	LeakyReLU $(a = 0.3)$
	Batch Normalization	
	Dropout (0.5)	
	Dense (11)	
	Sigmoid Activation	

TABLE II
COMPARISON OF MODEL ARCHITECTURES

Mixing Dataset	Model	LRAP	AUC
Monophonic	Gururani et al. [14]	0.738	0.839
	Proposed	**0.767**	**0.843**
Random	Gururani et al. [14]	0.750	0.853
	Proposed	**0.776**	**0.861**

shift per segment, in semitones $s = 12 \log_2(f_1/f_2)$. We then smooth the frequency shift vector. Initially, we apply a median filter, covering 90ms, to the initial vector. Afterwards, we merge the neighboring segments of the same or similar (± 1) semitone shifts and those with duration less than 70ms with their preceding segments. Finally, we shift the second segment by the computed semitone shifts using pyrubberband. After overlaying track pairs using the procedure described above, we end up with 55 sets of mixed tracks without repetition.

3) Tempo-Sync Mixing: We proceed in tempo-syncing since instruments often play in the same beat when they co-perform. For this method, we need to calculate the tempo, in Beats per Minute (BPM), for each audio track pair that will be overlaid. The 1-sec resolution however is not adequate for BPM detection, so we exploit the initial 3-sec IRMAS tracks and cut them into 1-sec segments after mixing. For a pair of tracks to be mixed, we detect their BPM using Librosa [18] and we apply time stretching using pyrubberband. Since the duration of the stretched segment will change, we either cut the segment or repeat its beginning, in order to set its duration to 3 sec. The final tracks are mixed with PyDub and cut afterwards.

4) Similar Genre Mixing: Since the audio segments in IRMAS are already annotated with genre labels, we experiment with combining segments that belong to the same genre. The available categories are classic, pop/rock, jazz/blues and country/folk. We produce the mixed tracks as in Random Mixing, by matching only segments from the same genre category. By doing so, we expect to better approximate actual music tracks, each of which relates to a specific genre.

IV. EXPERIMENTAL SETUP

A. Pre-Processing

We produce a new set of music segments by applying each of the above augmentation methods. For each set, every track is downsampled to 22.05 kHz, downmixed to mono and also normalized by the root mean square energy. We experiment with Constant Q Transform (CQT) as a feature representation of the audio segments, as it has been proven adequate for various MIR tasks, serving a perceptually motivated frequency scale [21]. Each segment is transformed into a CQT-spectrogram, using 96 bins, 12 bins per octave and 256 hop length, resulting in a 96×87 matrix.

The set of spectrograms is then partitioned into 5 subsets, each containing an equal amount of samples from every class. We use this scheme in order to perform multiple training sessions, using each subset in rotation as validation data and the remaining 4 for fitting the classifier. The advantage of this training method is that every sample is exploited for both training and validation, thus resulting in more representative estimations and further statistical inference.

B. Model Architecture and Training

As a baseline model we use the network proposed by Gururani et al. [14], which consists of four convolutional layers and two dense layers. Every convolutional layer uses zero padding and is followed by Batch Normalization, Max Pooling and Dropout layers. ELU activation is used throughout the model and a sigmoid activation is chosen for the output layer, as it gives a probability distribution around all potential classes. We further experiment with the architecture, proposing a model that uses double layers at each convolutional block of the model, while also using leaky-ReLU [17] as the activation function. Both architectures are presented in Table I.

We train both networks using binary cross-entropy loss. Adam optimizer is used to optimize the loss function, with an initial learning rate of 0.0001 and 10% decay rate per epoch, while the batch size is set to 128. We perform Learning Rate Reduction and Early Stopping by monitoring the validation loss with a patience of 5 and 7 epochs respectively.

C. Evaluation Metrics

We evaluate the proposed models at the IRMAS test set, consisting of 2355 music tracks. Since our models are trained to predict the instruments in 1-sec segments and the duration of the testing track ranges from 5 to 20 sec, we average the per-frame predictions and extract a single total prediction for each track. While this would normally result in flawed predictions, the fact that each labeled instrument is active for the whole duration of the track allows the above manipulation. In the following, we utilize the next three metrics:

1) Label Ranking Average Precision (LRAP): The LRAP metric has been proposed in [22] and is suitable for multi-label classification evaluation. LRAP is threshold-independent for recognition tasks and measures the classifier's ability to assign higher scores to the correct labels associated to each sample, compared to the false ones. Formally, given n test samples, a binary-indicator matrix y of their ground truth labels and the score f of each label, LRAP is defined as:

$$\text{LRAP}(y, f) = \frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{\|y_i\|_0} \sum_{j:y_{ij}=1} \frac{\ell_{ij}}{\text{rank}_{ij}},$$

TABLE III

Metrics\Mix Methods	Monophonic	Random	Genre	Tempo	Pitch	Combined
LRAP	0.767 ± 0.008	0.776 ± 0.006	0.774 ± 0.005	0.790 ± 0.003	0.795 ± 0.002	**0.805 ± 0.002**
Mean AUC	0.843 ± 0.002	0.861 ± 0.001	0.862 ± 0.003	0.865 ± 0.002	0.856 ± 0.001	**0.874 ± 0.001**
F_1 micro	0.616 ± 0.009	0.624 ± 0.006	0.617 ± 0.004	0.628 ± 0.003	0.635 ± 0.002	**0.647 ± 0.003**
F_1 macro	0.506 ± 0.006	0.528 ± 0.006	0.519 ± 0.004	0.531 ± 0.003	0.532 ± 0.003	**0.546 ± 0.004**

where we define $\ell_{ij} = \{k : y_{ik} = 1, f_{ik} \geq f_{ij}\}$, and also $\text{rank}_{ij} = |\{k : f_{ik} \geq f_{ij}\}|$.

2) Area Under ROC Curve (AUC): AUC is a threshold-independent metric, defined as the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. An overall accuracy is computed by averaging the AUC score for each instrument.

3) F_1 Score: F_1 is a widely used metric and will prove useful in evaluating our work compared to existing methods, as well as getting insight into possible class imbalance problems. In order to calculate an overall score, we compute the average of the per-instrument scores at both micro and macro scales.

V. RESULTS AND DISCUSSION

Comparison of Network Architectures: We first train the models using only the initial dataset, in order to get some baseline numbers. The results show that the proposed architecture significantly improves the LRAP score of the model proposed by Gururani et al. [14] (Table II) and surpasses most of the published efforts according to F1 score (Table IV). We attribute this improvement to the utilization of blocks of consecutive convolutional layers compared to [14], as well as the usage of batch normalization in the network compared to [19]. Additionally, results produced by a re-implementation of the architecture of [19] are similar to those originally reported, leading us to assume that the differences in audio preprocessing are of minor importance.

Comparison of Data Augmentation Techniques: Next, our model is trained using the randomly mixed dataset, to gain insights on the effects of mixed tracks. As shown in Table III, the model slightly improves on the monophonic one. In order to confirm that this improvement is attributed to the mixing technique itself and not the increased number of samples, we perform an additional training session on an augmented, using pitch-shifting of $\pm 2, \pm 4, \pm 6$ semitones, monophonic dataset with an equal size to the mixed one. The outcome does indeed not surpass the randomly mixed augmented dataset, although we observed a slight increase in metrics compared to the initial dataset, including a 77.2% LRAP.

We then proceed to train additional models using the data generated with the three augmentation strategies. A performance comparison between them is summarized in Table III. The strategy of mixing tracks of the same genre barely improves the result, mixing tempo-synced tracks though scores higher compared to the baseline, with a non-negligible improvement of 1.5% LRAP compared to random mixing. This indicates that leaving the tempo variations out of context helps the model identify certain instruments. This is also the case for the pitch-synced mixes, which provide 2% LRAP

TABLE IV

Models	F_1micro	F_1macro
Bosch et al. [10]	0.503	0.432
Han et al. [19]	0.602	0.503
Pons et al. [31]	0.589	0.516
Proposed - Baseline	0.616	0.506
Proposed - Combined	**0.647**	**0.546**

improvement compared to the random polyphonic baseline and ca. 3% compared to the monophonic case.

Usage of an Ensemble Classifier: The above results imply that by manipulating a musical attribute such as tempo or pitch, we produce models that identify better certain characteristics. Thus, we expect aggregated predictions of the augmentation models to produce more representative outcomes. In order to test this hypothesis, we average the predictions analyzed above for each examined combination of strategies along the same fold, resulting in 5 new sets of predictions for each combination. The best performing ensemble model (denoted as Combined) consists of aggregated predictions of all models, excluding the one trained on randomly mixed data. Its performance on the test set (see Table III) yields above 80% in terms of LRAP, outperforming the monophonic case by almost 4%, a non-negligible improvement given the complexity of IRMAS test set. Furthermore, the ensemble model compares favorably against a number of recently published experiments on IRMAS, achieving an improvement of approximately 3% in macro F_1-score, and 4% in micro F_1-score.

As an additional experiment, we train an ensemble classifier using the procedure described above, but replacing our proposed architecture with the one used by Gururani et al. [14]. This experimental setup results in a 77.9% LRAP, further indicating that proper data augmentation can improve instrument recognition results as much as architectural refinements.

Per-Instrument Results: Finally, we examine the per-instrument F_1 score, in order to get an insight into the effect of specific augmentation methods in recognition of various instrument types. The results are visualized in Fig. 3. We observe that data augmentation mainly improves the recognition results in instruments that have a primarily supporting role or tend to play in unison, such as woodwind instruments. On the other hand, recognition of naturally predominant instruments such as the electric guitar or the violin, and vocals is not improved, or even becomes more difficult, through data mixing.

VI. CONCLUSIONS

In this paper we investigated the problem of using monophonic data to train a polyphonic instrument classifier, focusing on mixing strategies to efficiently convert the monophonic
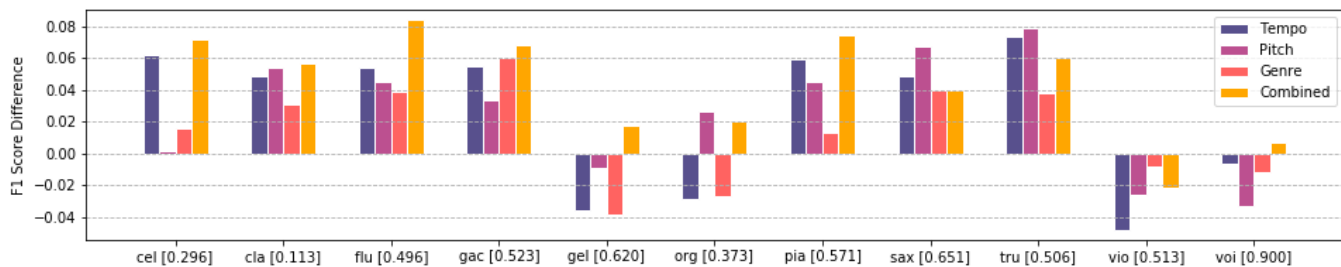
Fig. 3. Instrument-wise performance of augmentation models in terms of F_1-score difference from the baseline monophonic case, included in brackets.

dataset to polyphonic. The proposed model achieves a better interpretation of spectral representations of polyphonic audio, while the proposed mixing techniques prove to be adequate augmentation strategies, outperforming the monophonic case. Tempo and Pitch mixing strategies yield the best results, while the overall best performance is achieved by utilizing an ensemble of classifiers, inheriting different spatio-temporal characteristics from each model. Future work on this task could include mixing techniques focusing on musical traits like key alignment, as well as further investigation into multi-instrumental mixing, since preliminary experiments using random mixes of 3 instruments perform worse than the proposed strategies, indicating that more sophisticated augmentation methods and architectural refinements are required in this case.

REFERENCES

[1] K. Pati and A. Lerch, "A Dataset and Method for Electric Guitar Solo Detection in Rock Music", in Proc. AES Int. Conf. on Semantic Audio, Erlangen, Germany, 2017.

[2] J.-Y. Liu and Y.-H. Yang, "Event Localization in Music Auto-Tagging", in Proc. ACM Int. Conf. Multimedia, Amsterdam, The Netherlands, 2016.

[3] Z. Duan, J. Han, and B. Pardo, "Multi-Pitch Streaming of Harmonic Sound Mixtures", in IEEE/ACM Trans. Audio, Speech, and Language Processing, 22(1):138–150, 2014.

[4] R. Kumar, Y. Luo, and N. Mesgarani, "Music Source Activity Detection and Separation Using Deep Attractor Network", in Proc. Interspeech 2018, Hyderabad, India, 2018.

[5] A. Eronen and A. Klapuri, "Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features", in Proc. ICASSP 2000, Istanbul, Turkey, 2000.

[6] A. Diment, P. Rajan, T. Heittola, and T. Virtanen, "Modified Group Delay Feature for Musical Instrument Recognition", in 10th International Symposium on Computer Music Multidisciplinary Research (CMMR), Marseille, France, 2013.

[7] P. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic Classification of Musical Instrument Sounds", in Journal of New Music Research, 32(1):3–21, 2003.

[8] T. Heittola, A. Klapuri, and T. Virtanen, "Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation", in Proc. ISMIR 2009, Kobe, Japan, 2009.

[9] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps", EURASIP Journal on Applied Signal Processing, pp. 1-15, 2007.

[10] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals", in Proc. ISMIR, Porto, Portugal, 2012.

[11] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research", in Proc. ISMIR 2014, Taipei, Taiwan, 2014.

[12] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning Features of Music from Scratch", in Proc. ICLR 2017, Toulon, France, 2017.

[13] Y. Hung and Y. Yang, "Frame-Level Instrument Recognition by Timbre and Pitch", in Proc. ISMIR 2018, Paris, France, 2018.

[14] S. Gururani, C. Summers, and A. Lerch, "Instrument Activity Detection in Polyphonic Music Using Deep Neural Networks", in Proc. ISMIR 2018, Paris, France, 2018.

[15] B. McFee, E. Humphrey, and J. Bello, "A Software Framework for Musical Data Augmentation", in Proc. ISMIR 2015, Malaga, Spain, 2015.

[16] D. Park et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition", in Proc. Interspeech 2019, Graz, Austria, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", in Proc. CVPR 2015, Boston, MA, 2015.

[18] B. McFee et al., "librosa: Audio and Music Signal Analysis in Python", in Proc. 14th Python in Science Conference, Austin, TX, USA, 2015.

[19] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music", in IEEE/ACM Trans. Audio, Speech, and Language Processing, 25(1):208 – 221, 2017.

[20] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A Convolutional Representation for Pitch Estimation", in Proc. ICASSP 2018, Calgary, AL, Canada, 2018.

[21] C. Schoerkhuber and A. Klapuri, "Constant-Q Transform Toolbox for Music Processing", in Proc. SMC 2010, Barcelona, Spain, 2010.

[22] R. E. Schapire and Y. Singer, "Boostexter: A Boosting-Based System for Text Categorization", in Machine learning, 39(2-3):135–168, 2000.

[23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in Proc. ICLR 2015, San Diego, CA, USA, 2015.

[24] S. Gururani, M. Sharma, and A. Lerch. "An Attention Mechanism for Musical Instrument Recognition", in Proc. ISMIR 2019, Delft, the Netherlands, 2019.

[25] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, M. Müller, and H. Lukashevich, "Investigating CNN-based Instrument Family Recognition for Western Classical Music Recordings", in Proc. ISMIR 2019, Delft, The Netherlands, 2019.

[26] A. Cohen-Hadria, A. Roebel, and G. Peeters, "Improving Singing Voice Separation Using Deep U-Net and Wave-U-Net with Data Augmentation", in Proc. EUSIPCO 2019, A Coruna, Spain, 2019.

[27] M. Airaksinen, L. Juvela, P. Alku, and O. Räsänen, "Data Augmentation Strategies for Neural Network F0 Estimation", in Proc. ICASSP 2019, Brighton, UK, 2019.

[28] L. Rafael Aguilar, MG Yandre Costa, and N. Carlos Silla, "Exploring Data Augmentation to Improve Music Genre Classification with Convnets", in Proc. IJCNN 2018, Rio de Janeiro, Brazil, 2018.

[29] K. Lee and J. Nam, "Learning a Joint Embedding Space of Monophonic and Mixed Music Signals for Singing Voice", in Proc. ISMIR 2019, Delft, the Netherlands, 2019.

[30] Y. Hung, Y. Chen and Y. Yang, "Multitask Learning for Frame-level Instrument Recognition", in Proc. ICASSP 2019, Brighton, UK, 2019.

[31] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre Analysis of Music Audio Signals with Convolutional Neural Networks", in Proc. EUSIPCO 2017, Kos, Greece, 2017.

[32] S. Essid, G. Richard, and B. David, "Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies", in IEEE Transactions on Audio, Speech, and Language Processing, 14(1):68-80, 2005.

[33] D. Giannoulis and A. Klapuri, "Musical Instrument Recognition in Polyphonic Audio using Missing Feature Approach", IEEE Transactions on Audio, Speech, and Language Processing 21(9):1805-1817, 2013.

[34] J.G.A. Barbedo and G. Tzanetakis, "Musical Instrument Classification using Individual Partials", IEEE Transactions on Audio, Speech, and Language Processing 19(1):111-122, 2010.