Ensemble size classification in Colombian Andean string music recordings

Sascha Grollmisch[1,2], Estefanía Cano[2], Fernando Mora Ángel[3] and Gustavo López Gil[3] *

[1] Institute of Media Technology, TU Ilmenau, Ilmenau, Germany
[2] Semantic Music Technologies, Fraunhofer IDMT, Ilmenau, Germany
[3] Valores Musicales Regionales, Universidad de Antioquia, Medellín, Colombia
sascha.grollmisch@idmt.fraunhofer.de

Abstract. Reliable methods for automatic retrieval of semantic information from large digital music archives can play a critical role in musicological research and musical heritage preservation. With the advancement of machine learning techniques, new possibilities for information retrieval in scenarios where ground-truth data is scarce are now available. This work investigates the problem of counting the number of instruments in music recordings as a classification task. For this purpose, a new data set of Colombian Andean string music was compiled and annotated by expert musicologists. Different neural network architectures, as well as pre-processing steps and data augmentation techniques were systematically evaluated and optimized. The best deep neural network architecture achieved 80.7% file-wise accuracy using only feed forward layers with linear magnitude spectrograms as input representation. This model will serve as a baseline for future research on ensemble size classification.

Keywords: Ensemble Size Classification, Music Archives, Music Ensembles, Andean String Music.

1 Introduction

This work is motivated by the need of robust information retrieval techniques capable of efficiently extracting semantic information from large digital musical archives. With the advancements of deep learning techniques, numerous music information retrieval (MIR) methods have been proposed to address different information retrieval tasks, predominantly from a supervised machine learning perspective. In this work, we focus on the task of determining the size of musical ensembles, and aim to automatically classify music recordings according to the number of instruments playing in the track: solo, duet, trio, quartet, etc. Our long-term goal is to develop methods that minimally rely on manually annotated

*This work has been partially supported by the German Research Foundation (BR 1333/20-1).

data, and that can exploit commonalities between unlabeled data and the few annotations available (semi-supervised and few-shot learning). This will enable the usage of MIR techniques not only with archives of mainstream music, but also with non-western, under-represented, folk and traditional music archives. As described in section 2, not much work has been conducted on the topic of ensemble size classification in music. Consequently, this work focuses on systematically optimizing a baseline classification model in a fully supervised manner (see section 3) that can serve as a building block for future research on this topic. Detailed descriptions of the data set used and the optimization steps taken are presented in sections 3.1 and 3.2, respectively. Conclusions are presented in section 4, outlining possibilities to extend this work to semi-supervised and few-shot learning paradigms.

1.1 The ACMus Project

This research work was conducted in the context of the ACMus research project: *Advancing Computational Musicology - Semi-supervised and unsupervised segmentation and annotation of musical collections*[1]. The main goal of the project is to improve upon the limits of state-of-the-art machine learning techniques for semantic retrieval of musical metadata. In particular, ACMus focuses on leveraging semi-supervised and unsupervised techniques for segmentation and annotation of musical collections. The music collection in the *Músicas Regionales* archive at the Universidad de Antioquia in Medellín, Colombia is the focus of this research. The archive contains one of the most important collections of traditional and popular Colombian music, including music from the Colombian Andes, indigenous traditions, Afro-Colombian music, among others. The great diversity of the archive in terms of musical traditions, audio quality and formats (analogue, digital, field recordings), and musical sources (instrumental, vocal, speech, mixed), makes it a particularly challenging collection to work with. Besides developing methods for ensemble size classification, the ACMus project will also focus on developing methods for speech/music discrimination, meter recognition, and musical scale detection. The ACMus Project is a collaboration between Fraunhofer IDMT and Ilmenau University of Technology in Germany, and Universidad de Antioquia and Universidad Pontificia Bolivariana in Colombia.

2 Related work

To the best of the authors' knowledge, automatically determining the size of musical ensembles is a vastly unexplored topic in MIR research, and no state-of-the-art methods for the task have been proposed. Therefore, this section highlights source counting methods proposed in related fields such as polyphony estimation and speaker counting.

[1] `https://acmus-mir.github.io/`

2.1 Speaker Counting

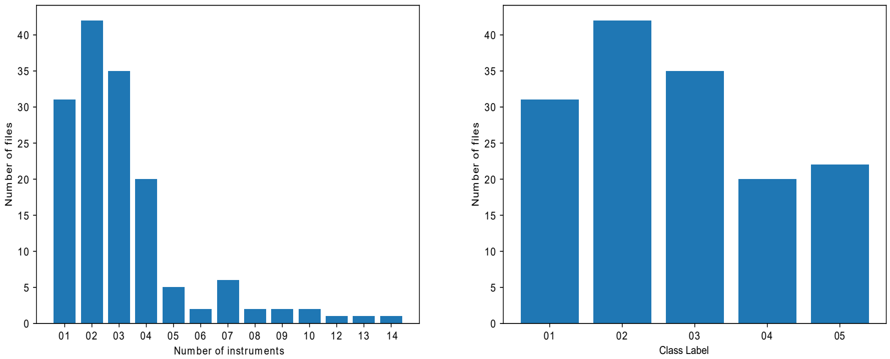
While a considerable amount of work on the topic of speaker counting for single channel recordings has been conducted, the problem has often been approached from a feature design perspective where features are specifically engineered to work with speech signals [10]. Works using more generic features such as [14][1] often assume that for the most part, only one speaker is active in the recording at a given time instant. In the case of music signals, this would be a strong assumption since musical instruments are expected to play simultaneously.
The task of audio source counting can be seen either as a regression or a classification problem when the number of maximum sources to be expected is known. In [12], the authors investigate the performance of both approaches for speaker counting using bi-directional long-short term memory neural networks (BLSTMs) with different input representations such as the linear magnitude spectrogram, the mel-scaled spectrogram, and the Mel Frequency Cepstral Coefficients (MFCCs) with linear magnitude spectrogram performing best. The data set comprised 55 hours of synthetically generated training material including signals with up to ten speakers. The system was tested on 5720 unique and unseen speaker mixtures. Even though regression could appear to be a good choice since the direct relationship of neighbouring classes is learned as well (a signal with 2 sources is closer to a signal with 3 sources than to a signal with 5), classification performed better. Based on these results, the classification approach was used in this work.

2.2 Polyphony Estimation

Polyphony estimation refers to the task of counting the number of simultaneous notes played by one or several instruments in a music recording. This can be used as a pre-processing step for multi-pitch estimation. It is important to note that polyphony estimation does not directly translate into ensemble size estimation, as several notes can be simultaneously played by a single instrument such as the guitar. Nevertheless, some relevant work on this topic is described here. Using a CNN with constant-Q transform of the audio data, the method in [2] achieved state-of-the-art performance for multi-pitch estimation. Large losses in accuracy were caused in particular by instruments playing closely harmonically related content. The authors in [6] examine this task separately with different classical instruments playing up to four simultaneous notes. Using training data of 22 minutes the proposed CNN architecture with mel-scaled spectrogram achieved a mean accuracy of 72.7% for a small evaluation set of only three songs.

3 Proposed Method for Ensemble Size Classification

Since no method has been proposed in the literature that could directly be applied to identify the number of instruments in Andean string music recordings, we focus on developing a baseline model systematically evaluated and optimized



(a) Class distribution in data set

(b) Large ensembles mapped to class *5*

Fig. 1: Distribution of the annotated classes in the data set. (a) Number of files per ensemble size. (b) Final class distribution with all large ensembles mapped to class 5.

using different neural network architectures. Since neural networks achieve state-of-the-art performance in related fields, as well as in other MIR tasks such as instrument recognition [5][4], other types of supervised classifiers were not evaluated. In this study, no pre-trained models were used as we wish to build a baseline that shows the potential of different neural networks for unseen tasks, avoiding possible biases from other data sets previously used for training.

3.1 Data set

For this study, 150 representative song fragments from the *Músicas Regionales* archive were selected and annotated by at least two experts per song in Universidad de Antioquia. All the songs are instrumental pieces without vocals, performed by ensembles of plucked string instruments from the Andes region in Colombia. The instruments in the data set include different kinds of acoustic guitars, bandolas, tiples, electric bass guitars, and occasionally percussion instruments such as the maracas. The ensembles sizes considered are soloist, duet, trio, quartet, and large ensembles (five or more instruments). The annotations in the data set include the ensemble size, as well as the list of all the instruments in the ensemble.

In most songs, all annotated instruments are active during the entire file; however, short sections where one instrument is temporarily inactive also occur, leading to some instances of weak labels. The data set comprises 54 minutes of audio, with song fragment duration ranging from 7 to 62 seconds. The distribution of the classes is shown in figure 1. Songs containing five or more instruments were mapped to the class 5. No genre, composer or tempo bias was found in the class distribution. Given that the original source of the recordings include digitized versions of tape recordings as well as more recent digital recordings, these

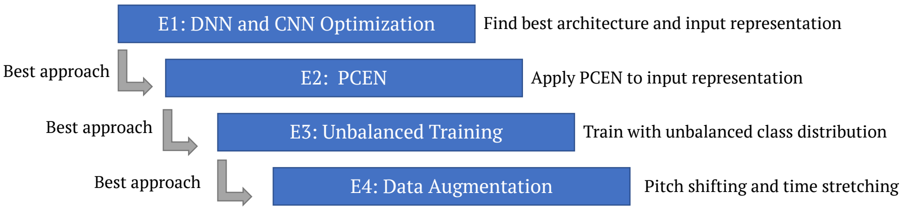

E1: DNN and CNN Optimization
Find best architecture and input representation
Best approach
E2: PCEN
Apply PCEN to input representation
Best approach
E3: Unbalanced Training
Train with unbalanced class distribution
Best approach
E4: Data Augmentation
Pitch shifting and time stretching

Fig. 2: Overview of the experimental setup. Four consecutive experiments (E1-E4) were performed to find the optimal architecture for our task.

files have been saved with a 96 kHz sampling rate, 24 bit-depth, and in stereo format. However, for monophonic analogue recordings, the stereo was obtained by duplicating the monophonic recording in both channels. Additionally, some of the older recordings only contain information below 8 kHz. To avoid biases during training, all files were downsampled to 12 kHz (to avoid sub-sampling artifacts), mixed to mono, and normalized to a maximum absolute amplitude of 1 for all the experiments.

3.2 Experimental setup

Four experiments were conducted in order to build a reliable baseline system, showing the upper boundaries for a fully supervised classification system with a neural network trained from scratch. As shown in figure 2, our work flow starts with Experiment 1 (E1), where different architectures and input representations are evaluated. The approach that shows best performance in E1 is then used in Experiment 2 (E2) to test the effects of per-channel energy normalization (PCEN) on the system. Similarly, E3 and E4 evaluate the effects of unbalanced training and data augmentation, respectively, on the best model from the previous experiment. In all our experiments, we performed 20 repetitions of random data set splits for testing all files and accounting for randomness during training of the networks. In each step, 70% of the files were randomly picked for training, 10% for early stopping during training, and 20% for evaluating the performance on unseen data. The test set was always balanced using the class with the smallest number of files and randomly subsampling the other classes.

Each network was trained for 500 epochs unless the validation loss stopped decreasing for 100 epochs. The Adam optimizer [7] with a learning rate of 0.001, Glorot initialization [3], categorical cross-entropy loss, and ReLU activation function (except softmax activation for the output layer), were used for all networks. For all experiments, the input representations were normalized to zero mean and standard deviation of one. The normalization values were calculated on the training set and applied to the validation and test sets. All experiments were conducted using Tensorflow.[2]

Experiment 1 (E1) - DNN and CNN models: E1 aimed at finding the best model architecture and input representation for a feed-forward neural network (DNN), and a convolutional neural network (CNN). Bayesian Optimization [11] was used to obtain an optimal combination of hyper-parameters and comparable results for all network architectures in a reasonable amount of time.[3]

As input features, a linear magnitude spectrogram obtained from the short-time Fourier transform (STFT) was compared to the mel-scaled spectrogram with a logarithmic frequency axis (Mel) using 128 mel bands.[4] For the DNN model, the spectral frames were smoothed using a moving average filter over time for each frequency bin to highlight stable structures over several time frames while keeping the same frequency resolution and input dimensionality. The length of the averging filter, STFT size, number of layers, number of units per layer, and dropout percentage between the layers were also subject to the Bayesian optimization. For the CNN model, several time frames were combined into patches, where the patch length was also optimized. The maximum patch duration was set to 3 seconds. The basic CNN architecture was inspired by the model proposed in [5] and the number of layers and filters, amount of Gaussian noise added to input, and dropout percentage between the layers were included in the optimization. The Bayesian optimization process was performed with 30 iterations and was only feasible because of the relatively small data set (see Section 3.1).

Experiment 2 (E2) - Per-Channel Energy Normalization (PCEN): In E2, the best architectures obtained in E1 were taken, and per-channel energy normalization (PCEN)[4] was applied to each audio file. PCEN suppresses stable background noise using adaptive gain control and dynamic range compression. This has proved to be beneficial for tasks with high loudness variations such as key word spotting [13]. In this study, PCEN was applied to test its potential to account for the great variability in audio quality in our data set. PCEN was evaluated with the default settings S1 ($power = 0.5, time\_constant = 0.4, max\_size = 1$), and with a second parameter setting S2 ($power = 0.25, time\_constant = 0.01, max\_size = 20$) experimentally chosen for highlighting harmonic structures. Figure 1a and 1b show the different input representations and PCEN settings for two audio files, one with three instruments and one with four. While S1 highlights temporal changes, S2 emphasizes harmonic structures.

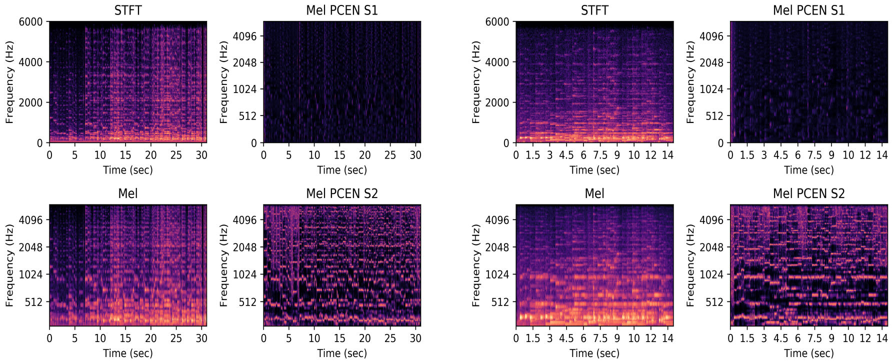
Experiment 3 (E3) - Unbalanced Training: In E1 and E2, the training data was balanced using random sub-sampling. For E3, class weights[5] were used

[2] Tensorflow (1.10): `www.tensorflow.org`

[3] Implementation from `https://github.com/fmfn/BayesianOptimization`

[4] Implementation from librosa (0.6.3): `https://librosa.github.io/`

[5] Implementation from sklearn (0.20.2): `https://scikit-learn.org/`



(a) Example with three instruments

(b) Example with four instruments

Fig. 3: Input representations for two example recordings. (a) Input representations of an example of a trio. (b) Input representations of an example of a quartet.

for training on the unbalanced data set using the best CNN and DNN models from E2. Additionally, the architectures were also evaluated using the unbalanced training set without class weights in order to determine its influence on classification performance.

Experiment 4 (E4) - Data augmentation (DA): Pitch shifting and time stretching have been previously used for audio data augmentation in tasks such as chord detection [8] and singing voice separation [9]. In E4, pitch shifting (±2 semitones), and time stretching (four steps between 90% and 110%) were applied only on the training data[4]. After data augmentation, the training set contained eight additional versions of each file.

3.3 Results

As evaluation measure, we use the mean file accuracy and standard deviation over all repetition steps. To calculate the file accuracy, the class confidences were summed up over all times frames, and the class with the highest confidence was chosen. Results are presented in Tables 1-4 and will be described in detail in the following sections. The best performing system is highlighted in bold in each table.

Experiment 1 (E1) - DNN and CNN models: Table 1 shows the results for E1. To give the reader an idea of the importance of parameter optimization, we present results for the best performing network, as well as for the worst performing one (above chance level 20%). With balanced training data and no data augmentation (E1), the highest classification accuracy (76.5%) was obtained by

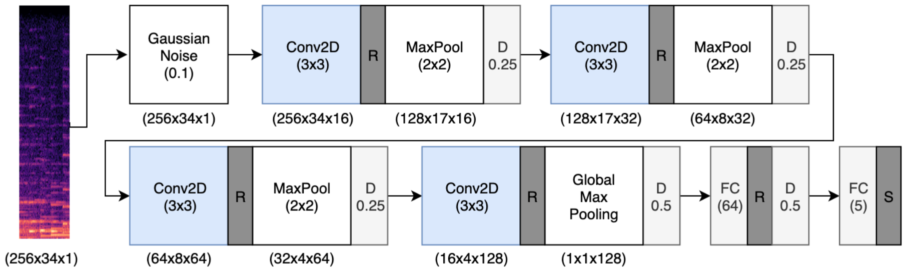


Fig. 4: Best CNN model architecture consisting of four convolutional layers (Conv2D) followed by ReLU activation (R), max pooling (MaxPool), and Dropout (D) for regularization. Global max pooling is applied before the dense layers (FC). The final dense layer uses softmax activation (S) for the classification. The corresponding output shapes are specified for each layer.

the DNN model with linear magnitude spectrogram (STFT). CNNs in general, as well as DNNs with mel-spectrogram, performed slightly worse. This suggests that with small audio training data sets, CNNs do not necessarily lead to the best performance, and that simpler and faster feed forward networks can lead to better results. Furthermore, linear magnitude spectrograms resulted in higher performance for both DNNs and CNNs. These results go in line with those reported in [12], where linear magnitude spectrogram resulted in better performance than the mel-spectrogram for speaker counting. Table 1 also shows how critical the choice of hyper-parameters is. Especially CNNs suffer when parameters are poorly chosen, leading to an accuracy of 20.5% for the worst model above chance level. Since there is so much variability in the CNNs' performance, it is possible that further optimization iterations may lead to better results and architectures than the ones found here.

Table 1: Mean accuracy, standard deviation in % for E1.

Optimization	DNN STFT	DNN Mel	CNN STFT	CNN Mel
Best	**76.5, 11.3**	72.5, 10.6	74.0, 8.7	71.3, 10.9
Worst	57.0, 14.8	65.5, 12.1	20.5, 1.6	20.5, 1.6

The final DNN model used a 2048 STFT window and hop size with logarithmic compression of the magnitudes, and a moving average filter 10 time frames long, covering in total 1.7 seconds. The 1024 unique values in the STFT were passed through a 0.1 dropout layer to one hidden layer with 512 units. The output was passed through a dropout of 0.5 to final softmax layer with 5 units, one for each class. The best CNN model is shown in detail in figure 4. The input representation was achieved from a STFT with a window and hop size of 512 samples and logarithmic compression of the magnitudes. Each patch consists of 34 STFT frames covering 1.45 seconds of audio.

Experiment 2 (E2) - PCEN: Table 2 shows the results of applying PCEN-S1 and PCEN-S2 to the input representations, as well as the best performing model from E1 (for comparison). As seen in the table, applying PCEN led to worse results when compared to E1. Between the two parameter settings of PCEN, the best results were achieved for S2 which highlights harmonic structures rather than temporal changes. In general, it appears that the suppression of possible background noise in our data when using PCEN results in the loss of discriminative information for ensemble classification. Therefore, PCEN is discarded as a processing step for the following experiments.

Table 2: Mean accuracy, standard deviation in % for E2.

PCEN	DNN STFT	DNN Mel	CNN STFT	CNN Mel
with PCEN-S1	56.0, 10.1	47.2, 9.4	60.3, 11.2	56.8, 12.3
with PCEN-S2	68.0, 10.4	67.2, 12.2	63.7, 8.3	49.2, 18.6
without PCEN (E1)	**76.5, 11.3**	72.5, 10.6	74.0, 8.7	71.3, 10.9

Experiment 3 (E3) - Unbalanced Training: Table 3 shows the results obtained with unbalanced training data, both with and without class weights. Additionally, the best performing architecture up to this point (E1) is included for comparison. The additional training data from the unbalanced training set improved the performance of all networks and lowered the standard deviation between data splits, leading to a more stable model regardless of the files chosen for training. The possible reason for the increased performance may be the increased variability of the training data since more conditions are covered in the training data set. Applying class weights led to nearly the same performance as without the weights. The reason for only having a minor impact may be that the initial data set was already nearly balanced.

Table 3: Mean accuracy, standard deviation in % for E3.

Unbalanced Training	DNN STFT	DNN Mel	CNN STFT	CNN Mel
with class weights	**80.7, 5.7**	74.8, 9.0	77.7, 7.9	73.3, 6.1
without class weights	79.7, 6.4	75.0, 9.0	77.5, 7.7	74.8, 8.8
balanced data set (E1)	76.5, 11.3	72.5, 10.6	74.0, 8.7	71.3, 10.9

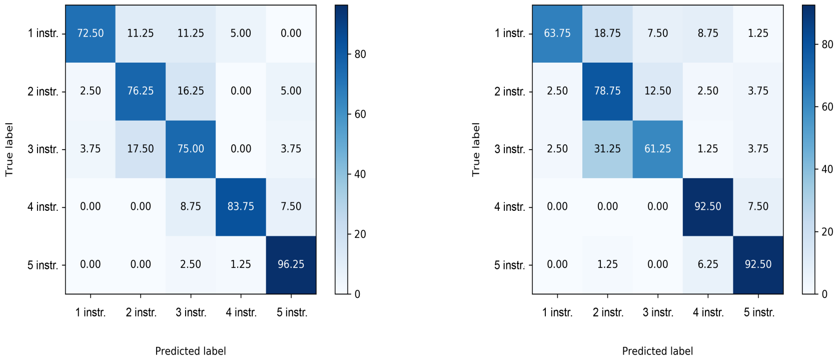
Experiment 4 (E4) - Data augmentation: Table 4 shows the results obtained with each data augmentation method, as well as the best performing architecture from E3 for comparison. Overall the best result is obtained without data augmentation using a DNN with STFT input. In contrast, the DNN model with Mel input, experiences a slight increase of performance when pitch shifting and time stretching are applied independently. Except for pure time stretching all data augmentation methods improved slightly the performance of the CNN with STFT. Using pitch shifting only led to the best CNN performance (using the STFT) with an accuracy slightly below the best DNN model. Results with the DNN go in line with those in [9] were data augmentation in small training data sets had very little impact on singing voice separation performance.

Table 4: Mean accuracy, standard deviation in % for E4.

Augmentation (DA)	DNN STFT	DNN Mel	CNN STFT	CNN Mel
with full DA	78.2, 7.7	68.5, 10.9	78.5, 6.5	77.2, 6.4
only time stretch	77.5, 7.9	78.2, 7.5	76.5, 7.1	76.2, 7.2
only pitch shift	75.2, 7.2	75.0, 9.3	80.2, 6.0	75.7, 8.2
without DA (E3)	**80.7, 5.7**	74.8, 9.0	77.7, 7.9	73.3, 6.1

3.4 Error Analysis

In order to get further insights about the classification errors of the best DNN and CNN models, figure 5 displays the mean confusion matrices for the best DNN and CNN from E3 (best overall models). Classification errors are highest between neighboring classes which shows that the network is implicitly capable of learning the relationships between classes (e.g., a duo is closer to a trio than to a quartet), and consequently, of learning useful classification features. This is in line with the findings in [6] and [12], where better performance was achieved for speaker counting with classification than with regression. It is intriguing why classification performance is relatively low for the one instrument class, which intuitively, appears to be a fairly simple classification problem. A possible explanation might be that the string instruments in our data set can simultaneously play relatively complex melodies and harmonies. This might blur the boundaries between class 1 and 2, since very similar music could alternatively be split into two different instruments. Class 5 achieved the highest classification accuracy. Since files in these class can contain up to 14 instruments, the difference between them and the other classes is probably much larger in terms of spectral content. This supports the assumption that meaningful features have been learned during training.



(a) Confusion matrix for DNN model (b) Confusion matrix for CNN model

Fig. 5: Mean confusion matrices for best models from E3.

4 Conclusions

In this work, the task of classifying the number of instruments in music recordings was addressed using a newly gathered data set of Colombian Andean string music. Apart from the challenges of the task itself, working with Andean string music comes with several difficulties: different recording conditions, scarce and expensive annotated data, and high similarity between the different instruments.

To build our baseline system, 150 tracks were annotated by expert musicologist in Colombia. Using this relatively small data set, several neural networks architectures were trained and optimized. The highest file-wise accuracy of 80.7% was achieved with a DNN, while the best CNN model attained 80.2%. Using linear magnitude spectrograms as input representation instead of its mel-scaled version, resulted in better performance in all experiments. All approaches clearly outperform the 20% chance level baseline which demonstrates the potential of this approach. In general, all networks had a minimum standard deviation of 6% between data splits, suggesting that the training set does not cover the full variance of recording conditions and instrument combinations. Neither the experiments with data augmentation using pitch shifting and time stretching nor those with PCEN showed a clear improvement in the robustness of the system. The optimization procedure showed that hyper-parameters optimization is critical when working with such a small data set. This system will serve as a baseline for future research on this topic where techniques for learning from few examples like transfer learning will be evaluated. Furthermore, techniques for incorporating unlabeled training data in a semi-supervised or unsupervised fashion will be explored.

References

1. Andrei, V., Cucu, H., Buzo, A., Burileanu, C.: Counting competing speakers in a timeframe - human versus computer. In: Interspeech Conference. ISCA, Dresden, Germany (2015)
2. Bittner, R.M., Mcfee, B., Salamon, J., Li, P., Bello, J.P.: Deep Salience Representations for F0 Estimation in Polyphonic Music. In: 18th International Society for Music Information Retrieval Conference. Suzhou, China (2017)
3. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics. Sardinia, Italy (2010)
4. Gómez, J.S., Abeßer, J., Cano, E.: Jazz Solo Instrument Classification With Convolutional Neural Networks, Source Separation, and Transfer Learning. In: 19th International Society for Music Information Retrieval Conference. Paris, France (2018)
5. Han, Y., Kim, J., Lee, K.: Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music. In: IEEE/ACM Transactions on Audio, Speech, and Language Processing. vol. 25, pp. 208–221 (jan 2017)
6. Kareer, S., Basu, S.: Musical Polyphony Estimation. In: Audio Engineering Society Convention 144. Milan, Italy (2018)

7. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations (ICLR). San Diego, USA (2015)
8. Nadar, C.R., Abeßer, J., Grollmisch, S.: Towards CNN-based acoustic modeling of seventh chords for automatic chord recognition. In: International Conference on Sound and Music Computing. Málaga, Spain (2019)
9. Prétet, L., Hennequin, R., Royo-Letelier, J., Vaglio, A.: Singing voice separation: A study on training data. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 506–510. Brighton, UK (May 2019)
10. Sayoud, H., Boumediene, T.H., Ouamour, S., Boumediene, T.H.: Proposal of a New Confidence Parameter Estimating the Number of Speakers – An experimental investigation-. Journal of Information Hiding and Multimedia Signal Processing **1(2)**(April), 101–109 (2010)
11. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian Optimization of Machine Learning Algorithms. In: 25th International Conference on Neural Information Processing Systems. pp. 2951–2959. Lake Tahoe, Nevada (2012)
12. Stoter, F.R., Chakrabarty, S., Edler, B., Habets, E.A.P.: Classification vs. Regression in Supervised Learning for Single Channel Speaker Count Estimation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 436–440. IEEE (apr 2018)
13. Wang, Y., Getreuer, P., Hughes, T., Lyon, R.F., Saurous, R.A.: Trainable frontend for robust and far-field keyword spotting. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5670–5674. IEEE (mar 2017)
14. Xu, C., Li, S., Liu, G., Zhang, Y.: Crowd ++ : Unsupervised Speaker Count with Smartphones Crowd ++ : Unsupervised Speaker Count with Smartphones. In: 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 43–52. ACM, Zurich, Switzerland (2013)