AUDIO EMBEDDINGS AS TEACHERS FOR MUSIC CLASSIFICATION

Yiwei Ding
Music Informatics Group
Georgia Institute of Technology
yding402@gatech.edu

Alexander Lerch
Music Informatics Group
Georgia Institute of Technology
alexander.lerch@gatech.edu

ABSTRACT

Music classification has been one of the most popular tasks in the field of music information retrieval. With the development of deep learning models, the last decade has seen impressive improvements in a wide range of classification tasks. However, the increasing model complexity makes both training and inference computationally expensive. In this paper, we integrate the ideas of transfer learning and feature-based knowledge distillation and systematically investigate using pre-trained audio embeddings as teachers to guide the training of low-complexity student networks. By regularizing the feature space of the student networks with the pre-trained embeddings, the knowledge in the teacher embeddings can be transferred to the students. We use various pre-trained audio embeddings and test the effectiveness of the method on the tasks of musical instrument classification and music auto-tagging. Results show that our method significantly improves the results in comparison to the identical model trained without the teacher's knowledge. This technique can also be combined with classical knowledge distillation approaches to further improve the model's performance.

1. INTRODUCTION

The classification of music has always been a widely popular task in the field of Music Information Retrieval (MIR). Music classification serves as an umbrella term for a variety of tasks, including music genre classification [1], musical instrument classification [2], and music auto-tagging [3]. The last decade has seen dramatic improvements in a wide range of such music classification tasks due to the increasing use of artificial neural networks [4–7].

One major contributing factor to these impressive accomplishments is the increased algorithmic complexity of the machine learning models which also means that the training process requires an increased amount of data. As not all tasks have this abundance of annotated data, transfer learning has been widely and successfully applied to various music classification tasks [8]. In transfer learning, a model is first pre-trained on a large-scale dataset for a (source) task that is somewhat related to the (target) task and then fine-tuned with a comparably smaller dataset of the target task [9]. This enables knowledge to be transferred across datasets and tasks. Transfer learning has been repeatedly shown to result in state-of-the-art performance for a multitude of MIR tasks [10–12].

Another side effect of the increasing model complexity is the slow inference speed. One way to address this issue is model compression by means of knowledge distillation. Here, a low-complexity (student) model is trained while leveraging the knowledge in the high-complexity (teacher) model [13, 14]. The teacher-student paradigm has met with considerable success in reducing the model complexity while minimizing performance decay [15, 16].

In this study, we integrate ideas and approaches from both transfer learning and knowledge distillation and apply them to the training of low-complexity networks to show the effectiveness of knowledge transfer for music classification tasks. More specifically, we utilize pre-trained audio embeddings as teachers to regularize the feature space of low-complexity student networks during the training process. Thus, the main contributions of this paper are a systematic study of

- the effectiveness of various audio embeddings as teachers for knowledge transfer,
- different ways to apply the knowledge transfer from teachers to students, and
- the impact of data availability on the performance of the investigated systems.

The models and experiments are publicly available as open-source code. [1]

2. RELATED WORK

This section first briefly introduces transfer learning and knowledge distillation, which are both often used to transfer knowledge between tasks and models, respectively, and then surveys the application of feature space regularization in the training of neural networks.

2.1 Transfer Learning

In transfer learning approaches, a model is pre-trained on a source task with a large dataset and subsequently fine-tuned on a (different but related) target task with a (typically

Attribution: Y. Ding and A. Lerch, "Audio Embeddings as Teachers for Music Classification", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://github.com/suncerock/EAsT-music-classification. Last accessed on June 21, 2023.

smaller) dataset [9]. By utilizing the knowledge learned from the source task, models trained following the transfer learning paradigm can often achieve significantly better results than the same models trained directly on the target task [17]; this is especially the case if these models have a large number of parameters and the training data for the target task is limited. In the case where fine-tuning the whole model might be too computationally expensive, another way to do transfer learning is to use the pre-trained embeddings and train only the classification head. This allows for a separation of the tasks of computing the embeddings and the classification itself.

Transfer learning has been successfully applied to a wide variety of areas ranging from computer vision [18, 19] to natural language processing [20]. In MIR, transfer learning has been used for a multitude of target tasks [8, 10, 11, 21]. Besides fine-tuning the whole model, pre-trained embeddings such as VGGish [22] and Jukebox [23] have also shown good performance on many tasks including auto-tagging [12, 24], instrument classification [4, 12], and music emotion recognition [12, 24–26].

One disadvantage of transfer learning is the slow inference speed. In most cases, the model has a large number of parameters, which means that both fine-tuning (if done on the whole model) and inference potentially lead to a high computational workload.

2.2 Knowledge Distillation

Approaches for knowledge distillation aim at model compression, i.e., reducing the complexity of the network. The knowledge of a (usually high-complexity) pre-trained network (the teacher) is transferred to a different (low-complexity) network (the student) during the training phase, in which the student not only learns from the ground truth labels but also from the teacher predictions. This is achieved by adding a "distillation loss" term to the student's loss function to learn from the teacher's prediction [13, 14].

The most popular distillation loss is the Kullback-Leibler divergence between the logits of the student and the teacher, with a hyperparameter called temperature to soften the probability distribution of the teacher's prediction over classes [13]. The soft target provides more "dark" knowledge than the ground truth hard label [27, 28]. The Pearson correlation coefficient has also been proposed as a distance measure between the logits as an alternative to the Kullback-Leibler divergence [29].

Besides learning from logits, the student network can also try to learn from the feature map from the intermediate layers of the teacher network [30–32]. As the feature maps of the student and teacher do not necessarily share the same dimension and the same size, a variety of ways to match the feature space of the student and the teacher have been proposed [31, 33, 34]. Therefore, feature-based knowledge distillation has more flexibility than the logits-based traditional approach, which, at the same time, also makes it more challenging to find the best way of matching the feature space [35, 36].

2.3 Feature Space Regularization

Feature-based knowledge distillation is a technique of regularizing the feature space of the network during training. Besides knowledge distillation, there exists a wide variety of other ways to implement regularization. One example is contrastive learning, which aims at contrasting the features of instances with positive labels against negative labels [37, 38]. Contrastive learning has been shown to improve the performance of neural networks on music auto-tagging [39, 40] and music performance assessment [41].

Regularizing the feature space using pre-trained audio embeddings has also been reported to be effective in music classification [42] and music source separation [43], where Hung and Lerch proposed to use pre-trained embeddings to help structure the latent space during training. This technique is similar to but different from both transfer learning and knowledge distillation. In transfer learning, the same model is used on two different datasets, and a typical setting is that knowledge from the large dataset will be transferred to the small dataset. In knowledge distillation, only one dataset is used and the typical setting is that the knowledge will be transferred from a large model to a small model. In comparison, regularizing the feature space using embeddings requires neither the dataset nor the model to be the same, yet still allows to transfer knowledge learned by the teacher model from a large dataset to the low-complexity student network for a different (small) dataset.

3. METHODS

Inspired by the promising preliminary results of prior work [42], we integrate the idea of transfer learning and knowledge distillation by using pre-trained audio embeddings as teachers to regularize the feature space of the student network during training. The overall pipeline is illustrated in Figure 1.

3.1 Loss Function

Similar to knowledge distillation [13], we rewrite our loss function as

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{pred}} + \lambda\mathcal{L}_{\text{reg}} \tag{1}$$

where $\mathcal{L}_{\text{pred}}$ is the loss function for conventional neural network training, $\mathcal{L}_{\text{reg}}$ is the loss function that measures the distance between the student network's feature map and the pre-trained embeddings, and $\lambda \in [0, 1]$ is a weighting hyper-parameter.

3.2 Regularization Location

Different stages in a neural network output different feature maps, and the optimal location to apply regularization continues to be controversially discussed in feature-based knowledge distillation [36]. In this study, we investigate either regularizing only the final feature map before the classification head as shown in Figure 1 or regularizing the feature maps at all stages of the student network.

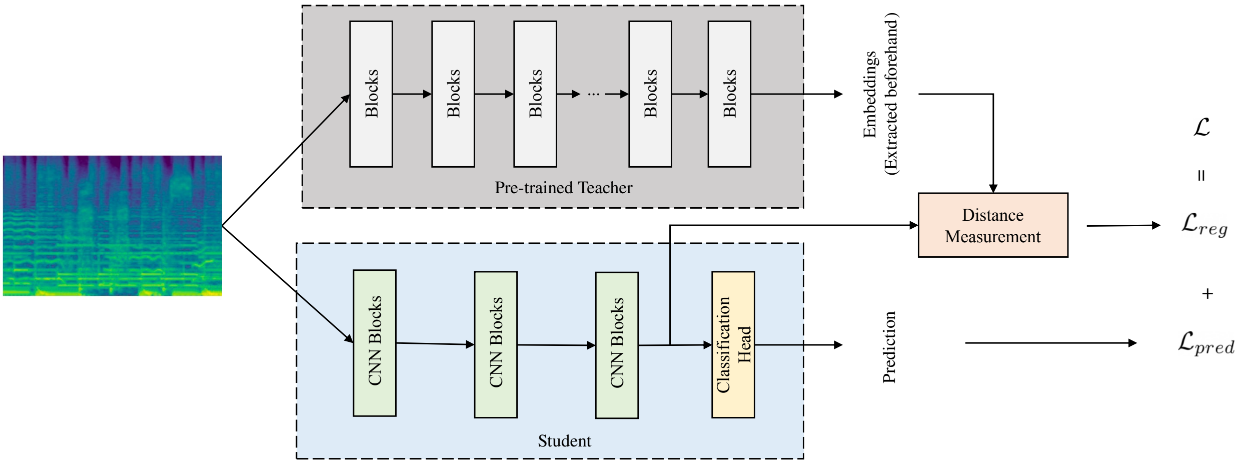

Figure 1: Overall pipeline of training a model by using pre-trained embeddings as teachers. The training loss is a weighted sum (weighting factor omitted in the figure) of prediction loss and regularization loss. The regularization loss measures the distance between pre-trained embedding and the output feature map after the feature alignment. During inference, only the bottom part with the blue background is used.

3.3 Feature Alignment

To measure the distance between the student feature map $l \in \mathbb{R}^{T_\text{s} \times C_\text{s}}$ and the pre-trained teacher embeddings $v \in \mathbb{R}^{T_\text{t} \times C_\text{t}}$ which might have different numbers of time frames (i.e., $T_\text{s} \neq T_\text{t}$), we first align the intermediate feature map with the pre-trained embeddings in time by repeating the one with fewer time frames, then compute the distance for each frame and finally average them along the time axis.

3.4 Distance Measure

Considering that pre-trained embeddings and feature maps have often different dimensionalities, the use of distance measures that are independent of dimensionality allows for easier application.

3.4.1 Cosine Distance Difference

Cosine distance difference [2] as proposed in previous work [42, 43] measures the difference in the cosine distance between pairs of samples. Given n pairs of samples of single-time-frame features $l_1, l_2, ..., l_n$ and pre-trained embeddings $v_1, v_2, ..., v_n$, the cosine distance difference for one pair is

$$D_{ij} = |d_{\cos}(l_i, l_j) - d_{\cos}(v_i, v_j)|, \tag{2}$$

and the distance for this time frame is averaged among all pairs.

3.4.2 Distance Correlation

Distance correlation was proposed as a generalization of classical correlation to measure the independence between two random vectors in arbitrary dimensions [44]. It is capable of handling features of different dimensionality; furthermore, correlation-based distance measures have been shown to be effective in knowledge distillation [29, 32]. Using the same notation as above, we define

$$a_{ij} = \|l_i - l_j\|, \tag{3}$$

$$\bar{a}_{i.} = \frac{1}{n}\sum_{j=1}^{n} a_{ij}, \quad \bar{a}_{.j} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad \bar{a}_{..} = \frac{1}{n^2}\sum_{i,j=1} a_{ij} \tag{4}$$

$$A_{ij} = a_{ij} - \bar{a}_{i.} - \bar{a}_{.j} + \bar{a}_{..} \tag{5}$$

where $i, j \in \{1, 2, ..., n\}$, and similarly, $b_{ij} = \|v_i - v_j\|$ and $B_{ij} = b_{ij} - \bar{b}_{i.} - \bar{b}_{.j} + \bar{b}_{..}$.[3] The distance for the time frame is then

$$\mathcal{L}_\text{reg} = 1 - \mathcal{R}_n^2(l, v) = 1 - \frac{\mathcal{V}_n^2(l, v)}{\sqrt{\mathcal{V}_n^2(l, l)\mathcal{V}_n^2(v, v)}} \tag{6}$$

where

$$\mathcal{V}_n^2(l, l) = \frac{1}{n^2}\sum_{i,j=1}^{n} A_{ij}^2, \quad \mathcal{V}_n^2(v, v) = \frac{1}{n^2}\sum_{i,j=1}^{n} B_{ij}^2,$$
$$\mathcal{V}_n^2(l, v) = \frac{1}{n^2}\sum_{i,j=1}^{n} A_{ij}B_{ij}.$$

Note that $\mathcal{V}_n^2(l, l)$ and $\mathcal{V}_n^2(v, v)$ will be 0 if and only if all the n samples of features (or embeddings) within one batch are identical [44], which we assume not to occur here.

To optimize both distance measures during training, block stochastic gradient iteration is used, which means that the distance is computed over mini-batches instead of the whole dataset [45, 46]. With stochastic approximation, the computational complexity of the distance measure for n samples is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(mn)$ where m is the batch size.

It is worth mentioning that both distance measures ensure that if the distance is zero, the feature maps would

[2] has been referred to in previous work as Distance-based Regularization (Dis-Reg) [42, 43].

[3] Eq. (3) uses 2-norm following the implementation in `https://github.com/zhenxingjian/Partial_Distance_Correlation`.

differ from the pre-trained embeddings by only an orthogonal linear transformation, which can be modeled in a single linear layer. Therefore, if the regularization loss is zero, the student would have the same performance as the teacher in classification.

4. EXPERIMENTAL SETUP

We test the effectiveness of using pre-trained embeddings as teachers on two different tasks, datasets, and models with four different pre-trained embeddings as follows.

4.1 Tasks, Datasets, Models, and Metrics

4.1.1 Musical Instrument Classification with OpenMIC

Musical instrument classification is a multi-label classification problem. We use the OpenMIC dataset [2], which provides weakly labeled audio snippets of length 10 s. Following prior work [4, 49], we use the suggested test set and randomly split 15% of the training data as the validation set, resulting in 12,692 training observations, 2,223 validation observations, and 5085 test observations. To ensure a consistent sample rate, the audio is resampled to 32 kHz [5, 49]. As the dataset is not completely labeled, i.e., parts of the labels are missing and not labeled as positive or negative, the missing labels are masked out when computing the loss function as suggested in previous work [5, 10, 49].

We use receptive field regularized ResNet (CP-ResNet) [5] for this task, as it reaches state-of-the-art performance when trained only on the OpenMIC dataset (i.e., neither trained with transfer learning nor trained with any knowledge distillation). CP-ResNet has a ResNet-like structure [19] with an added hyper-parameter ρ to control the maximum receptive field of the ResNet. We set $\rho = 7$ to match the setting which provides the best results in the original work [5].

The results are reported with the metrics mean Average Precision (mAP) and F1-score. The F1-score is calculated in a macro fashion, which means that for each instrument, the F1-score is computed for both the positive labels and the negative labels and then averaged, and the final F1-score is the mean of the F1-scores of all instruments. The detection threshold for the prediction is set to 0.4 following previous work [5].

4.1.2 Music Auto-Tagging with MagnaTagATune

Similar to musical instrument classification, music auto-tagging is also a multi-label classification problem. We use the MagnaTagATune dataset [3] for this task, which comes with audio clips of approximately 29.1 s. Following previous work, we use only the top 50 labels and exclude all the songs without any positive label from the dataset [7, 50]. For comparability, the data split is adopted from previous work, with audio files in the directories '0' to 'b' being the training set, 'c' being the validation set, and 'e' and 'f' being the test set [48, 51], resulting in 15,247 training clips, 1,529 validation clips, and 4,332 test clips.

We apply a modified fully convolutional neural network (FCN) [6] to this task. It is the simplest model among the benchmark models for the MagnaTagATune dataset [48] and consists of several convolution and max-pooling layers. To further reduce the complexity of the model, we apply the MobileNet-like modification [52] to the network by breaking the 3×3 convolutions into depth-wise separable convolutions and 1×1 convolutions.

The results are evaluated with mAP and ROC-AUC.

4.2 Pre-trained Embeddings

4.2.1 VGGish

VGGish [22] is a widely used embedding in MIR, with a VGG network [53] being trained on a large number of Youtube videos. The open-source PyTorch implementation is used to extract VGGish features [4] which by default extracts 128 principle components and then quantizes them to 8 bit. The time resolution is 960 ms.

4.2.2 OpenL3

The OpenL3 embedding [54,55] is trained on a music subset of AudioSet [56] in a self-supervised paradigm. The audio embeddings are extracted using the open-source Python package `OpenL3` [5] with the dimensionality being 512. To keep consistent with VGGish, the time resolution is set to 960 ms.

4.2.3 PaSST

PaSST [10] is a 7-layer transformer trained on AudioSet for acoustic event detection. It applies the structure of a vision transformer [16, 57] and proposes the technique of Patchout to make the training efficient. We use the open-source code [6] released by the authors to extract the 768-dimensional embeddings. The time resolution is also set to 960 ms.

4.2.4 PANNs

PANNs [11] include several convolutional neural networks and are also trained on AudioSet for acoustic event detection. We use the default CNN14 model from the official repository [7] . The embedding dimensionality is 2048. Different from other embeddings, PANNs provide only one global embedding for each clip of audio. Pilot experiments have shown that extracting the embeddings for short segments and concatenating them does not improve performance.

4.3 Systems Overview

The following systems are evaluated for comparison:

- Baseline: CP ResNet (on OpenMIC) and Mobile FCN (on MagnaTagATune) trained without any extra regularization loss.

[4] `https://github.com/harritaylor/torchvggish`. Last accessed on April 4, 2023.

[5] `https://github.com/marl/openl3/tree/main`. Last accessed on April 4, 2023

[6] `https://github.com/kkoutini/PaSST/tree/main`. Last accessed on April 4, 2023.

[7] `https://github.com/qiuqiangkong/audioset_tagging_cnn`. Last accessed on April 4, 2023.

OpenMIC	None		VGGish		OpenL3		PaSST		PANNs	
	mAP	F1	mAP	F1	mAP	F1	mAP	F1	mAP	F1
CP ResNet* [5]	.819	.809	-	-	-	-	-	-	-	-
SS CP ResNet* [5]	.831	.822	-	-	-	-	-	-	-	-
Teacher$_{\text{LR}}$	-	-	.803	.799	.803	.798	**.858**	**.837**	.853	**.834**
KD (w/ mask) **	-	-	.829	.820	.823	.813	.851	.834	.848	.823
EAsT$_{\text{Cos-Diff}}$	-	-	.838	.824	**.838**	.820	.837	.822	.836	.814
EAsT$_{\text{Final}}$	-	-	**.842**	**.828**	.835	**.822**	.847	.830	.849	.828
EAsT$_{\text{All}}$	-	-	.836	.823	.835	**.822**	.845	.827	.845	.827
EAsT$_{\text{KD}}$	-	-	.836	.825	.836	.821	.852	.834	**.857**	.831

MagnaTagATune	None		VGGish		OpenL3		PaSST		PANNs	
	mAP	AUC	mAP	AUC	mAP	AUC	mAP	AUC	mAP	AUC
FCN† [6]	.429	.900	-	-	-	-	-	-	-	-
Mobile FCN	.437	.905	-	-	-	-	-	-	-	-
Teacher$_{\text{LR}}$	-	-	.433	.903	.403	.890	**.473**	**.917**	**.460**	.911
KD	-	-	.447	.911	.439	.907	.454	.912	.448	.909
EAsT$_{\text{Cos-Diff}}$	-	-	.446	.906	.438	.907	.453	.912	.453	.911
EAsT$_{\text{Final}}$	-	-	.454	**.912**	.447	.910	.459	.912	.449	.909
EAsT$_{\text{All}}$	-	-	**.455**	.911	**.452**	**.911**	.458	.913	.457	.911
EAsT$_{\text{KD}}$	-	-	.441	.908	.437	.904	.461	.915	.459	**.912**

Table 1: Results on OpenMIC (above) and MagnaTagATune (below) dataset for different models regularized with different pre-trained embeddings. Best performances are in bold, and best results excluding the teachers are underlined. *Reported results [5], SS means being trained with shake-shake regularization [47]. **When using KD, the missing labels in OpenMIC were masked to avoid potentially adding more training data. †Results from the open-source re-implementation [48].

- Teacher$_{\text{LR}}$: logistic regression on the pre-trained embeddings (averaged along the time axis), which can be seen as one way to do transfer learning by freezing the whole model except for the classification head.
- KD: classical knowledge distillation where the soft targets are generated by the logistic regression.
- EAsT$_{\text{Cos-Diff}}$ (for Embeddings-As-Teachers): feature space regularization as proposed by Hung and Lerch that uses cosine distance difference and regularizes only the final feature map [42].
- EAsT$_{\text{Final}}$ and EAsT$_{\text{All}}$: proposed systems based on distance correlation as the distance measure, either regularizing only at the final stage or at all stages, respectively.
- EAsT$_{\text{KD}}$: a combination of classical knowledge distillation and our method of using embeddings to regularize the feature space. The feature space regularization is done only at the final stage.

We perform a search of λ for each of the EasT systems and choose the best-performing value on the validation set. [8]

5. RESULTS

This section presents the results of different systems and their performance in the case of limited training data.

5.1 Results on OpenMIC and MagnaTagATune

Table 1 shows the results on the OpenMIC and the MagnaTagATune datasets.

We can observe that the models trained with the extra regularization loss consistently outperform the non-regularized ones on both datasets, with all features, and all regularization methods. This means that the knowledge in the embeddings is successfully transferred to the student networks and consistently enhances the performance.

Although EAsT$_{\text{Final}}$ appears to give better results on the OpenMIC dataset while EAsT$_{\text{All}}$ seems to have slightly better performance on the MagnaTagATune dataset, the difference between them is very small, meaning that the model does not benefit significantly from being regularized by pre-trained embeddings at earlier stages where the feature maps are still relatively low-level.

The results for the teacher systems show that the older VGGish and OpenL3 embeddings are clearly outperformed by the more recently proposed embeddings PaSST and PANNs. In fact, the teacher systems for the newer embeddings perform so strongly that the students can rarely outperform them, while the student systems trained with VGGish and OpenL3 provide better results than the corresponding teachers. We can see that whether the teachers themselves have an excellent performance or not, students benefit from learning the additional knowledge from these embeddings, and the students' upper limit is not bounded by the performance of teachers.

Comparing KD and the EAsT$_{\text{Final}}$ or EAsT$_{\text{All}}$ systems,

[8] For all the hyperparameters, please refer to the config files in our GitHub.

Model	Parameters (M)	Iteration / s
VGGish	72.1	172.2
OpenL3	4.69	117.9
PaSST	86.1	18.7
PANNs	79.7	70.6
Mobile FCN	0.34	319.3
CP ResNet	5.52	205.3

Table 2: Comparison of the model complexity.

we can see that with VGGish and OpenL3 embeddings, regularizing the feature space provides better results than simply using the teachers' soft targets. On the other hand, for the PaSST and PANNs embeddings, classical knowledge distillation provides competitive results. The possible reason is that the soft targets given by "weak" teachers might have provided too much incorrect information to the students while the high-quality soft targets generated by the "strong" teachers provide good guidance for the students' training.

The combination system EAsT_{KD} gives us better results with PaSST and PANNs embeddings (with the exception of no noteworthy improvement with the PaSST embedding on the OpenMIC dataset) while for VGGish and OpenL3 embeddings, the performance is not as good as $\text{EAsT}_{\text{Final}}$ or EAsT_{All} in most cases. This observation is in accordance with our speculation that traditional knowledge distillation performs best with a "strong" teacher. While learning from audio embeddings benefits a student network even more in the presence of a "strong" teacher, learning from "weak" embeddings can still improve the model's performance.

5.2 Comparison of Model Complexity

Table 2 lists the number of parameters as well as rough inference speed measurements [9] of the models.

The numbers of parameters only take the backbone structure (i.e., excluding the final classification head) into account so that it does not vary across datasets with different numbers of classes. Iterations per second are tested with 128×1000 input spectrograms.

We can see that Mobile FCN and CP ResNet are much faster in inference than pre-trained models.

5.3 Limited Training Data

To investigate the impact of limited training data on our methods, we present the system performances for reduced training data, i.e., for 25%, 50%, and 75% of the original training data. The results are shown in Figure 2. We use VGGish and PaSST as the pre-trained embeddings.

We can observe that limiting the training data has the greatest impact on the baseline systems, which show the biggest performance drop.

On the OpenMIC dataset, $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ have similar decreases in mAP, and the KD system is less affected. An interesting finding is that when the VGGish embedding is used, KD shows better performance for limited data amounts while it is outperformed by $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ when the whole OpenMIC dataset is available. This means using embeddings as teachers might still require a sufficient amount of data to have good guidance on the student models.

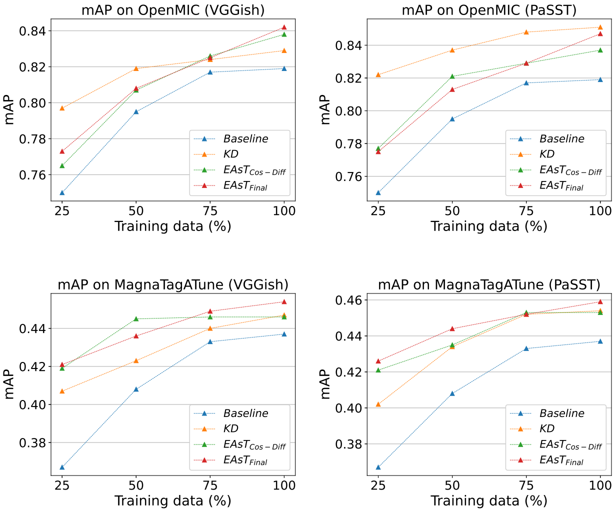

Figure 2: Results with limited training data on two datasets.

On the MagnaTagATune dataset, however, the $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ systems show less performance decay than either KD or the baseline when the training data is limited. This suggests that in our training settings, there is no certain answer to which method is least affected by the lack of training data, and the answer might be dependent on specific tasks, models, and data.

6. CONCLUSION AND FUTURE WORK

In this paper, we explored the use of audio embeddings as teachers to regularize the feature space of low-complexity student networks during training. We investigated several different ways of implementing the regularization and tested its effectiveness on the OpenMIC and MagnaTagATune datasets. Results show that using embeddings as teachers enhances the performance of the low-complexity student models, and the results can be further improved by combining our method with a traditional knowledge distillation approach.

Future work will investigate the performance of our method on a wider variety of downstream tasks and embeddings. Moreover, as there have been a wide variety of models to extract audio and music embeddings, we speculate that using an ensemble of different pre-trained embeddings also has considerable potential. Finally, the flexibility of feature-based knowledge distillation offers a wide range of possible algorithmic modifications. Our focus will be on evaluating different distance measures and regularizing the network using features from different stages of the teacher network instead of using only the output embeddings.

[9] reference GPU: NVIDIA 2070 Super

7. REFERENCES

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[2] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open data-set for multiple instrument recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 438–444.

[3] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 387–392.

[4] S. Gururani, M. Sharma, and A. Lerch, "An attention mechanism for musical instrument recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 83–90.

[5] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1987–2000, 2021.

[6] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 805–811.

[7] T. Kim, J. Lee, and J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 366–370.

[8] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 141–149.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Proceedings of INTERSPEECH 2022*, 2022, pp. 2753–2757.

[11] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[12] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 256–263.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[14] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[15] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCV)*, 2019, pp. 2902–2911.

[16] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 10 347–10 357.

[17] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 491–507.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[19] 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.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 4171–4186.

[21] P. Alonso-Jiménez, D. Bogdanov, and X. Serra, "Deep embeddings with essentia models," in *Late Breaking Demo of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[22] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[23] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020. [Online]. Available: http://arxiv.org/2005.00341

[24] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 88–96.

[25] E. S. Koh and S. Dubnov, "Comparison and analysis of deep audio embeddings for music emotion recognition," in *AAAI Workshop on Affective Content Analysis*, 2021.

[26] D. Bogdanov, X. Lizarraga Seijas, P. Alonso-Jiménez, and X. Serra, "Musav: a dataset of relative arousal-valence annotations for validation of audio models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 650–658.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[28] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[30] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[31] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1921–1930.

[32] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5007–5016.

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4133–4141.

[34] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[35] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[36] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 3048–3068, 2022.

[37] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[38] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[39] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 673–681.

[40] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 825–833.

[41] P. Seshadri and A. Lerch, "Improving music performance assessment with contrastive learning," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 634–641.

[42] Y.-N. Hung and A. Lerch, "Feature-informed embedding space regularization for audio classification," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2022, pp. 419–423.

[43] ——, "Feature-informed latent space regularization for music source separation," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2022.

[44] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[45] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1686–1716, 2015.

[46] X. Zhen, Z. Meng, R. Chakraborty, and V. Singh, "On the versatile uses of partial distance correlation in deep learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 327–346.

[47] X. Gastaldi, "Shake-shake regularization of 3-branch residual networks," in *Workshop Track of the International Conference on Learning Representations (ICLR)*, 2017.

[48] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," in *Proceedings of the Sound and Music Computing (SMC)*, 2020, pp. 331–337.

[49] H.-H. Chen and A. Lerch, "Music instrument classification reprogrammed," in *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2023, pp. 345–357.

[50] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," 2019. [Online]. Available: http://arxiv.org/1906.04972

[51] M. Won, S. Chun, O. Nieto, and X. Serrc, "Data-driven harmonic filters for audio representation learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 536–540.

[52] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[54] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[55] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[56] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.