A Hierarchical Deep Learning Approach for Minority Instrument Detection

Dylan Sechet, Francesca Bugiotti, Matthieu Kowalski, Edouard D'hérouville, Filip Langiewicz

A HIERARCHICAL DEEP LEARNING APPROACH FOR MINORITY INSTRUMENT DETECTION

*Dylan Sechet, Francesca Bugiotti and Matthieu Kowalski**

CentraleSupelec, Inria, CNRS, LISN
Université Paris-Saclay,
Paris-Saclay, France
dylan.sechet@student-cs.fr
francesca.bugiotti@centralesupelec.fr
matthieu.kowalski@universite-paris-saclay.fr

Edouard d'Hérouville and Filip Langiewicz

Linkaband
Paris, France
forename@linkaband.com

ABSTRACT

Identifying instrument activities within audio excerpts is vital in music information retrieval, with significant implications for music cataloging and discovery. Prior deep learning endeavors in musical instrument recognition have predominantly emphasized instrument classes with ample data availability. Recent studies have demonstrated the applicability of hierarchical classification in detecting instrument activities in orchestral music, even with limited fine-grained annotations at the instrument level. Based on the Hornbostel-Sachs classification, such a hierarchical classification system is evaluated using the MedleyDB dataset, renowned for its diversity and richness concerning various instruments and music genres. This work presents various strategies to integrate hierarchical structures into models and tests a new class of models for hierarchical music prediction. This study showcases more reliable coarse-level instrument detection by bridging the gap between detailed instrument identification and group-level recognition, paving the way for further advancements in this domain.

1. INTRODUCTION

The identification of instruments within an audio excerpt poses an enduring challenge in the field of Music Information Retrieval (MIR). This task is inherently intricate due to the poly-instrumental nature of real-world music, where the pitches of multiple instruments often intertwine. Furthermore, the task is complicated by substantial variations in timbre and performance style among instruments, further hindering recognition endeavors. Even trained musicians may encounter perceptual similarities among specific instruments, adding another layer of complexity to the recognition process.

Instrument identification bears significant implications across various domains, including music cataloging and discovery. It aids in tasks such as song retrieval [1], facilitates genre recognition systems [2], and contributes to recommendation systems [3]. While the recognition of more common instruments benefits from abundant available data, challenges arise in genres such as orchestral or opera, as well as with rare or non-western instruments, for which data is much more scarce.

Hierarchical classification systems have been proposed to address the complexities of instrument recognition. These systems enable the prediction of instruments at various levels of specificity, demonstrating particular promise in handling imbalanced datasets and scenarios involving few-shot learning [4]. However, existing work in this domain has been confined to specific genres and a restricted set of instruments. This study assesses the scalability of hierarchical approaches on a more complex dataset, like MedleyDB, and proposes different formulations of the hierarchical problem.

1.1. Instrument detection

The field of instrument detection incorporates a diverse array of methodologies, spanning from signal processing techniques to contemporary deep learning approaches. Meanwhile, multi-label classification for audio signals has attracted considerable interest across various domains [5].

In the 2000s, Marques et al. [6] conducted instrument classification on brief $0.2s$ music excerpts utilizing Gaussian Mixture Models (GMM) and Support Vector Machines (SVM), with features extracted through Mel-Frequency Cepstral Coefficients (MFCC). In a similar vein, Essid et al. [7] showcased the advantages of GMMs over SVMs by employing MFCC features preprocessed with Principal Component Analysis (PCA), even in the context of longer mono-instrument samples.

More recently, Deep learning models, which have seen successful applications across various domains [8], have demonstrated promise in mono-instrument detection as well. Initially designed for image recognition tasks, Convolutional Neural Networks (CNNs) have been effectively repurposed to handle spectrogram-like features such as MFCCs or Constant-Q Transforms. Solanky et al. highlighted in [9] the efficiency of an AlexNet-inspired CNN model in predominant instrument recognition, while in [10], Avramidis et al. introduced performance enhancements in instrument recognition by integrating recurrent components into CNN architectures.

Attention-based models have also emerged as an promising alternative to CNNs in the field of audio classification: transformers, initially introduced for text classification [11], have been adapted to image recognition tasks [12]. Jamil et al. [13] used a vision transformer architecture for audio classification to distinguish harmless from malicious drones. In the domain of Music Information Retrieval, Regunath et al. [5] were able to outperform a CNN architecture using a vision transformer for predominant instrument recognition in polyphonic settings.

* We thank and acknowledge Agathe Gioan, Xavier Jeunot, and Aaron Broderick, for the productive discussions and the joint work that preceded this study.

1.2. Hierarchical classification for audio

Exploration of hierarchical structures for classifying audio segments has been a subject of prior research across diverse domains. For instance, hierarchical methods have been employed in classifying bird songs, with a class tree rooted in biological taxonomy [14]. Notably, these methods operated on more extended audio excerpts than the frame-level analysis.

In the realm of Music Information Retrieval, Fu et al. introduced in [15] a hierarchical approach tailored for singing voice classification and transcription. On the other hand, Essid et al. [7] explored hierarchical classification using GMMs, focusing primarily on synthetic music extracts rather than actual recordings and not at the frame level. In a recent study in 2023, Krause et al. [16] delved into hierarchical classification methods explicitly designed for orchestral and opera pieces. Their research demonstrated performance improvements, particularly in scenarios with limited fine-grained annotations. Furthermore, Garcia et al. [4] investigated hierarchical classification for few-shot learning situations, aiming to enable model adaptation to unseen classes, contrasting with the study's utilization of predefined classes.

1.3. Work on rare instrument detection

In the domain of rare audio source detection, where annotated data is scarce or nonexistent, previous research efforts have aimed to address this challenging task. Various strategies have been explored in the domain of few-shot learning, including hierarchical methodologies akin to those proposed by Garcia et al. [4], along with approaches centered on continual learning [17]. Continual learning techniques focus on easily incorporating new instruments into a model as additional data becomes available. Moreover, attempts have been made to capitalize on weakly annotated data, where instrument presence is identified but precise activation times are not specified, yielding only incremental enhancements [18].

The utilization of pre-training strategies has emerged as a pivotal area of interest. In 2023, Zong et al. [19] employed isolated notes for pre-training before transitioning to training on polyphonic data, albeit with a specific emphasis on predominant instrument recognition exclusively. Another explored avenue involves synthetic data generation achieved through layering mono-instrumental excerpts with tempo and pitch shifting to produce realistic artificial multi-instrument tracks [20].

Furthermore, model reprogramming has been proposed, involving training a smaller model to map inputs to the input space of a larger pre-trained model. This technique, akin to transfer learning, harnesses the generalization capabilities of the larger model to mitigate data imbalance, consequently significantly reducing training time requirements [21].

1.4. Contributions and outline of the paper

In this study, we introduce several methodologies aimed at efficiently incorporating hierarchical instrument structures into our predictive models, and evaluate this novel class of models tailored for hierarchical music prediction. Importantly, our evaluations are conducted on the MedleyDB dataset. This dataset is renowned for its expansive and varied content, which allows us to overcome constraints related to particular music genres and a restricted instrument set. As far as we know, this is the first work on polyphonic instrument recognition using the MedleyDB dataset, providing crucial baseline performances in this domain.

The paper is organized as follows. Section 2 offers an overview of the MedleyDB dataset used in our study while Section 3 delves into the neural network architecture selected for our research. In Section 4, we explore the various training strategies implemented to address the hierarchical structures of instruments. The numerical results derived from our evaluations are presented in Section 5. Finally, the conclusions and insights are summarized in Section 6.

2. HIERARCHICAL DATASET

This section focuses on the hierarchical dataset utilized in our study. The primary dataset of interest is MedleyDB, emphasizing its characteristics and composition. We then discuss the challenges of establishing a train/test split for MedleyDB to prevent overfitting and ensure a balanced instrument distribution within the sets. Finally, we introduce a labeling scheme incorporating instrument group hierarchies and utilize the Hornbostel-Sachs classification system to categorize instruments based on sound production methods, balancing granularity and computational efficiency.

2.1. MedleyDB

MedleyDB [22] is a dataset of 122 annotated polyphonic recordings containing a large diversity of genres and instruments. It was curated primarily to support research on melody extraction by providing melody f0 annotations, but each track also contains precise instrument activations, making it usable for instrument recognition. The dataset is filtered to only include the 94 tracks with no instrumental bleeding, in order to prevent erroneous instrument activation detections. As seen in Fig. 1, the distribution of instruments in MedleyDB is quite tail-heavy, featuring many instruments that appear in only a few of the tracks. This makes the resulting dataset extremely challenging. Indeed, it contains nearly as many instruments as tracks, which means the rarer instruments are usually showcased in a minimal context only.

We divide each track into non-overlapping frames of one-second duration. We consider an instrument active within a frame if it is active at any point during that duration. This method raises concerns regarding the potential misclassification of instruments if they are only briefly active at the beginning or end of a frame. However, our analysis demonstrates that such occurrences are rare, with less than 0.26% of frames containing an instrument active for less than 0.1 second.

2.2. Data train-test split

We split the MedleyDB dataset into train and test recordings to train and evaluate our MIR system. To our knowledge, no standard train/test split has been established for MedleyDB in prior work. Establishing such a split is challenging: using extracts from the same song in both training and testing has been shown to lead to overfitting, but some instruments are very rare and only appear in a single recording. Ensuring a similar label distribution then becomes quite challenging, especially given that we are working with only 94 tracks for 76 instruments. Ultimately, we select 20% of the recordings to ensure a similar instrument distribution between both sets. Due to the inclusion of a 17-minute long recording in the test set, we obtained a test set that is slightly larger than expected, with 14000 excerpts in the training set and 5000 in the test set. We were unable to pick an alternate split to reduce the test set size, as alternatives resulted in strong instrument distribution shifts between

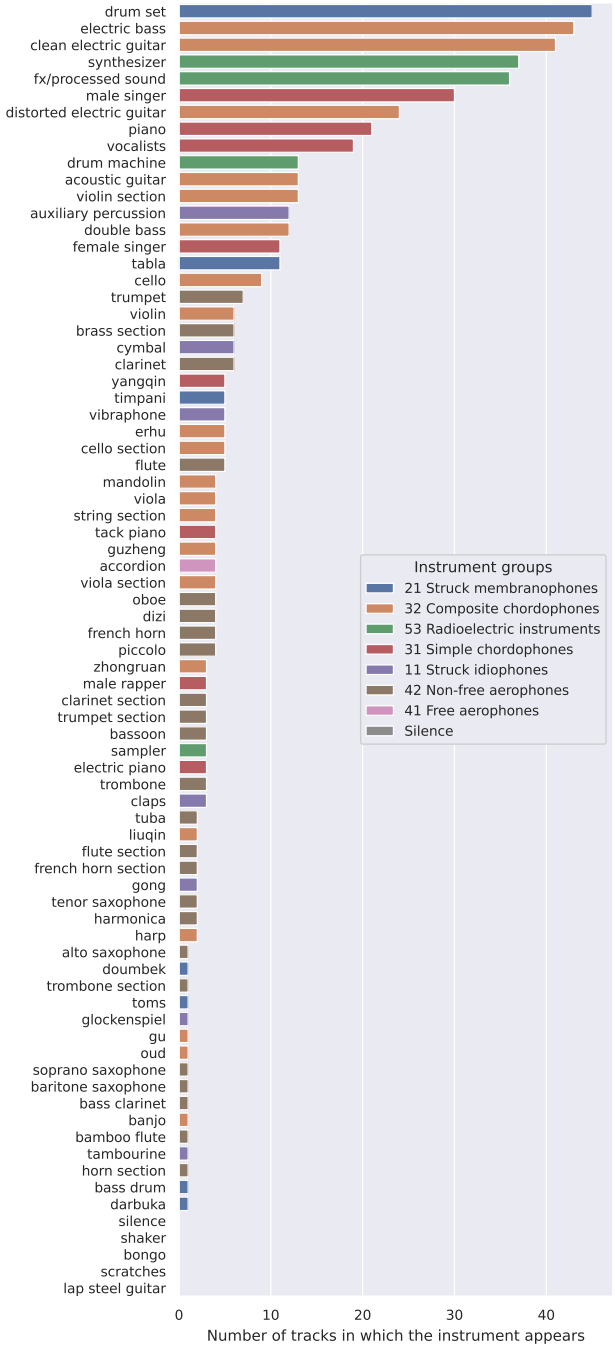

Figure 1: *Tail-heavy distribution of MedleyDB instrument occurrence*. The numbers in the legend refer to the Hornobel-Sachs taxonomy.

the training and the test data. This strong constraint on the dataset also made us unable to use k-fold validation. Special care is further taken to ensure the test set features various music genres. In the end, four instruments, each appearing in a single track, are present only at test time.

2.3. Hierarchical classification

We specify the labels further than a simple instrument name, adding labels per instrument group. We therefore end up with two primary sets of labels: $\mathcal{I}$, indicating the instrument's name, and $\mathcal{G}$, containing labels for instrument groups.

The hierarchical classification system selected is referred to as Hornbostel-Sachs [23], organizing instruments according to their sound production method. This classification system is versatile and can effectively categorize a wide array of instruments from diverse cultural backgrounds. Its adaptability is particularly advantageous for datasets like ours, which are characterized by diverse instruments. Moreover, the Hornbostel-Sachs features up to five levels of depth, enabling us to configure the level of precision of the tree easily. For this study, we opted for a depth of two, balancing granularity and computational efficiency. With this configuration, we split all instruments into 8 different groups, shown in Fig. 1.

However, this classification system has its drawbacks. Indeed, categorizing instruments based on their sound production method does not directly account for the output sound profile. This aspect poses challenges, especially when dealing with synthesized sounds. For instance, under this taxonomy, a drum machine would fall into a distinct class from a traditional drum despite producing similar sounds. We chose not to address these challenges specifically, as differentiating between synthetic and acoustic instruments may be required in certain contexts. The taxonomy can easily be adapted for tasks that do not require such a distinction.

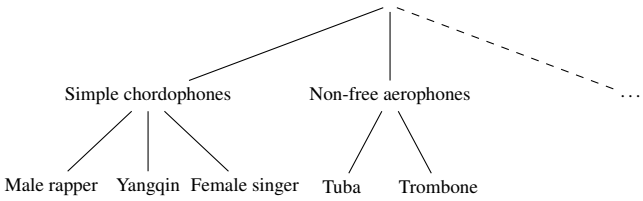

Figure 2: *Partial representation of the Hornbostel-Sachs class tree*

3. MODEL ARCHITECTURE

In this section, we give details on the model's architecture used as the base brick of our hierarchical classification system.

Note that the model is not the primary focus of the paper, and alternative architectures (e.g., based on ResNets [24]) could also be used here. We employ a convolutional network inspired by the VGG architecture [25], featuring a series of conv-conv-pool processing blocks. This architecture was chosen because VGGish models have shown good performance for MIR from spectral features for various downstream tasks [26], and require significantly less computing performance than transformer-based approaches. This series of processing blocks create feature maps of depth 64, then 128, and finally 256 while aggregating context along the temporal and pitch dimensions and are followed by a standard classification head. Batch normalization is further used after each layer for regularization, and a leaky ReLU is used for activation. We finally apply dropout before fully connected layers in the classification head. The exact architecture is specified in Table 1.

The network takes MFCC of an audio extract as input and outputs a vector of 85 values in $[0, 1]$ corresponding in activities of all classes in $\mathcal{I} \cup \mathcal{G}$. The MFCC input was chosen to consist of 1 s

of audio segments, computed using a hop-size of 1s on recordings sampled at 22.5 kHz, using 80 bins.

Layer	Output shape	Parameters
Input	(1, 80, 22)	
Conv2d	(64, 80, 22)	640
Batch normalization	(64, 80, 22)	128
Conv2d	(64, 80, 22)	36 928
Batch normalization	(64, 80, 22)	128
MaxPool2d	(64, 40, 11)	
Conv2d	(128, 40, 11)	73 856
Batch normalization	(128, 40, 11)	256
Conv2d	(128, 40, 11)	147 584
Batch normalization	(128, 40, 11)	256
MaxPool2d	(128, 20, 5)	
Conv2d	(256, 20, 5)	295 168
Batch normalization	(256, 20, 5)	512
Conv2d	(256, 20, 5)	590 080
Batch normalization	(256, 20, 5)	512
MaxPool2d	(256, 6, 1)	
Conv2d	(256, 1, 1)	393 472
Batch normalization	(256, 1, 1)	512
Squeeze	(256)	
Dropout	(256)	
Dense	(256)	65 792
Dropout	(256)	
Dense	(128)	32 896
Dropout	(128)	
Dense	(85)	10 965
Output: Sigmoid	(85)	

Table 1: *Model architecture used for our classification system. Leaky ReLUs are used as the activation function.*

4. MODEL TRAINING

We have tested four different approaches to model training in a hierarchical context, which we highlight here. We start by focusing on the impact of various loss functions, before introducing a new multi-model architecture[1].

4.1. Standard approach

In our initial approach, we treat the labels from the combined set $\mathcal{I} \cup \mathcal{G}$ as a unified entity and train the model on these grouped labels. This method has the advantage of being relatively straightforward but completely disregards the inherent hierarchical structure within the data. Consequently, it may yield inconsistent predictions, as nothing prevents the model from mistakenly predicting a group label along with an instrument that doesn't belong to that group.

As a first approach, we train a model using a standard cross-entropy loss. To counterbalance the pronounced class imbalance within the dataset, the loss is reweighted by inverse label frequency. This standard loss reweighting technique forms a good baseline but remains extremely limited. Therefore, we also test a loss built specifically for imbalanced datasets, the focal loss $\mathcal{L}_f$. This loss, initially defined for object detection [27], is defined as a slight variation on the cross-entropy loss:

$$\mathcal{L}_f(\hat{y}, y) = -(1 - p_t(\hat{y}, y))^2 \cdot \log(p_t(\hat{y}, y)) \quad (1)$$

with p_t the predicted probability of the correct class. This loss function has the advantage of dynamically giving more importance to misclassified samples during training. Indeed, for a sample classified correctly with high confidence, $1 - p_t$ nears 0, which causes the term to have little impact on the loss.

This approach makes for a good baseline, but is unable to treat the labels in $\mathcal{I}$ and $\mathcal{G}$ differently. In a second approach, we attempt to apply a weight to each tree level in the loss function. For a given loss function $\mathcal{L}$, we define:

$$\mathcal{L}_{weighted}(\hat{y}, y) = \mathbb{1}_{\mathcal{I}}(y) \cdot \alpha \mathcal{L}(\hat{y}, y) + \mathbb{1}_{\mathcal{G}}(y) \cdot (1-\alpha) \mathcal{L}(\hat{y}, y). \quad (2)$$

We then run a grid search for different α values, with $\mathcal{L}$ a cross-entropy loss. The results are presented in Fig. 3. The maximal F1-score across all nodes is obtained for $\alpha = 0.1$, that is, putting much more emphasis on the group-level loss term. The curve presented in Fig. 3, however, shows no clear trend.

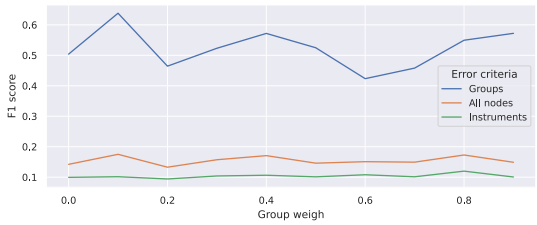

Figure 3: *F1-score depending on alpha value for $\mathcal{L}_{weighted}$.*

Every model is trained for 30 epochs using the Adam optimizer with a batch size of 32 and a learning rate of 0.001. The training was done on an Nvidia 1660Ti card, with each model taking around 30 minutes to train.

4.2. Specialized models

The previous approaches remained limited by treating labels from $\mathcal{I}$ and $\mathcal{G}$ as interchangeable, without any true accounting for the data's hierarchical structure. To overcome this problem and improve performance, we abandon the idea of a generalized model predicting groups and instruments in one pass and instead build a two-pass prediction system. To do so, we define a first model for group prediction, followed by specialized models for instrument prediction within each group. We train eight models using this approach: one group model trained with labels from $\mathcal{G}$ and seven specialized models, each predicting a subset of instruments from $\mathcal{I}$.

For simplicity, all models use the same VGG-like architecture, and all models are trained using the focal loss on the entirety of the dataset. This model has a much greater capacity than the baseline models. However, artificially increasing the capacity of the baseline models (by adding two extra conv-conv-pool blocks) shows no significant performance increase, which allows us to suggest that any changes in performance are due to the change in architecture, not in capacity. At inference time, the models are run in succession:

[1]The code to train our model is publicly available on github.

the group-level model is run first, followed by each instrument-level model. This has a significant impact on inference speeds, making them eight times slower. The effect could likely be mitigated by implementing a gating structure, and only calling the instrument-level models if the group-level prediction is above a given threshold.

5. RESULTS AND DISCUSSION

Considering the pronounced data imbalance and the absence of prioritization between false positives and negatives within the application, we use the F1-score metric [28] for evaluation:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{3}$$

As shown in Table 2, the balanced cross-entropy performs similarly to the focal loss, with the latter having a slightly better performance for instrument prediction. This is unsurprising, given that the focal loss allows reweighting at a label granularity rather than simply for the tree levels. The fact that both performances are similar suggests that the focal loss' primary role is probably in rebalancing loss terms between group and instrument-level labels.

On the other hand, the weighted cross-entropy approach shows inferior performance and fails to learn instrument labels. Overall, we notice that performance for groups is significantly higher than for instruments across all models. This result is in accordance with our initial expectations, given that the reason for implementing groups was hopes for better performance in groups even when fine-grain instrument detection is unachievable.

	Groups			**Instruments**		
	F1	Precision	Recall	F1	Precision	Recall
Balanced cross-entropy	0.74	**0.76**	0.72	0.41	**0.53**	0.35
Focal loss	0.74	**0.76**	0.73	0.43	0.52	0.37
Weighted cross-entropy	0.64	0.51	**0.86**	0.17	0.52	0.06
Group-specialized models	**0.78**	**0.76**	0.81	**0.45**	0.50	**0.40**

Table 2: *Performance of the different models. Averages are micro-averages, giving equal weight to each sample. The best method for each metric uses boldface.*

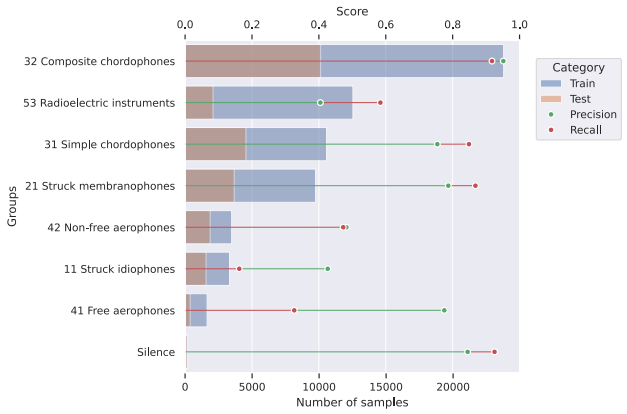

Figure 4: *Precision and recall per group, and as functions of the number of training and test samples for group-specialized models.*

This performance disparity is greatly lessened at a group level, as can be seen in Fig. 4. We can, however, notice the specific case of struck idiophones, which shows a much lower recall of 16%. Looking closer, we notice that this group is often misclassified as the *Struck membranophones* group. That is not very surprising, given the considerable overlap between some of the instruments within each group. For instance, a gong or cymbals will be classified as *Struck idiophones*, but any other auxiliary percussion will be considered a membranophone by default. A non-negligible amount of *Struck membranophones* instruments are also misclassified as *Radioelectric instruments*: this is likely due to the presence of the drum machine in the latter group. This shows the limitation of the chosen Hornbostel-Sachs class tree, which is very flexible in both depth and height but also can be prone to separating similarly-sounding instruments into very different groups.

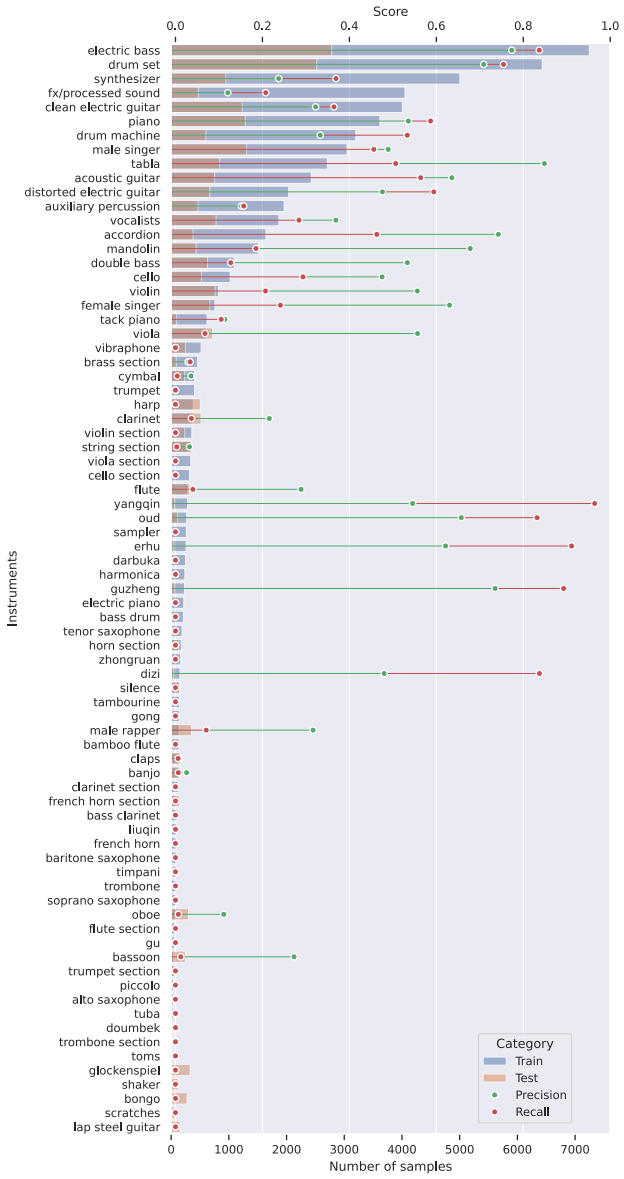

Figure 5: *Precision and recall per instrument, and as functions of the number of training and test samples for group-specialized models.*

We observed that our models are generally conservative, with recall scores notably lower than precision, especially at the instrument level. As depicted in Fig. 5, the model demonstrates reasonable performance for only about fifteen of the most common instruments, with performance sharply declining to almost zero precision and recall rates for most of the remaining dataset. Exceptions exist, with instruments such as the yangqin, the erhu, and the dizi showing some of the best performances. Given that these instruments all belong to traditional Chinese music, we can assume that the model has, to a degree, learned to recognize this distinctive genre and its associated instruments.

Furthermore, we are also able to confirm that the error of the model is caused by generalization issues. The performance of the model on the training set is excellent, as can be seen in Fig. 6 and Fig. 7. Initial experiments with a validation set also allowed us to check that the model did not overfit the training data.

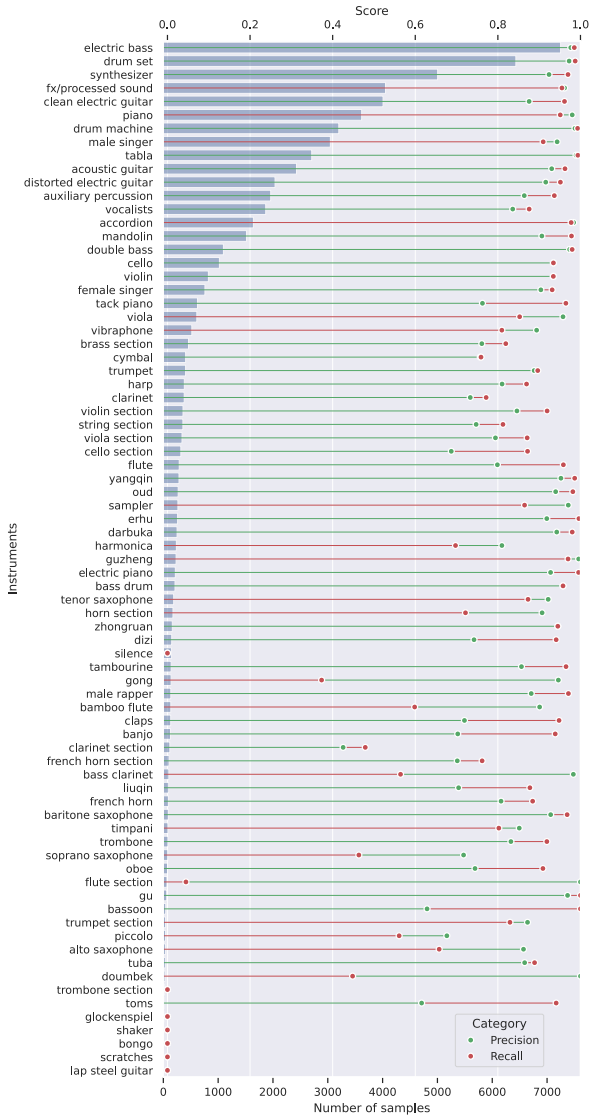

Figure 6: *Precision and recall for group-specialized models on the training data.*

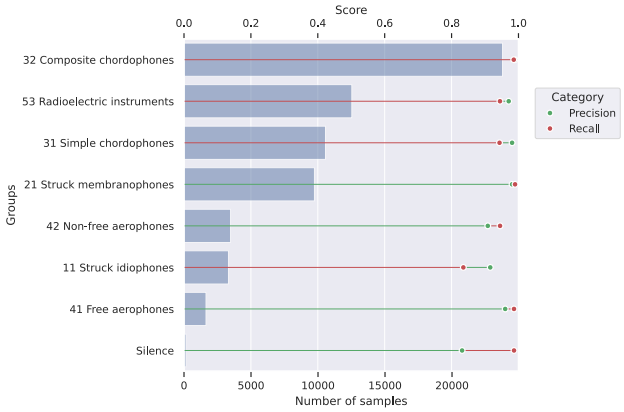

Figure 7: *Precision and recall on training data for group-specialized models.*

An important complicating element in instrument prediction lies in instrument co-occurrence. Let us define $C \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ where $C(i,j)$ represents the total instances of both the ith and jth instrument appearing together in a training set excerpt [2]. This co-occurrence matrix is subsequently normalized within the range of [0, 1] utilizing the methodology outlined in [29]:

$$C'(i,j) = \begin{cases} 0 & \text{if } i = j \\ \dfrac{C(i,j) - \min C(\cdot,j)}{\max C(\cdot,j) - \min C(\cdot,j)} & \text{otherwise.} \end{cases} \tag{4}$$

Because of the chosen normalization, the matrix is not symmetric and should be read "row-wise."

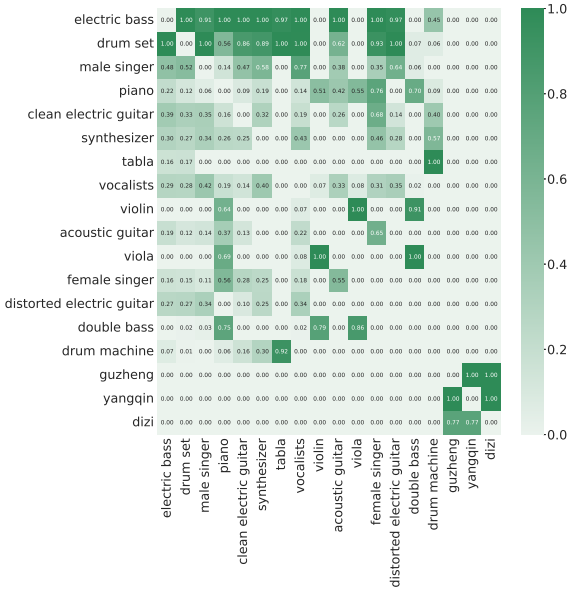

Figure 8: *Excerpt of the normalized co-occurrence of instrument labels in the training data.*

An excerpt from this matrix, in Fig. 8, shows these strong relations between some instruments. For instance, we can confirm the

[2] $|X|$ is the cardinal number of set X

speculated strong co-occurrence rate between Chinese instruments or notice that the violin and viola are always simultaneously present in the training data. Furthermore, displayed in Fig. 9 (resp. Fig. 10) is a co-occurrence matrix illustrating instances of ghost detection (resp. missed detection). Specifically, within Fig. 9, the entry at (i, j) denotes the occurrences of instrument j in an excerpt when the model incorrectly predicted a false positive for i. In Fig. 10, the element at (i, j) signifies the occurrences of predicted instrument j in an excerpt where i was erroneously identified as a false negative. These outcomes are standardized using the same methodology described in Eq. (4), and the results should be read "row-wise."

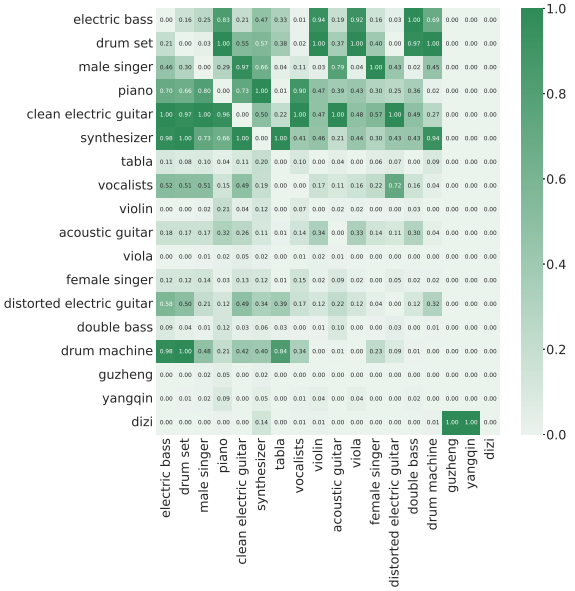

Figure 9: *False positive co-occurrence.*

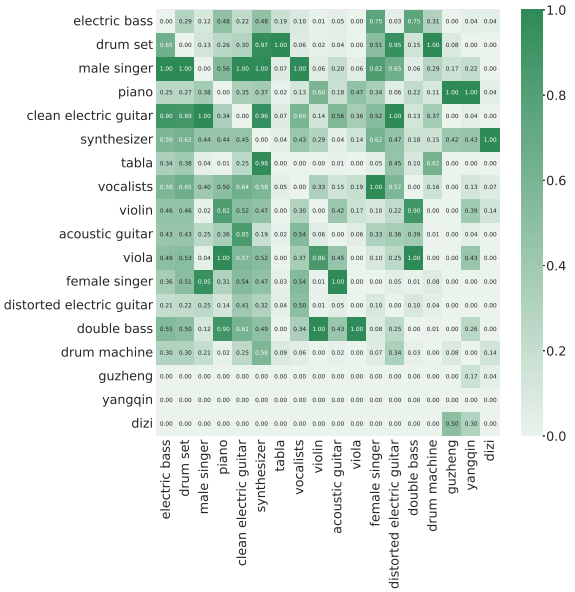

Figure 10: *False negative co-occurrence.*

Interestingly, instrument co-occurrence does not invariably result in false predictions, as the outcomes appear to be instrument-specific. Notably, the model's proficiency in recognizing instruments varies significantly. For instance, the model does not seem to have learned to effectively recognize the dizi, and seems to be detecting the guzheng and the yangqin as a proxy instead. Besides, the model encounters challenges in distinguishing between specific instrument categories. For example, it frequently confuses digital drum machines with drum sets and mixes the double bass with the electric bass. An interesting fact is the ghostly detection of a distorted electric guitar when singers, electric bassists, and drum set players are present. This result aligns with expectations due to the widespread use of these instruments in Western music.

6. CONCLUSION

This paper shows that the hierarchical approach proves highly beneficial in rare instrument recognition within complex datasets. While the F1-score at an instrumental level shows poor performance of 45%, the group-level score reaches up to 78%, allowing for much more reliable coarse-level instrument detection.

Looking ahead, there are a few areas that could be explored further. It would be interesting to investigate how to assess the system's adaptability to new instruments, particularly within established groups, to gauge its flexibility across various musical contexts. This would also allow us to bridge the gap between hierarchical systems and few-shot learning approaches. The current system's performance could also be evaluated on different datasets. Future works should also explore alternative input features for the neural network, such as audio scattering [30], and consider different hierarchical systems more tailored to machine learning methodologies. The chosen instrument hierarchy is likely to have a strong impact on results, and exploring automatic hierarchical classification for instruments [31] represents an intriguing avenue to improve detection. Alternative model architectures should also be explored, such as the promising Vision Transformer-based models. From the performance point of view, the specialized models can be simplified, making them smaller and faster to run. Such a study would make the models more efficient, which is crucial in real-world applications.

7. REFERENCES

[1] M. D. Ferreira, D. C. Corrêa, M. A. Grivet, G. T. dos Santos, R. F. de Mello, and L. G. Nonato, "On accuracy and time processing evaluation of cover song identification systems," *Journal of New Music Research*, vol. 45, no. 4, pp. 333–342, 2016.

[2] B. L Sturm, "Classification accuracy is not enough: On the evaluation of music genre recognition systems," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 371–406, 2013.

[3] Y. Song, S. Dixon, and M. Pearce, "A survey of music recommendation systems and future perspectives," in *9th international symposium on computer music modeling and retrieval*. Citeseer, 2012, vol. 4, pp. 395–410.

[4] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *International Society for Music Information Retrieval Conference*, 2021.

[5] L. C. Reghunath and R. Rajan, "Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, pp. 11, Dec. 2022.

[6] J. Marques and P. J Moreno, "A study of musical instrument classification using gaussian mixture models and support vector machines," *Cambridge Research Laboratory Technical Report Series CRL*, vol. 4, pp. 143, 1999.

[7] S. Essid, G. Richard, and B. David, "Musical instrument recognition on solo performances," in *2004 12th European signal processing conference*. IEEE, 2004, pp. 1289–1292.

[8] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science*, vol. 2, no. 6, pp. 420, Aug. 2021.

[9] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, vol. 14, Jan. 2019.

[10] K. Avramidis, A. Kratimenos, C. Garoufis, A. Zlatintsi, and P. Maragos, "Deep convolutional and recurrent networks for polyphonic instrument classification from monophonic raw audio waveforms," in *2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 3010–3014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems (NeurIPS)*, 2017, vol. 30.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2020.

[13] S. Jamil, M. S. Abbas, and A. M. Roy, "Distinguishing Malicious Drones Using Vision Transformer," *AI*, vol. 3, no. 2, pp. 260–273, Mar. 2022.

[14] A. L. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon, and J. P. Bello, "Chirping up the Right Tree: Incorporating Biological Taxonomies into Deep Bioacoustic Classifiers," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, May 2020, pp. 901–905.

[15] Z. Fu and L. Su, "Hierarchical classification networks for singing voice segmentation and transcription.," in *International Society for Music Information Retrieval Conference*, 2019, pp. 900–907.

[16] M. Krause and M. Müller, "Hierarchical Classification for Instrument Activity Detection in Orchestral Music Recordings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2567–2578, 2023.

[17] Y. Wang, N. J. Bryan, M. Cartwright, J. Pablo Bello, and J. Salamon, "Few-Shot Continual Learning for Audio Classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, June 2021, pp. 321–325.

[18] D. Mukhedkar, "Polyphonic Music Instrument Detection on Weakly Labelled Data using Sequence Learning Models," School Elect. Eng. Com- put. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2020.

[19] L. Zhong, E. Cooper, J. Yamagishi, and N. Minematsu, "Exploring isolated musical notes as pre-training data for predominant instrument recognition in polyphonic music," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 2312–2319.

[20] A. Kratimenos, K. Avramidis, C. Garoufis, A. Zlatintsi, and P. Maragos, "Augmentation Methods on Monophonic Audio for Instrument Classification in Polyphonic Music," in *2020 28th European Signal Processing Conference*, Jan. 2021, pp. 156–160.

[21] H.-H. Chen and A. Lerch, "Music instrument classification reprogrammed," in *International Conference on Multimedia Modeling*. Springer, 2023, pp. 345–357.

[22] R. M Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J.P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research.," in *International Society for Music Information Retrieval Conference*, 2014, vol. 14, pp. 155–160.

[23] A. M. von Hornbostel and C. Sachs, "Systematik der Musikinstrumente. Ein Versuch," *Zeitschrift für Ethnologie*, vol. 46, no. 4/5, pp. 553–590, 1914, Publisher: Dietrich Reimer Verlag GmbH.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[25] K Simonyan and A Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[26] A. Grollmisch, E. Cano, C. Kehling, and M. Taenzer, "Analyzing the Potential of Pre-Trained Embeddings for Audio Classification Tasks," in *2020 28th European Signal Processing Conference*, Jan. 2021, pp. 790–794, ISSN: 2076-1465.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 2980–2988.

[28] CJ van Rijsbergen, *Information retrieval*, Butterworth-Heinemann, 1979.

[29] R. Huang, F. Zheng, and W. Huang, "Multilabel remote sensing image annotation with multiscale attention and label correlation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 6951–6961, 2021.

[30] J. Andén and S. Mallat, "Multiscale scattering for audio classification.," in *International Society for Music Information Retrieval Conference*. Miami, Florida, 2011, pp. 657–662.

[31] G. Peeters, "Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization," in *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.