ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095162

AN ATTENTION-BASED APPROACH TO HIERARCHICAL MULTI-LABEL MUSIC INSTRUMENT CLASSIFICATION

Zhi Zhong, Masato Hirano, Kazuki Shimada, Kazuya Tateishi, Shusuke Takahashi, Yuki Mitsufuji

Sony Group Corporation, Tokyo, Japan

ABSTRACT

Although music is typically multi-label, many works have studied hierarchical music tagging with simplified settings such as single-label data. Moreover, there lacks a framework to describe various joint training methods under the multi-label setting. In order to discuss the above topics, we introduce hierarchical multi-label music instrument classification task. The task provides a realistic setting where multi-instrument real music data is assumed. Various hierarchical methods that jointly train a DNN are summarized and explored in the context of the fusion of deep learning and conventional techniques. For the effective joint training in the multi-label setting, we propose two methods to model the connection between fine- and coarse-level tags, where one uses rule-based grouped max-pooling, the other one uses the attention mechanism obtained in a data-driven manner. Our evaluation reveals that the proposed methods have advantages over the method without joint training. In addition, the decision procedure within the proposed methods can be interpreted by visualizing attention maps or referring to fixed rules.

***Index Terms*—** Music Tagging, Hierarchical Classification, Multi-label Classification, Instrument, Attention

1. INTRODUCTION

Multi-label music tagging is a classification task in which the goal is to predict multiple semantic tags for a given music piece. Tags can indicate the genres, moods and instruments of the music. Therefore, this task is meaningful for applications such as music recommendation or music retrieval. Music tags organized in a tree-like structure, *i.e.*, a hierarchy as shown in Fig. 1, present the domain knowledge (what kind of tags are musically correlated), bringing benefits including improved tagging performance [1, 2, 3, 4]. While music tagging datasets typically have a flat hierarchy [5, 6, 7, 8], there has been growing interests in hierarchical tagging in the field of music information retrieval [9].

Several works have tackled hierarchical music tagging. Parmezan *et al.* have investigated hierarchical genre classification using conventional machine learning methods without deep learning [10]. A few works have studied deep neural network (DNN)-based hierarchical methods under a simplified problem setting. For example, Garcia *et al.* tackled few-shot instrument classification for single-instrument data [11], while Nolasco *et al.* have studied instrument representation learning with single-note data [12]. Toward hierarchical multi-label music tagging task, Krause *et al.* reported on several DNN-based hierarchical methods on singing activity detection [13]. Many of their discussions are devoted to training separate DNNs, rather than training these DNNs jointly.

Although real music is typically polyphonic and multi-instrument, many works have addressed hierarchical music tagging with simplified data. Therefore, DNN-based hierarchical music tagging under a realistic problem setting is yet to be discussed extensively. Moreover, there lacks a framework to describe various joint training methods under the multi-label setting. While using multiple separate models is effective, DNNs have been shown capable of learning hierarchical information during the joint training [1, 2, 3, 4, 11, 12].

In this paper, we address hierarchical multi-label music tagging with joint training methods of a single DNN. To study hierarchical multi-label music tagging, we introduce the multi-label music instrument classification task, which involves instruments organized in a 2-level hierarchy. The contributions of this paper are as follows. First, the task provides a more **realistic scenario**, where we address real music that is typically polyphonic, multi-instrument and diverse in genre. Second, we categorize and explore various **joint training methods** for DNN under a framework similar to the categorization of conventional hierarchical methods, to facilitate further exploration of the fusion of deep learning and conventional techniques. Finally, for the effective joint training in the multi-label setting, we propose **ResAtt** and grouped max pooling (**GMP**) for applying a residual attention layer or max pooling operations to model the connection between fine- and coarse-level tags.

2. RELATED WORK

In the field of audio, existing studies on hierarchical classification have primarily focused on sound event detection tasks. In [1], the connection between coarse- and fine-level tags is formulated as a grouped summation pooling. The formulation requires fine-level predictions that are normalized by a softmax activation, which focuses on single-label tasks. Zharmagambeto *et al.* combine a DNN with a decision tree [4], which encourages the DNN to learn a representation that contains hierarchical information. However, a separate classifier is used in the inference phase, so the classifier is not optimized by hierarchical information.

Nolasco *et al.* studied hierarchical metric learning of music instruments [12] in the Nsynth dataset [14] (single note by single instrument). A polyphonic version of the instrument task was addressed in [11] under the few-shot learning setting, with single-instrument data taken from the stem tracks of MedleyDB dataset [15]. After filtering out tags that are not sufficiently fine, *e.g.*, "Drum", hierarchies are induced on the basis of instrument categorization used in the music world [16]. This idea inspired us to introduce our own tone-base hierarchy (shown in Fig. 1) into the dataset that is used in our experiments.

Hierarchical multi-label music tagging is discussed in [13] in terms of singing activity detection. Loss items are introduced to build the connection between coarse and fine predictions made by a single DNN to improve the detection performance. A framework to describe various DNN-based hierarchical methods is also introduced in [13], however, many of the studies have been devoted to separate multi-model approaches, rather than training these models jointly.

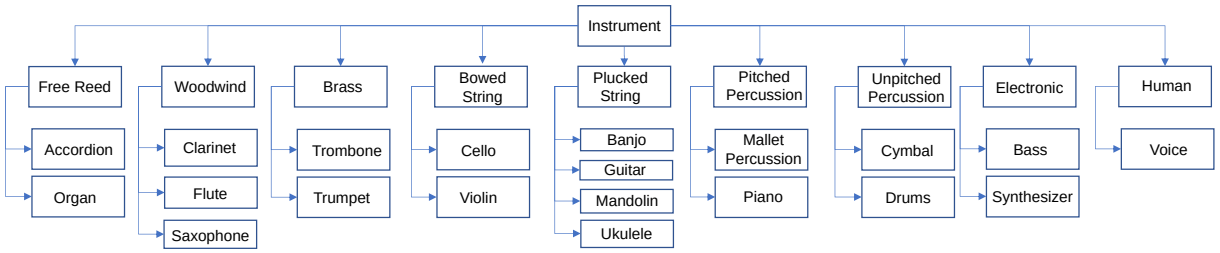
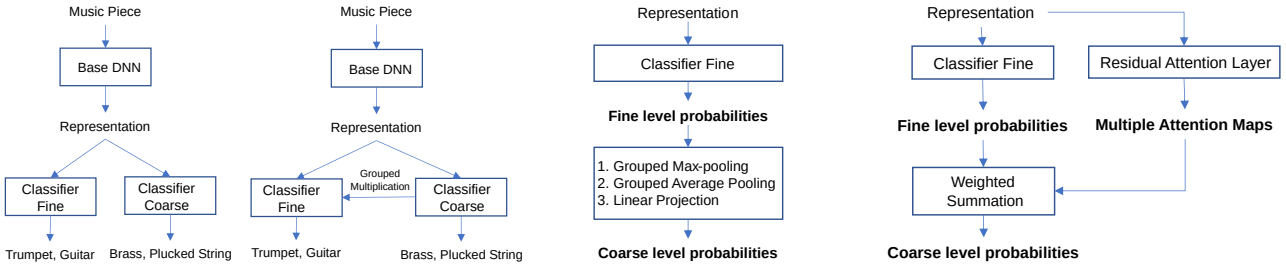


Fig. 1. Induced tone-base 2-level instrument hierarchy in OpenMIC dataset



(a) Level-wise approach (b) Top-down approach (c) Bottom-up approach (d) ResAtt: Proposed bottom-up approach

Fig. 2. Joint training approaches for DNN

3. PROPOSED METHODS

Suppose the N_{dim}-dimension representation of a music piece is extracted by a base DNN as $\mathbf{Z}^{N_{\text{dim}}}$. The representation is then projected into fine-level probabilities, $\mathbf{P}_{\text{fine}} \in \mathbb{R}^{N_{\text{fine}} \times 1}$, by a classifier through sigmoid activations. Binary classification is performed by converting $\mathbf{P}_{\text{fine}}$ into binaries using tag-wise thresholds. A 2-level (fine/coarse) hierarchy is assumed. Coarse-level probabilities are denoted as $\mathbf{P}_{\text{coarse}} \in \mathbb{R}^{N_{\text{coarse}} \times 1}$.

3.1. Hierarchical Approaches for Joint Training

Our main focus in this paper is jointly training a DNN for hierarchical multi-label classification. We begin by summarizing various related methods similarly to the categorization of conventional hierarchical techniques described in [13, 17], and demonstrate how deep learning is combined with conventional techniques.

Level-wise approach. Conventionally, one model is prepared for each level in the hierarchy [17, 13]. The idea has been adapted to the joint training framework in [13], which could be illustrated as the structure shown in Fig. 2 (a). Although this approach helps the base DNN to learn hierarchical information, classifiers remain independent during training. In [13], two loss items are introduced to optimize these classifiers with hierarchical information.

Top-down approach. In convention, a model is first trained to classify coarse tags, then separate models are prepared at each coarse tag to classify its child (fine-level) tags [17, 10, 13]. We summarize it as "coarse first, fine last". In order to adapt this approach to jointly train a DNN, we connect a simplified soft decision tree (SDT) [18] similar to [4] after the base DNN. In an SDT, the fine-level probability is the multiplication of the leaf node and its parent node (coarse-level probability) [18], which is called the grouped multiplication between a coarse tag and its child fine-level tags in Fig. 2 (b). Since a multi-label task is assumed, softmax activations in the SDT are replaced by sigmoid. Unlike [4], we use the SDT for inference, as the SDT is jointly optimized during training. However, in the top-down approach, an error made in the coarse level is difficult to correct, which may affect its fine-level performance [13].

Bottom-up approach. In the conventional bottom-up approach, only a non-hierarchical (flat) model is trained with fine-level tags. During inference, coarse tags are predicted by aggregating fine-level predictions with pre-defined rules [13, 17]. We summarize it as "fine first, coarse last". The core problem of the conventional bottom-up approach is that, models are not jointly optimized with "bottom-up rules", resulting in non-optimal performance. Another major drawback is similar to the top-down approach, where fine-level errors propagate to the coarse level [13, 17].

A common rule for the bottom-up aggregation is that, when a fine-level tag is assigned to a music sample, the parent coarse-level tag will be assigned to the music as well, *e.g.*, a music sample will be annotated with "Woodwind" if it is annotated with "Flute". We call this rule as the grouped max-pooling (GMP), because the rule is equal to applying max-pooling to different groups of child fine-level tags. Obviously, the fine- and coarse-level tags in a hierarchical dataset also satisfy the same rule [11, 13, 19].

We emphasize the concept of the **bottom-up approach with joint training**, where DNNs are optimized jointly with aggregation rules. Based on the fact that the fine- and coarse-level labels in a dataset satisfy rules similar to max-pooling, we propose to apply GMP **during training** (Fig. 2 (c)) to inform the DNN of the hierarchical structure in the dataset and improve tagging performance.

3.2. ResAtt: Attention-based Bottom-up Method

We also propose ***ResAtt***, which models the connection between fine- and coarse-level tags by the attention mechanism. ResAtt can use the attention map with elements valued between 0 - 100% to tell the system which fine-level tags are and are not important for a specific coarse-level tag. This idea is inspired by the insight that max-pooling is equal to an operation which produces a 0/1 binary attention map to select out the most important element from an input vector. Hence, both proposed methods can be understood as attention-based approaches.

As shown in Fig. 2 (d), a residual attention layer is used to generate the attention map for each coarse-level tag, denoted as $\mathbf{W} \in \mathbb{R}^{N_{\text{fine}} \times N_{\text{coarse}}}$; the attention map is applied to fine-level predictions, to finally predict coarse-level tags through the following formula:

$$\mathbf{P}_{\text{coarse}} = \mathbf{W}^T \cdot \mathbf{P}_{\text{fine}} \in \mathbb{R}^{N_{\text{coarse}} \times 1}, \quad (1)$$

where T is the transpose of matrix. A softmax activation is applied to the N_{fine} dimension of attention map $\mathbf{W}$ to ensure that the resulting coarse-level probabilities remain meaningful (below 100%).

In ResAtt, to obtain the prediction of coarse-level tags, high quality results from the fine-level classifier are required first. This helps to optimize the classifiers and base DNN jointly during training. Unlike the top-down or the joint training method with GMP, which is informed of the hierarchical structure of the dataset via the tree structure or the max-pooling rule, ResAtt has no access to such prior knowledge, but explores proper aggregation rules in a data-driven manner. In Sec. 5, we will discuss how this feature helps to prevent fine-level errors from propagating to the coarse level.

Following [1, 11], the overall Binary Cross Entropy (BCE) loss is formulated as the weighted summation of level-wise losses, *i.e.*,

$$L_{\text{BCE}} = \lambda L_{\text{BCE}}^{\text{fine}} + (1-\lambda) L_{\text{BCE}}^{\text{coarse}}, \tag{2}$$

where λ is the weight for fine level tags.

4. EVALUATION

4.1. Tone-base Hierarchical Dataset

OpenMIC [20] is a music instrument classification dataset that offers a task close to real applications; 10-second music clips of various genres are taken from the FMA dataset [19] and annotated concerning 20 instruments in a multi-label manner. While official annotations only include the fine level, we introduce a 2-level instrument hierarchy (Fig. 1) based on the tonal properties of instruments. This pre-processing is similar to the methodology described in [11]. However, our hierarchy is different from [11] in that, we include annotations that are considered not "fine" enough in [11] into our hierarchy, such as "Mallet Percussion" or "Drums", so we can use all music clips provided by OpenMIC.

Since there is no official validation set for hyperparameter tuning, 15% of the training set data is taken out for validation, following the practice in [21]. Stratified sampling by scikit-multilearn library [22] is used.

Many instruments remain not annotated in the OpenMIC dataset, so the dataset is released with a masking file, telling users what specific instruments are examined in a track. We follow the common practice of using this masking file to calculate the loss during training and the metrics during evaluation [21, 23]. The pseudocode for loss calculation can be written as $\text{Loss} = L_{\text{BCE}}(\text{predictions}[\text{mask}], \text{label}[\text{mask}])$.

4.2. Experiments

We use the CNN14 architecture pretrained on AudioSet [24] as the base DNN of all methods shown in Fig.2, where $N_{\text{dim}} = 2048$, and the fine level classifier is a single linear layer. Raw music data are converted to mono-channel at 16kHz sampling rate, which are further transformed into 64-bin mel-spectrograms (frequency range: [50Hz, 8kHz]), via short-time Fourier transform with 32-ms Hann window and 10-ms hop size. The input audio length is the same as the clip length in OpenMIC.

We evaluate various approaches to compare them with our proposed methods, of which the model size has been kept in almost the same level. We examine the following methods with eight different random seeds and compute their performance metrics. **Flat baselines**. We fine tune the CNN14 model as the flat baseline for the fine level. Coarse-level predictions of the model is produced by applying GMP at the inference phase (conventional bottom-up approach used in [4, 13]). Fine-level results reported in [23] are compared with our baseline. **Level-wise approach**. We evaluate the DNN shown in Fig. 2 (a) with two parallel linear layers. This architecture is trained by incorporating Eq. 2 with loss items proposed in [13]. **Top-down approach**. We evaluate the DNN with an SDT classifier shown in Fig. 2 (b), where each level in the tree is a single linear layer. **Bottom-up approach with joint training**. We evaluate ResAtt in Fig. 2 (d), whose attention layer is implemented by reshaping the output of a linear layer, which is as light weighted as the level-wise or the top-down approach. We evaluate the proposed joint training method with GMP as well. To compare with ResAtt, we replace the attention mechanism with a linear projection (LP) layer. Similarly, we replace max-pooling with average pooling to form the grouped average pooling (GAP) method to compare with the joint training method with GMP. We jointly train GAP and LP as in Fig. 2 (c).

During the training, the batch size is 16. The maximum learning rate is 1e-4. The first 5 epochs are trained with linearly increasing learning rates (linear warm-up) to make the training more stable. The Adam optimizer [25] with weight decay of 1e-4 is used. We utilize SpecAugment [26] for data augmentation. Grid searches are carried out for the weight λ in the loss function (Eq. 2) across all methods. We empirically set the searching range as $\lambda \in 0.70, 0.75, 0.80, 0.85, 0.90$. The setting with the lowest validation loss for fine-level tags is used for evaluation.

4.3. Metrics

For objective evaluation, we use the macro average of ROC-AUC, PR-AUC (also known as mean average precision), and F1 score. All metrics are calculated using the scikit-learn library [27]. F1 scores are calculated on the basis of 0/1 binary predictions, which are produced by thresholds optimized on the validation set.

5. RESULTS

The average of evaluation results across eight random seeds are presented in Tab. 1, where digits denote various design choices.

Our model for fine-level classification outperformed PaSST-S in [23]. This implies that our choice of base DNN as well as our training settings are feasible.

The level-wise approach combined with loss items in [13] improved coarse-level performance by a large margin, and improved the fine-level performance especially for the F1 Score. However, it is difficult to interpret the decision procedures within this model. Meanwhile, the top-down approach is interpretable because of its tree structure in the classifier [18]. In the coarse level, the top-down approach has a performance similar to the level-wise approach, but in the fine level, the approach resulted in deteriorated PR-AUC value. As described in Sec. 3.1, its fine-level performance may have been limited by its coarse-level performance.

The only difference between the proposed joint training method with GMP and our baseline is that, the proposed method optimizes the DNN with the GMP rule by joint training. The joint training method with GMP is interpretable through the GMP rule, *e.g.*, "Woodwind" is predicted because at least one of "Flute", "Clarinet" or "Saxophone" is predicted with high probability. As mentioned in Sec. 3.1, fine- and coarse-level annotations in the tone-base hierarchical dataset satisfy rules similar to GMP, which explains the large performance improvements brought by GMP joint training. Replacing the max-pooling with average pooling operations results in the GAP method. Since in the dataset a coarse-level label is not the average of its child fine-level labels, jointly training the DNN with GAP resulted in lower performance.

Table 1. Results at the tone-base hierarchy (%)

		Fine Level			Coarse Level		
Comments	Method	ROC-AUC	PR-AUC	F1 Score	ROC-AUC	PR-AUC	F1 Score
	PaSST-S [23]	-	84.3	-	n/a	n/a	n/a
baseline	Flat fine-level classification & inference phase bottom-up	91.7	85.6	83.3	92.4	89.6	84.4
Fig. 2 (a)	Level-wise approach with loss items in [13] (our tuning)	91.9	85.8	**83.7**	93.0	90.4	85.2
Fig. 2 (b)	Top-down approach	91.9	85.4	83.6	93.0	90.4	85.3
	Bottom-up approach with joint training						
Fig. 2 (c)	1. Grouped average pooling (GAP)	91.7	85.6	83.2	91.7	87.6	84.6
Fig. 2 (c)	2. Linear projection (LP)	91.8	85.8	83.6	75.3	70.9	67.6
Fig. 2 (c)	3. Grouped max-pooling (GMP) (proposed)	91.9	85.9	83.6	**93.1**	**90.7**	85.4
Fig. 2 (d)	4. ResAtt (proposed)	**92.0**	**86.0**	**83.7**	**93.1**	**90.7**	**85.5**

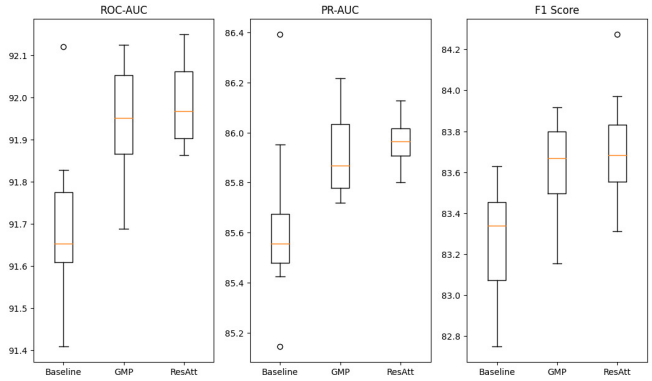


Fig. 3. The boxplot of fine-level metrics for the baseline and proposed models. Orange lines denote the medians

Contrary to the joint training method with GMP, ResAtt has little access to prior knowledge of the hierarchical structure in the dataset, but attempts to extract the aggregation rule from labels alone. ResAtt achieves scores that are equal to or slightly higher than the joint training method with GMP. The LP method utilizes a learnable projection layer, instead of the attention layer, but failed to predict coarse-level tags. This is because that a linear layer cannot flexibly adjust its projection rule for various input, which reveals how important it is to use attention mechanism in the bottom-up approach with joint training.

Since the improvements from the baseline in fine-level metrics are small in terms of absolute value, we compare the proposed methods with the baseline in box plots (Fig. 3). Higher medians and boxes demonstrate their advantages over the baseline. Although ResAtt has slightly higher medians, the difference between the joint training method with GMP and ResAtt is not significant.

Attention map samples with tag-wise probabilities made by ResAtt are shown in Fig. 4. All maps are similar to the GMP operation, but vary depending on input music, which implies that ResAtt has extracted the aggregation rule without prior knowledge. On the lower side, although the "piano" in the fine level is wrongly predicted as 83%, ResAtt did not pass this error to the coarse level by giving "piano" less attention. This shows that ResAtt can learn flexible aggregation that is difficult to describe with fixed rules, and in some cases, it can even prevent the error from propagating to the coarse level. Such behaviors are impossible for the joint training method with GMP, as the method accumulates errors made in the fine level [13, 17], which is a possible reason for the slightly higher F1 Score of ResAtt in the coarse level. The decision procedure within ResAtt is interpretable on the basis of attention map visualization.

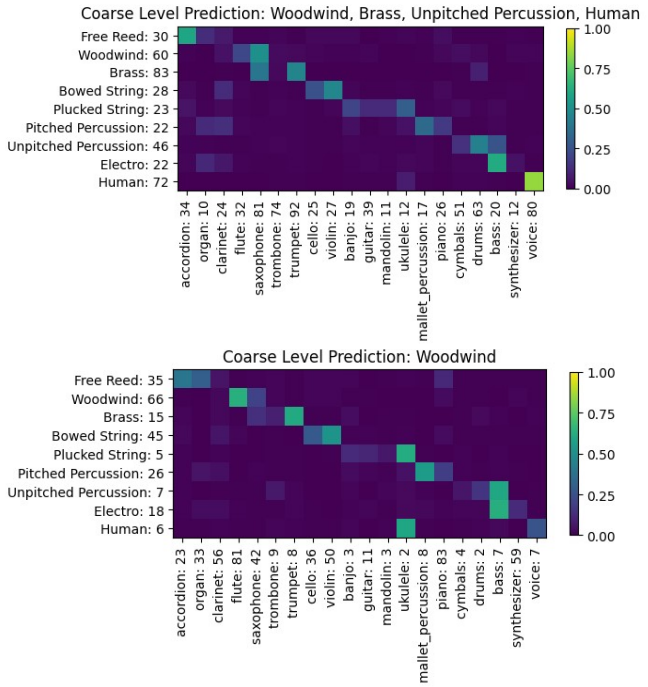


Fig. 4. Attention maps. Coarse tag predictions are listed at the top; tag-wise probabilities (%) are listed with tag names.

6. CONCLUSION AND FUTURE WORK

We investigated hierarchical multi-label music instrument classification as a case study of hierarchical music tagging. We extended hierarchical instrument classification to the multi-label setting and realistic music data, with an induced tone-base hierarchy. Various hierarchical methods that jointly train a DNN are summarized in the context of the fusion of deep learning and conventional techniques. For the effective joint training in the multi-label setting, we propose two methods to model the connection between fine- and coarse-level tags, where one uses the attention mechanism obtained in a data-driven manner, the other uses rule-based grouped max-pooling, which is explained as a binary attention. Evaluation results indicate that proposed methods, especially the ResAtt, are promising as the bottom-up methods with joint training. In addition to its performance, ResAtt can learn flexible aggregation that is difficult to describe with fixed rules and even prevent errors from propagating. By visualizing attention maps, the interpretability of ResAtt can be enhanced. Future work involves extending current methods to other tasks in music tagging.

7. REFERENCES

[1] Abelino Jimenez, Benjamin Elizalde, and Bhiksha Raj, "Sound event classification using ontology-based neural networks," in *Proc. of NeurIPS 2018*, 2018, vol. 9.

[2] Arindam Jati, Naveen Kumar, Ruxin Chen, and Panayiotis Georgiou, "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *Proc. of ICASSP 2019*. IEEE, 2019, pp. 6–10.

[3] Yiwei Sun and Shabnam Ghaffarzadegan, "An ontology-aware framework for audio event classification," in *Proc. of ICASSP 2020*. IEEE, 2020, pp. 321–325.

[4] Arman Zharmagambetov, Qingming Tang, Chieh-Chi Kao, Qin Zhang, Ming Sun, Viktor Rozgic, Jasha Droppo, and Chao Wang, "Improved representation learning for acoustic event classification using tree-structured ontology," in *Proc. of ICASSP 2022*. IEEE, 2022, pp. 321–325.

[5] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie, "Evaluation of algorithms using games: The case of music tagging.," in *Proc. of ISMIR 2009*, 2009, pp. 387–392.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere, "The million song dataset," in *Proc. of ISMIR 2011*, 2011.

[7] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The mtg-jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, ICML 2019*, Long Beach, CA, United States, 2019.

[8] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra, "Evaluation of cnn-based automatic music tagging models," in *Proc. of 17th Sound and Music Computing*, 2020.

[9] Minz Won, Janne Spijkervet, and Keunwoo Choi, *Music Classification: Beyond Supervised Learning, Towards Real-world Applications*, https://music-classification.github.io/tutorial, November 2021.

[10] Antonio Rafael Sabino Parmezan, Diego Furtado Silva, and Gustavo EAPA Batista, "A combination of local approaches for hierarchical music genre classification.," in *Proc. of ISMIR 2020*, 2020, pp. 740–747.

[11] Hugo Flores Garcia, Aldo Aguilar, Ethan Manilow, and Bryan Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proc of ISMIR 2021*, 2021.

[12] Inês Nolasco and Dan Stowell, "Rank-based loss for learning hierarchical representations," in *Proc. of ICASSP 2022*. IEEE, 2022, pp. 3623–3627.

[13] Michael Krause and Meinard Müller, "Hierarchical classification of singing activity, gender, and type in complex music recordings," in *Proc. of ICASSP 2022*. IEEE, 2022, pp. 406–410.

[14] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," 2017.

[15] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research," in *Proc. of ISMIR 2014*, 2014, vol. 14, pp. 155–160.

[16] Erich M. von Hornbostel and Curt Sachs, "Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann," *The Galpin Society Journal*, vol. 14, pp. 3–29, 1961.

[17] Carlos N Silla and Alex A Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011.

[18] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E. Gonzalez, "Nbdt: Neural-backed decision trees," in *ICLR 2021*, 2021.

[19] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "FMA: A dataset for music analysis," in *Proc. of ISMIR 2017*, 2017.

[20] Eric Humphrey, Simon Durand, and Brian McFee, "Openmic-2018: An open data-set for multiple instrument recognition.," in *Proc. of ISMIR 2018*, 2018, pp. 438–444.

[21] Siddharth Gururani, Mohit Sharma, and Alexander Lerch, "An attention mechanism for musical instrument recognition," in *Proc. of ISMIR 2019*, 2019.

[22] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," *ArXiv e-prints*, Feb. 2017.

[23] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer, "Efficient training of audio transformers with patchout," in *Proc. of Interspeech 2022*, 2022.

[24] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM TASLP*, vol. 28, pp. 2880–2894, 2020.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. of Interspeech 2019*, 2019.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.