Audio Engineering Society

Convention Paper 10201

Presented at the AES 157th Convention 2024
October 8–10, New York, NY, USA

This paper was peer-reviewed as a complete manuscript for presentation at this convention. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Music auto-tagging in the long tail: A few-shot approach

T. Aleksandra Ma and Alexander Lerch

Music Informatics Group, Georgia Institute of Technology

Correspondence should be addressed to T. Aleksandra Ma and Alexander Lerch (tengofma@gmail.com, alexander.lerch@gatech.edu)

ABSTRACT

In the realm of digital music, using tags to efficiently organize and retrieve music from extensive databases is crucial for music catalog owners. Human tagging by experts is labor-intensive but mostly accurate, whereas automatic tagging through supervised learning has approached satisfying accuracy but is restricted to a predefined set of training tags. Few-shot learning offers a viable solution to expand beyond this small set of predefined tags by enabling models to learn from only a few human-provided examples to understand tag meanings and subsequently apply these tags autonomously. We propose to integrate few-shot learning methodology into multi-label music auto-tagging by using features from pre-trained models as inputs to a lightweight linear classifier, also known as a linear probe. We investigate different popular pre-trained features, as well as different few-shot parametrizations with varying numbers of classes and samples per class. Our experiments demonstrate that a simple model with pre-trained features can achieve performance close to state-of-the-art models while using significantly less training data, such as 20 samples per tag. Additionally, our linear probe performs competitively with leading models when trained on the entire training dataset. The results show that this transfer learning-based few-shot approach could effectively address the issue of automatically assigning long-tail tags with only limited labeled data.

1 Introduction

In the rapidly evolving world of digital music, the ability to efficiently organize and retrieve music from expansive databases is crucial for streaming platforms, content creators, and consumers alike. However, mapping tags to musical characteristics is challenging due to the subjectivity inherent in both language and music perception, which varies across individuals and cultures. Usually, music catalog owners either maintain an in-house annotation team to tag songs in their music repository or use an out-of-the-box music auto-tagging solution. Human tagging, while more flexible and accurate when compared with auto-tagging solutions, is labor-intensive and expensive. Music auto-tagging solutions, on the other hand, are fast and scalable but heavily reliant on a predefined set of (training) tags, as these solutions are commonly trained with supervised learning methods [1, 2]. These supervised methods require a large amount of training data and ground truth label pairs. The reliance on a fixed set of tags limits their adaptability and personalization to the diverse and dynamic vocabulary of users across different cultures and demographics. For instance, a music branding company curating playlists for a high-end hotel may require a detailed taxonomy of different types of jazz, but most auto-taggers' taxonomies stop at a general classification of jazz. Similarly, wedding music varies significantly across cultures, highlighting the need for more nuanced and customizable tagging systems.

Few-shot learning approaches [3] present a framework for models to learn new tags from just a few examples, thereby mitigating the need for extensive retraining. This method could enable music catalog owners, such as production music libraries, to save considerable resources on annotation when they need to expand the set of tags, modify their current tag taxonomy, migrate to a new taxonomy, or consolidate two or more taxonomies. An adaptable system thus ensures compatibility across different catalogs and could also enable end users to apply their own, use-case specific taxonomies to their music libraries.

In this study, we propose the utilization of pre-trained audio embeddings as inputs to a simple multi-label, few-shot music auto-tagging model. An ablation study is conducted to examine the impact of the number of classes and the quantity of training samples per class on model performance to estimate the viability of this approach in different scenarios.

In summary, the main contributions of this study are

- the introduction of few-shot multi-label classification to the music auto-tagging task,
- a comparison of various pre-trained embeddings for music auto-tagging,
- a systematic investigation of the effects of number of tag classes and training samples on few-shot classifier performance.

The remainder of this paper is structured as follows: Sect. 2 discusses related work in music auto-tagging and few-shot learning. Sect. 3 describes our method, the dataset, the pre-trained features used in our experiments, as well as the methods and experimental setup. Sect. 4 details the results and analysis. Finally, Sect. 5 concludes the paper with a summary of our findings.

2 Related Work

Music auto-tagging is inherently a multi-label problem [4], meaning that each music track can be associated with multiple tags simultaneously, rather than being limited to a single category. Tags span various semantic categories, including genres, instruments, and mood. These tags can range from highly objective, such as "violin," to highly subjective, such as "romantic." Additionally, tags are usually crowd-sourced or user-defined and may present challenges such as ambiguity and subjectivity. There have been efforts that focus on addressing these issues, capturing semantic similarities both directly from audio and from text [5, 6, 7].

Initially, efforts to automate the music annotation process focused on leveraging low-level acoustic features, such as zero-crossing rate [8], and mid-level features, such as rhythm [9], combined with standard classifiers such as k-Nearest-Neighbors. As time progressed, deep learning models have significantly improved auto-tagging performance through carefully designed network architecture [1, 10, 11, 12]. While still requiring data-label pairs, they are much better at capturing complex patterns and representations within the audio data, allowing for better accuracy. With the increase in computing power, large-scale models have been trained on open-access audio datasets [13] to learn powerful representations that could be used as feature extractors for various downstream tasks in the audio domain [14, 15, 16]. Additionally, there has been an abundance of research aimed at reducing the size and complexity of network architectures by leveraging various pre-trained embeddings [17, 18]. However, previous work in transfer learning for music auto-tagging generally relies on the complete training dataset and may be dependent on specific datasets. Consequently, these methods are often unable to learn an effective classifier from very limited data.

Few-shot learning presents a promising alternative. It refers to a type of machine learning problem where a computer program learns from only a limited number of examples containing supervised information for the target task [3]. One common formulation of this problem is the N-way-K-shot classification, where N-way represents the number of classes (in our case: tags) the model needs to learn, and K-shot refers to the number of training samples per class the model uses for learning. In the current literature, few-shot learning is primarily approached in two ways: meta-learning and transfer learning.

Meta-learning approaches aim to simulate the inference process during training by partitioning the training data into support and query sets [19, 20, 21, 22]. Transfer learning approaches, on the other hand, involve using pre-trained models as feature extractors and fine-tuning them on the unseen few-shot datasets. Recent research in image classification has demonstrated that transfer learning approaches can yield competitive results comparable to state-of-the-art meta-learning approaches

[23], particularly when a diverse set of pre-trained embeddings is combined [24]. One advantage of transfer learning approaches is that because they are trained on a different set of data, they tend to generalize well and thus can perform well with minimal data, which is particularly important for addressing long-tailed tags with a low number of labeled samples. Therefore, we focus on using pre-trained embeddings in a few-shot setting for multi-label music classification.

3 Experimental Setup

This section details the experimental setup used to evaluate our few-shot classifier design. We describe the methods employed, the dataset and pre-trained features utilized, and the experimental procedures. The primary goal of our experiments is to investigate the performance of few-shot classifiers under various conditions and compare them to state-of-the-art models.

3.1 Methods

We employ transfer learning approaches by fine-tuning four different types of pre-trained embeddings on a commonly used music auto-tagging dataset through a simple linear probe. A linear probe is a linear classifier trained on frozen features extracted from a large pre-trained model. It helps to assess how well these embeddings encode task-relevant information with minimal training. Much of the research in few-shot classification [19, 20, 21, 22, 23, 24] frames the target task as a simple multi-class problem; however, music auto-tagging is inherently a multi-label task [4], i.e., multiple tags can be active for the same input. To align our few-shot classifier with the nature of the task, we employ Binary Cross-Entropy loss, coupled with a fully connected layer followed by a Sigmoid activation in our simple linear probe, similar to Wang et al. [25]. This approach essentially trains a separate logistic regression classifier for each class using a one-vs-rest strategy, thus allowing the concurrent presence of multiple tags in a single sample.

The experimental setup is detailed in Figure 1. To simulate the few-shot setting, we sample $N \times K$ (N: number of classes, K: number of training samples per class) training data points from the training set. The entire validation set is used for early stopping, and the test set is employed to compute the performance metrics of our few-shot classifiers. We use the mean

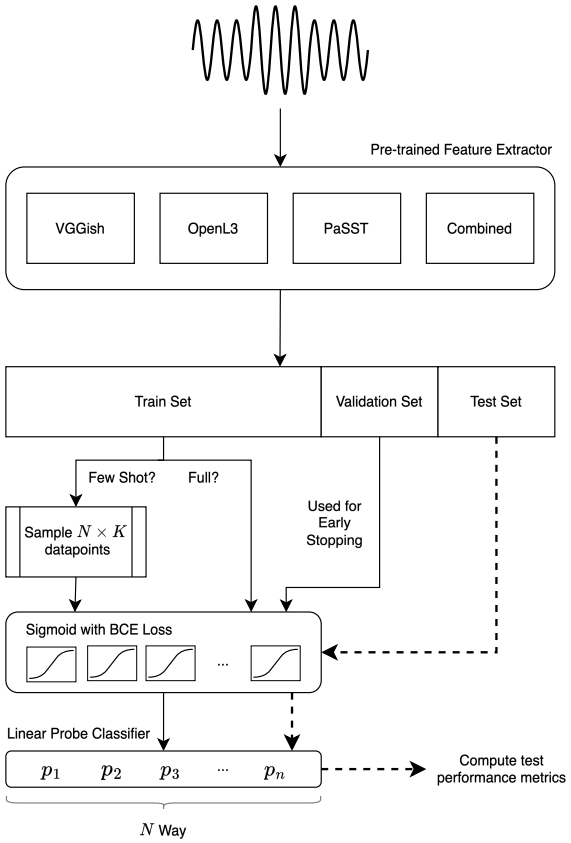


Fig. 1: Experimental setup: few-shot linear probes are trained on $N \times K$ number of data points, and full linear probes are trained with all the data in the training set. Test performance metrics are calculated using probabilities from the Sigmoid activation function on the full test set.

average precision (mAP) and the mean area under the receiver operating characteristics curve (AUC-ROC) to measure the classifier performance to be comparable with previous work[1, 26]. Both mAP and AUC-ROC are calculated as the mean of the binary scores for each class, averaged across all classes. mAP is the mean of the binary average precision score of each class, focusing on how many of the tagged songs are correctly tagged. AUC-ROC, on the other hand, is the mean of the binary AUC-ROC scores for each class, focusing more on how many of the songs that do not have a specific tag are wrongfully tagged. Using both metrics gives us a comprehensive evaluation of our classifiers.

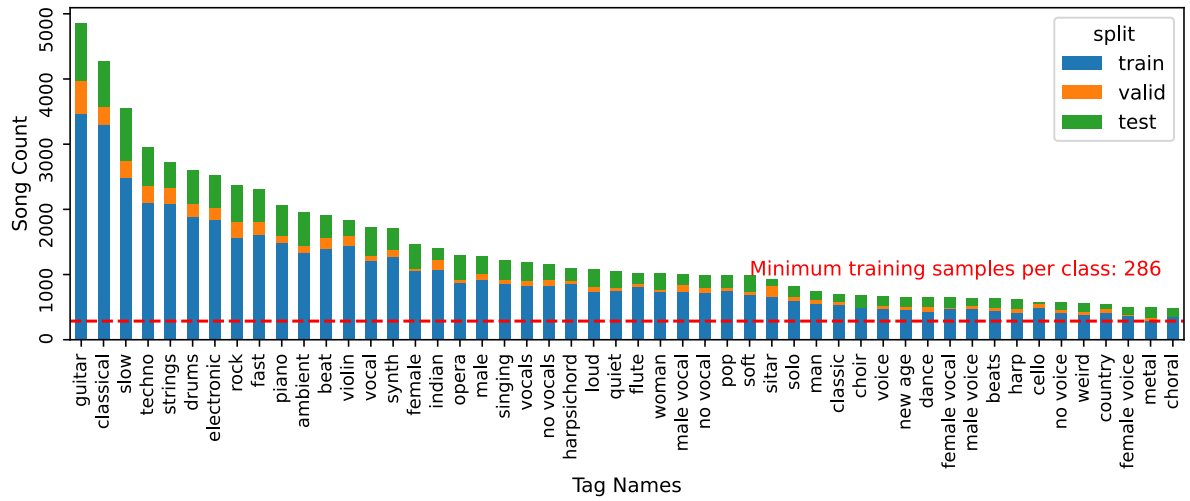


Fig. 2: Distribution of Top 50 tags of MagnaTagATune by split.

3.2 Data

The MagnaTagATune dataset [27] is one of the most widely used datasets for automatic music tagging systems. It comprises 25,863 audio clips, each lasting 29.1 seconds. The dataset includes 188 tags, ranging from specific instruments like *violin* or *drums* to broader musical qualities such as *classical* or *jazz*. Most automatic tagging models only utilize the 50 most frequent tags [1], discarding the songs not associated with these tags. In our study, we adopt the same split of the top 50 tags as [1] to ensure our performance metrics are comparable to previous work. Figure 2 shows the distribution of the top 50 tags and their composition in the data split. It is worth noting that even the tag with the least amount of training samples has 286 associated clips. At the same time, the remaining 138 tags that are not in the top 50 tags only have 534 clips in total. The tag with the most training data only has 55 associated clips. This long-tail distribution could also potentially benefit from few-shot approaches.

3.3 Pre-trained Features

All pre-trained features are extracted using the whole clip. We have chosen three pre-trained embeddings —-VGGish, OpenL3, and PaSST— because they are extracted from models trained on large-scale audio datasets and have demonstrated strong performance in previous studies [15, 28, 29]. Each of these embeddings will be introduced in more detail below.

For these embeddings, multiple frames of embeddings are extracted over the 29.1-second audio clips. These frames are then aggregated into the mean and standard deviation of all frames at each position of the embedding, which results in a vector that is twice as long as its original dimension. This ensures that all four embeddings have the same time resolution, allowing for a fair comparison.

3.3.1 VGGish

VGGish [15] is a pre-trained model based on the VGG architecture [30], which is known for its deep convolutional layers initially designed for image recognition tasks. It has been adapted for the audio domain by training on a vast collection of YouTube videos. The released VGGish embeddings were trained on 16 kHz mono audio and were post-processed through a PCA transformation and quantization to 8 bits. VGGish produces 128-dimensional embeddings. In our setup, the input feature vector is 256-dimensional after aggregation.

3.3.2 OpenL3

OpenL3 [28] is a model trained through self-supervised learning of audio-visual correspondence in videos. The

model we use to extract embeddings from is trained specifically on a music subset of the AudioSet dataset, designed to extract a richer set of audio features. OpenL3 produces 512-dimensional embeddings. In our setup, the input feature vector is 1024-dimensional after aggregation.

3.3.3 PaSST

PaSST [29], or Patchwise Self-Supervised Transformer, is a novel method that optimizes and regularizes transformers on audio spectrograms. The authors propose an efficient training method called Patchout, which also serves as a regularizer, and disentangles the transformer's positional encoding into a time and frequency positional encoding. We use the model variant that has pre-trained time positional encoding for 30 s to accommodate for our audio input length. PaSST produces 768-dimensional embeddings. In our setup, the input feature vector is 1536-dimensional after aggregation.

3.3.4 Combined Embeddings

Inspired by Chowdhury et al. [24], who demonstrated that building a few-shot classifier on top of multiple pre-trained models results in better classification performance, we combine the three embeddings introduced above. Each embedding is first normalized and aggregated into mean and standard deviation, then concatenated into a 2816-dimensional feature vector.

3.4 Experiments

We conducted a series of experiments to comprehensively evaluate our few-shot classifier design, with each research question addressed in the subsections below. In this section, we refer to models trained with the pre-trained features using the entire training set as "full linear probe" models, and models trained with $N \times K$ training data points as "few-shot linear probe" models.

3.4.1 Exp. 1: Full Linear Probe

To understand how a simple model architecture using pre-trained features can compare to state-of-the-art models trained from scratch, we trained our simple multi-label neural network with four inputs: VGGish, OpenL3, PaSST, and the concatenation of all three. The resulting four models were trained with the entire training dataset from the original training split. We then compared the performance of the best-performing full linear probe with state-of-the-art auto-tagging models that is trained from audio[1], as well as linear probes using other pre-trained models from previous work [26].

3.4.2 Exp. 2: Training Data Efficiency

Our next research question examines how data efficient our four types of pre-trained features are, e.g., what impact the number of training samples per class has on the classifier performance. We approach this question by examining two metrics: mAP and the correlation coefficient between the trained weights of the few-shot linear probe and the full linear probe. For this experiment, we trained 5 different 50-way few-shot classifiers with each type of pre-trained features, gradually increasing the number of training samples per class (K) from 1 to 20. This results in 20 50-way few-shot classifiers. mAP is calculated as described in sect. 3.1. Similar to the approach used in [24], weight correlation is calculated by taking the L1 norm of the weights at each embedding position. For instance, if we use VGGish to train our linear probe, we have 256 such norms. We then calculate the Pearson correlation coefficient between these norms from the 20-shot classifiers and those from the full linear probe, expecting that a higher correlation indicates performance closer to the full linear probe, the (presumably) best case.

3.4.3 Exp. 3: Impact of Number of Tags

In practice, when music catalog owners need to expand their tag vocabulary, they often rely on their internal mapping of tags and music. Thus, the ability of a model to learn effectively from a few samples is crucial. This experiment explores how the number of classes impacts performance and how many songs are needed for our simple few-shot classifier to perform competitively.

We conducted an ablation study by varying the number of classes (N) and the number of samples per class (K). For each pre-trained embedding, we trained a series of few-shot classifiers, with N and K in the below ranges:

- N Range: [2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
- K Range: [1, 5, 10, 15, 20]

This resulted in 55 few-shot classifiers for each type of pre-trained embedding, with the number of training data points ranging from 2 ($N = 2, K = 1$) to 1000 ($N = 50, K = 20$). To maintain comparability, classes

Approach	mAP	AUC-ROC
LP_{VGGish}	0.42	0.90
LP_{OpenL3}	0.43	0.90
LP_{PaSST}	0.45	0.91
$LP_{Combined}$	**0.47**	**0.92**
SC-CNN w/ Res [1]	0.46	0.91
Probing Jukebox [26]	0.41	**0.92**

Table 1: Performance comparison between our full linear probes, a state-of-the-art auto-tagging model trained from audio, and a probing model using Jukebox embeddings [26].

and training samples were added incrementally. For instance, a 5-way classification model includes the two tags used in a 2-way classification model, and a 20-shot model includes the five samples used in a 5-shot model.

4 Results and Discussion

This section presents the results from the experiments listed above.

4.1 Exp. 1: Full Linear Probe

For our first experiment, we report in Table 1 the performance metrics of our linear probes LP∗ trained and evaluated on 29.1-second audio excerpts with the same data split as proposed by Won et al. [1]. We also provide two state-of-the-art performance comparison systems on the top 50 tags of the MagnaTagATune dataset: (i) SC-CNN is a 7-layer CNN with a fully connected layer and residual connections, trained on 3.69-second audio excerpts, and (ii) Jukebox is a probing classifier with inputs as embeddings extracted by a pre-trained codified audio language model, trained on 29.1-second audio excerpts [26]. We observe that our linear probing classifiers perform competitively with both the end-to-end model trained specifically for this task and the linear probe with additional language information encoded. This indicates that all the pre-trained embeddings used here contain very relevant information for music tagging and a low-complexity linear classifier is sufficient to achieve results close to the state-of-the-art with such inputs. The PaSST embedding seems to outperform OpenL3 and VGGish embeddings. The better performance of the linear probe classifier with the combined (concatenated) embedding input suggests

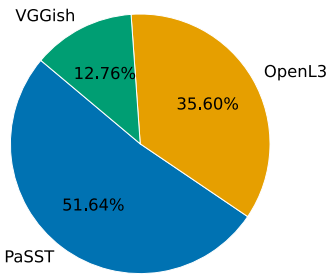


Fig. 3: Percentage of weight magnitude sum of each pre-trained embeddings in the overall weight magnitude sum of the combined feature.

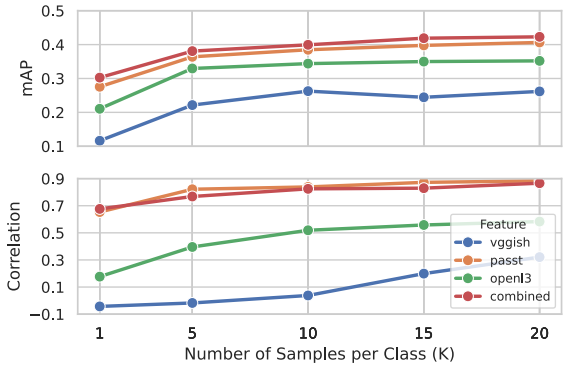


Fig. 4: Performance comparison of 50-way classifiers dependent on the number of training samples per class: Top: mAP, Bottom: Correlation coefficient between 20-shot probe weights and full probe weights.

that our few-shot network learns different aspects from each embedding, and combining them allows these features to complement each other, resulting in higher overall performance metrics.

To investigate the impact of each pre-trained embedding on $LP_{Combined}$, we calculated the percentage of the weight magnitude of each single embedding in the weight magnitude of the combined embedding. Figure 3 shows that the PaSST embeddings have the highest impact, followed by OpenL3 and VGGish.

4.2 Exp. 2: Training Data Efficiency

The second experiment investigates the data efficiency of each pre-trained features through observing the impact of the number of training samples per class for a 50-way classifier.

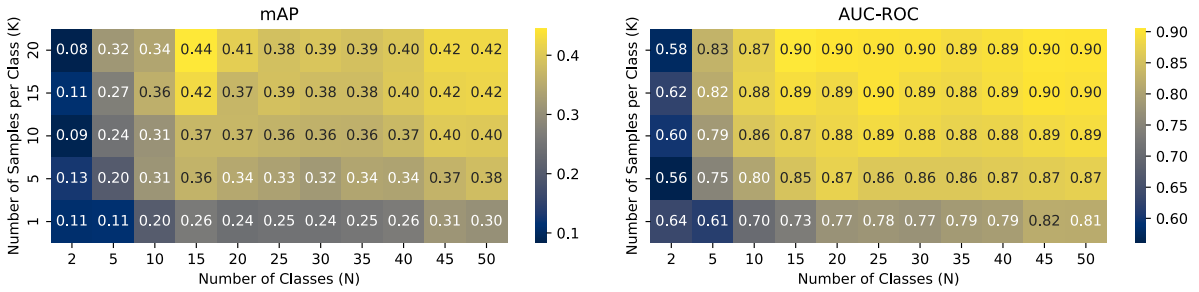


Fig. 5: Heatmap of how mean average precision (mAP) and area under the receiver operating characteristics curve (AUC-ROC) changes as the number of classes (horiz.) and the number of training samples per class (vert.) increase. Only the best-performing model is shown.

The top plot in Fig. 4 shows the mAP of the linear probing classifiers with different types of pre-trained embeddings as the number of samples per class increases. Unsurprisingly, the mAP increases in most cases with increasing number of training samples, but increasing the number of training samples per class beyond 10 only yields small performance gains. LP_{PaSST} tends to consistently show the best performance across all training set sizes, followed by LP_{OpenL3} and LP_{VGGish}. Using combined embeddings as input yields better metrics than any of the single embeddings. Figure 4 bottom shows the weight correlation between the linear probe classifiers trained with different training set sizes and the weights of the full linear probe that was trained with all available training data. Assuming that the latter represents the best-case outcome, the plot shows increasing correlation with increasing number of training samples. LP_{PaSST} and $LP_{Combined}$, in particular, show a high correlation starting from a low number of training samples, indicating highly efficient training data utilization.

4.3 Exp. 3: Impact of Number of Tags

While the results presented above focused on the performance in a 50-way scenario, i.e., a classification into 50 classes, in this experiment we evaluate how the number of classes (or labels) influences the performance. As the number of target classes can vary dependent on the specific task, understanding the impact of this can be crucial in a real-world scenario.

Figure 5 depicts the changes in mAP and AUC-ROC as the number of samples per class increases for classifiers with varying numbers of classes. The metrics shown represent the best performing pre-trained features discussed in the previous subsection. As reported in Exp. 2, performance increases as we increase the number of training samples per class K for the classifiers with the same number of classes N (the results of Exp. 2 can be revisited in the right column of the matrix). On the other hand, as we increase the number of classes while keeping the number of samples per class constant, performance tends to be stable except for very low N's and K's. Even though increasing the number of classes adds negative examples for the previous classes in the training set, it seems like simple binary classifiers are not dramatically impacted by this training set imbalance.

It is important to note that all tags in the test set always have the same amount of positive and negative samples in all our experiments because we use the entire test set to calculate performance metrics. If we constrained the test set to include only songs with at least one associated tag, it would inflate the AUC-ROC with increasing class number. This is because AUC-ROC involves measuring the false positive rate (FPR), calculated by dividing the number of false positives by the sum of true negatives and false positives. With the constraint, moving from a lower number of classes to a higher number of classes increases the number of true negatives in the test set, thus exaggerating the AUC-ROC.

5 Conclusion

In this paper, we explored the potential of few-shot learning for multi-label music auto-tagging by training a linear probe using pre-trained embeddings. The pre-trained embeddings we experimented with are VGGish,

OpenL3, PaSST, and all three combined. Our results demonstrate that combining embeddings from different pre-trained models can yield both performance on par with state-of-the-art models and improved training data efficiency. We showed that using pre-trained embeddings with linear probing is also a very suitable approach for few-shot settings and systematically evaluated the performance loss introduced by the few-shot scenario. Our results emphasize that few-shot learning utilizing pre-trained embeddings can be an effective strategy for music tagging and a viable solution for long-tail tags with limited labeled samples.

To our knowledge, this paper is the first attempt to investigate multi-label few-shot learning for music auto-tagging. We believe this approach can effectively handle situations where only a limited amount of data is available. In the future, we plan to explore methods for incorporating novel classes learned from a few-shot classifier into existing auto-tagging models, in order to build a truly customizable auto-tagging system.

References

[1] Won, M., Ferraro, A., Bogdanov, D., and Serra, X., "Evaluation of CNN-based automatic music tagging models," in *Proceedings of 17th Sound and Music Computing*, 2020.

[2] Marques, G., Gouyon, F., Langlois, T., and Domingues, M. A., "Three current issues in music autotagging," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[3] Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M., "Generalizing from a Few Examples: A Survey on Few-shot Learning," *ACM Computing Surveys (CSUR)*, 53(3), pp. 1–34, 2020.

[4] Fu, Z., Lu, G., Ting, K. M., and Zhang, D., "A Survey of Audio-Based Music Classification and Annotation," *IEEE Transactions on Multimedia*, 13(2), pp. 303–319, 2011.

[5] Hennequin, R., Royo-Letelier, J., and Moussallam, M., "Audio Based Disambiguation of Music Genre Tags," in *Proceedings of the International Society for Music Informational Retrieval Conference (ISMIR)*, 2018.

[6] Epure, E. V., Khlif, A., and Hennequin, R., "Leveraging knowledge bases and parallel annotations for music genre translation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[7] Epure, E. V., Salha, G., and Hennequin, R., "Multilingual Music Genre Embeddings for Effective Cross-lingual Music Item Annotation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[8] McKinney, M. F. and Breebaart, J., "Features for Audio and Music Classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2003.

[9] Laurier, C. and Herrera, P., "Automatic Detection of Emotion in Music: Interaction with Emotionally Sensitive Machines," in *Machine learning: Concepts, methodologies, tools and applications*, 2012.

[10] Pons, J. and Serra, X., "Musicnn: Pre-trained Convolutional Neural Networks for Music Audio Tagging," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[11] Lee, J., Park, J., Kim, K. L., and Nam, J., "Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[12] Choi, K., Fazekas, G., and Sandler, M., "Automatic tagging using deep convolutional neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[13] Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M., "Audio Set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[14] Koh, E. S. and Dubnov, S., "Comparison and Analysis of Deep Audio Embeddings for Music Emotion Recognition," in *AAAI Workshop on Affective Content Analysis*, 2021.

[15] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al., "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[16] Lee, J. and Nam, J., "Multi-Level and Multi-Scale Feature Aggregation Using Pretrained Convolutional Neural Networks for Music Auto-tagging," *IEEE signal processing letters*, 24(8), pp. 1208–1212, 2017.

[17] Ding, Y. and Lerch, A., "Audio Embeddings as Teachers for Music Classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[18] Ding, Y. and Lerch, A., "Embedding Compression for Teacher-to-Student Knowledge Transfer," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[19] Snell, J., Swersky, K., and Zemel, R., "Prototypical Networks for Few-shot Learning," *Advances in Neural Information Processing Systems*, 2017.

[20] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M., "Learning to Compare: Relation Network for Few-Shot Learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[21] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T., "Meta-learning with Memory-augmented Neural Networks," in *International Conference on Machine Learning*, 2016.

[22] Finn, C., Abbeel, P., and Levine, S., "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, pp. 1126–1135, PMLR, 2017.

[23] Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C., and Huang, J.-B., "A Closer Look at Few-shot Classification," in *International Conference on Learning Representations*, 2019.

[24] Chowdhury, A., Jiang, M., Chaudhuri, S., and Jermaine, C., "Few-shot Image Classification: Just Use a Library of Pre-trained Feature Extractors and a Simple Classifier," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[25] Wang, Y., Bryan, N. J., Cartwright, M., Bello, J. P., and Salamon, J., "Few-shot Continual Learning for Audio Classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[26] Castellon, R., Donahue, C., and Liang, P., "Codified Audio Language Modeling Learns Useful Representations for Music Information Retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[27] Law, E., West, K., Mandel, M. I., Bay, M., and Downie, J. S., "Evaluation of Algorithms Using Games: The Case of Music Tagging," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2009.

[28] Arandjelovic, R. and Zisserman, A., "Look, Listen and Learn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[29] Koutini, K., Schlüter, J., Eghbal-Zadeh, H., and Widmer, G., "Efficient Training of Audio Transformers with Patchout," in *Proceedings of INTERSPEECH*, 2021.

[30] Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.