LOOK, LISTEN, AND LEARN MORE: DESIGN CHOICES FOR DEEP AUDIO EMBEDDINGS

Jason Cramer[1,⋆] *Ho-Hsiang Wu*[1,⋆] *Justin Salamon*[1,2] *Juan Pablo Bello*[1,2]

[1]Music and Audio Research Laboratory, New York University, USA
[2]Center for Urban Science and Progress, New York University, USA
{jtcramer, hohsiangwu, justin.salamon, jpbello}@nyu.edu

ABSTRACT

A considerable challenge in applying deep learning to audio classification is the scarcity of labeled data. An increasingly popular solution is to learn deep audio embeddings from large audio collections and use them to train shallow classifiers using small labeled datasets. Look, Listen, and Learn (L^3-Net) is an embedding trained through self-supervised learning of audio-visual correspondence in videos as opposed to other embeddings requiring labeled data. This framework has the potential to produce powerful out-of-the-box embeddings for downstream audio classification tasks, but has a number of unexplained design choices that may impact the embeddings' behavior. In this paper we investigate how L^3-Net design choices impact the performance of downstream audio classifiers trained with these embeddings. We show that audio-informed choices of input representation are important, and that using sufficient data for training the embedding is key. Surprisingly, we find that matching the content for training the embedding to the downstream task is not beneficial. Finally, we show that our best variant of the L^3-Net embedding outperforms both the VGGish and SoundNet embeddings, while having fewer parameters and being trained on less data. Our implementation of the L^3-Net embedding model as well as pre-trained models are made freely available online.

***Index Terms*—** Audio classification, machine listening, deep audio embeddings, deep learning, transfer learning.

1. INTRODUCTION

Machine listening is an active area of research concerned with the development of computational methods to derive meaning from sound, in which the use of deep learning has seen growing popularity and success [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. However, obtaining sufficient labeled data is difficult and costly for audio-related tasks, where annotation often requires listening to entire recordings [14, 15, 16, 17]. As a result, most existing machine listening datasets are relatively small and narrowly focused [1, 13, 18], and only recently has the community seen a large-scale dataset with the introduction of AudioSet [7]. While a significant improvement on prior efforts, noisy and incomplete labels, as well as vocabulary mismatches, means that there is still a range of machine listening problems for which AudioSet remains insufficient to support end-to-end learning.

Recently the community has turned to transfer learning [19] as a solution to the issue of data scarcity. In this family of techniques, an embedding model is trained to solve a task for which a large amount of data is available and then used to generate input features to train a model on a target task for which limited data are available. There are multiple examples of deep embedding solutions in the literature. VGGish [6, 9, 10, 20] is an audio embedding produced by training a modified VGGNet model [21] to predict video tags from the Youtube-8M dataset [22]. SoundNet [3, 11, 23, 24, 25, 26] generates embeddings by training a deep audio classifier to predict the output of a deep image classifier, pre-trained on standard image recognition datasets such as ImageNet [27]. More recent approaches successfully leverage triplet learning [28] under contextual constraints, such as the temporal proximity of audio samples in video streams [12].

A notable approach, known as Look, Listen, and Learn (L^3-Net), uses a self-supervised learning method to train a model to detect if a video frame corresponds to an audio frame [4, 5]. The approach stands out in several respects. First, it does not require any annotated data. Second, it produces powerful embeddings leading to state-of-the-art performance in sound classification, outperforming SoundNet and other non-embedding approaches [4]. Third, this remarkable performance is obtained while utilizing a relatively simple convolutional architecture. Given the above, the L^3-Net paradigm has the potential to become a go-to solution for a wide range of machine listening tasks for which labeled data is scarce. However, a number of important design choices are only briefly discussed and their effect is not fully characterized, a problem compounded by the fact that there is no open L^3-Net implementation currently available. These choices may have a non-negligible effect on the performance and computational cost of the model.

In this paper we seek to address this gap by systematically exploring important implementation choices of the L^3-Net embedding, in the context of sound event classification. We address the following questions: (1) is it beneficial to use an audio-informed input representation? (2) is it important that the training data for the embedding match the downstream classification task? and (3) how much training data is sufficient for training the embedding? In addition, we compare L^3-Net against other popular embeddings, VGGish and SoundNet, on three well-known datasets; release an open-source implementation of the approach and its variants; and provide pre-trained embeddings for community use.

2. LOOK, LISTEN AND LEARN (L^3-NET)

The L^3-Net approach [4] proposes a means of learning embeddings via the auxiliary task of audio-visual correspondence (AVC) which aims to determine whether a video image frame and a 1 s audio seg-

[*] Authors contributed equally to this work

The authors would like to thank Relja Arandjelović for his clarification of L^3-Net implementation details, as well as Brian McFee and Mark Cartwright for their valuable feedback. This work is partially supported by NSF awards no 1544753 and 1633259.

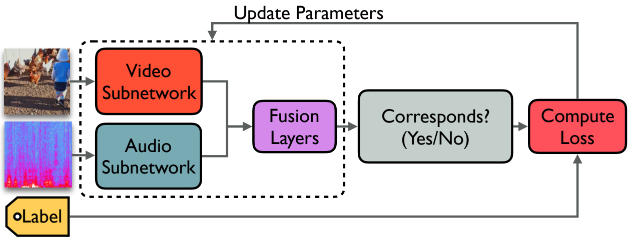

Fig. 1. High-level architecture of L^3-Net.

ment come from the same video and overlap in time. The L^3-Net architecture as shown in Figure 1 has three distinct parts: the vision and the audio subnetworks which extract visual and audio features respectively, and the fusion layers which use both modalities to predict correspondence. Since both matched and mismatched image-audio pairs can be generated automatically from the training data (by taking the image and audio from the same video or from different videos, respectively), no manual labeling is required in order to train the model on this binary classification task.

The audio and vision subnetworks use four blocks of convolutional and max-pooling layers, the outputs of which are flattened, concatenated, and passed to the fully-connected fusion layers to produce the correspondence probability. The audio embedding is obtained from the final output layer of the audio subnetwork, replacing the layer's non-linear activation with a max-pooling layer, the output of which is flattened. The authors use a max-pooling size leading to a final embedding dimensionality of 6144. For further details about the L^3-Net model architecture see [4].

While the embedding holds promise for downstream tasks, there are design choices left unexplained that may impact the efficacy and computational cost of the embedding. To better understand the behavior of the embedding, we explore three design choices that have the potential to impact the performance of the embedding:

2.1. Input representation

The original L^3-Net uses a linear-frequency log-magnitude spectrogram as the input to the audio subnetwork. However, it is more common in the audio machine learning literature to use Mel-frequency log-magnitude spectrograms as input to convolutional networks. Mel spectrograms are designed to capture perceptually relevant information more efficiently with less frequency bands compared to a linear spectrogram [29]. Perhaps more importantly, when a sound is pitch-shifted the pattern created by its harmonic partials change when using a linear frequency scale, whereas with a (quasi) logarithmic frequency scale such as the Mel scale, pitch shifts result in a vertical translation of the same harmonic pattern, meaning that convolutional filters should generalize better when using the latter.

2.2. Training data domain and match to downstream tasks

The authors of L^3-Net use video content which they expect to have a high degree of AVC to train the embedding model. Originally they used the Flickr dataset [3], and subsequently a subset of the AudioSet dataset [5]. The labels provided by AudioSet help to understand the types of content in the videos and how they affect the behavior of the embedding models. The authors use a subset of videos with mostly people playing musical instruments while the downstream tasks contain environmental sound sources. We examine whether matching the audio domain used to train the embedding with the domain of the downstream task improves performance. A priori we expect matching the domains to have a positive effect.

2.3. Amount of training data

The authors train their embedding models with 60M samples, but do not discuss how the amount of training data used affects the efficacy of the embeddings. Since training these models can take significant time and computational resources, it is beneficial to quantify the trade-off between the amount of training data and performance on the downstream classification tasks.

3. EXPERIMENTAL DESIGN

We employ a two-stage experimental design, first training a deep audio embedding, and then evaluating the audio embedding as a feature extractor in a downstream classification task.

3.1. Deep audio embedding model

We use AudioSet [7] to train the L^3-Net audio embedding models. For each 10 s video in AudioSet, we download a 30-fps h.264-encoded video and a 48 kHz FLAC audio file. We were able to acquire $\sim$2M AudioSet videos. For the benefit of other researchers we release the code we have developed for obtaining these videos[1].

We train the embedding models using one of two subsets of AudioSet, a *music* subset and an *environmental* subset. The music subset replicates the one used in [5] which includes videos of people playing musical instruments and using tools, chosen for the expected high level of AVC. The environmental subset includes categories such as human sounds, animal sounds, and other sounds found in natural acoustic environments. We filter the videos using AudioSet labels, obtaining 296K and 195K videos for the music and environmental subsets respectively. We use 80% of the data for training, 10% for validation, and 10% for testing. For training, videos are sampled using the `pescador` [30] framework. For each video, we follow the sampling and augmentation scheme in [4], sampling 224x224 image patches and 1 s audio clips. We generate 60M training samples, 10M validation samples, and 10M testing samples.

We train the embedding models for 300 epochs, with 4096 batches of size 64 per epoch, corresponding to the model seeing 78.6M training samples. We use the Adam optimizer [31] to minimize binary cross-entropy loss with L2 regularization, with an initial learning rate and weight decay factor of 10^{-5}, and Adam parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We compute the spectrograms on the GPU with TensorFlow [32] using `kapre` [2] [33]. We compute HTK Mel-spectrograms [34] with either 128 or 256 Mel bands. The model parameters are chosen from the epoch with the highest validation accuracy. Each model took approximately ten days to train on four parallel GPUs. To evaluate whether the embedding model has been sufficiently trained, we look at the binary classification accuracy on the AVC task for the test set of our two AudioSet subsets.

3.2. Downstream task: environmental sound classification

For the environmental sound classifier, we use a multi-layer perceptron (MLP) with two fully-connected hidden layers of size 512 and

[1] `https://github.com/marl/audiosetdl`

[2] The version of `kapre` used for our experiments contained a minor bug affecting how spectrogram energy is normalized. We have verified that this issue does not have any significant effect on our experimental results, and have since fixed this issue in our released embeddings.

128 respectively, and an output layer with a size corresponding to the number of classes in the dataset being evaluated. The MLP is trained to predict the class of a 1 s audio clip (a single embedding frame). At test time we segment the audio clip into overlapping windows, compute their embeddings, sum the class likelihoods output by the MLP over all windows, and take the class with the highest total likelihood as the clip prediction, as per [4]. We experiment with 3 well-known, open datasets:

- UrbanSound8K [1] consists of 8732 audio clips of up to 4 s in length, labeled with one of ten urban environmental sound event categories such as air conditioner, dog bark, and jackhammer. The dataset comes separated into ten equally-sized cross-validation folds.
- ESC-50 [18] consists of 2000 5 s audio clips, labeled with one of 50 environmental sound categories, such as glass breaking, car horn, and wind. The dataset contains 40 examples per category and comes separated into five equally-sized cross-validation folds.
- DCASE 2013 scene classification dataset (SCD) [13] consists of 200 30 s audio clips, labeled with one of ten auditory scenes such as busy street, restaurant, and park. It contains 20 samples per category and comes separated into equally-sized train and test sets.

We consider the following design choices:

- Input representation: We compare embeddings using a linear spectrogram with 257 bins (Linear) used by the original L^3-Net and Mel spectrograms with either 128 (M128) or 256 (M256) Mel bins spanning the entire audible frequency range.
- Training data domain and match to downstream tasks: We look at embeddings trained on the environmental (Env) and music (Music) subsets of AudioSet, representing matched and mismatched conditions (with respect to the downstream tasks) respectively.
- Amount of training data: We evaluate different checkpoints of the best L^3-Net embedding model variant, taken every 2.6M samples.

Finally, we also train classifiers using the SoundNet [3] and VGGish [9] embeddings for comparison, using the pre-trained embedding models provided by their respective authors.

3.3. Methodology for comparing embeddings

For all downstream datasets, we perform cross validation using the predefined splits. We use 10% of the downstream training data for validation, stratified with respect to classes. We compute embeddings from overlapping 1 s windows with a 0.1 s hop (except with SoundNet, implemented with non-overlapping frames). Each design choice is evaluated independently by averaging the results over all other design variations not relevant to the comparison. We use the Wilcoxon signed-rank test [35] with $p < 0.05$ to test for statistical significance.

We use the validation set for early stopping with a patience of 20 epochs, training the MLP for up to 50 epochs. The embeddings are standardized prior to training. For each cross-validation split, we tune the hyperparameters on the validation set over initial learning rates of $\{10^{-5}, 10^{-4}, 10^{-3}\}$ and weight decay factors of $\{10^{-5}, 10^{-4}, 10^{-3}\}$. These models took up to 2 hours to train on a single GPU.

4. RESULTS

4.1. Input representation

The results for different input representations are shown in Figure 2. In all datasets, we see that Mel spectrograms outperform linear spectrograms, with the 256-bin variant performing the best; this advantage is statistically significant on both UrbanSound8K and ESC-50, confirming our hypothesis described in Section 2.1. M128 still performs better than the Linear variant, suggesting that Mel bins indeed capture relevant information in the audio signal more efficiently. In the case of DCASE 2013 SCD, the dataset is so small that all embeddings perform comparably and obtain near-perfect accuracy.

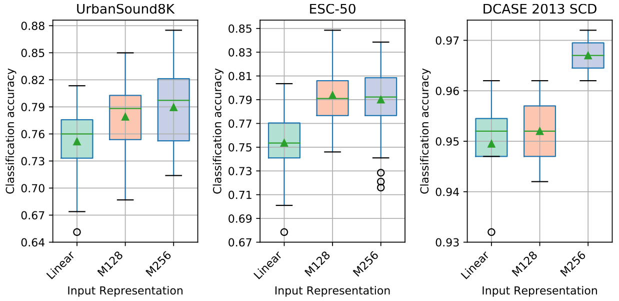

Fig. 2. Classification accuracy vs. input representation.

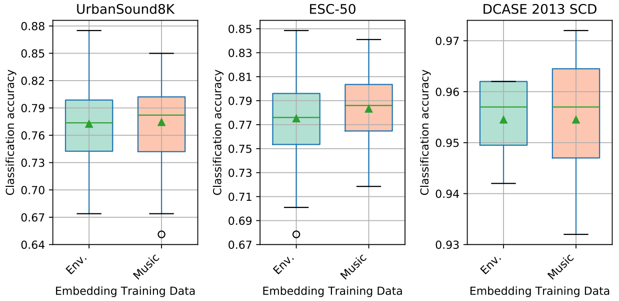

Fig. 3. Classification accuracy vs. training subset.

4.2. Training data domain and match to downstream tasks

Before turning to the downstream task, we first evaluate the impact of using matched/mismatched train/test domains on the performance of L^3-Net on the AVC task itself, presented in Table 1. As expected, the model performs better on AVC when the train and test audio domains are matched. Next, we examine how this influences performance on the downstream classification task as shown in Figure 3. Surprisingly, matching domains has no positive influence on performance, and in the case of ESC-50 it slightly decreases performance. This suggests that it might be more important to use audio content that maximizes the discriminative power of the embedding, independently of the downstream domain. In this case we expect videos of people playing musical instruments to have a greater degree of AVC than environmental videos on average, which is potentially a more important factor influencing the efficacy of the resulting embedding.

4.3. Amount of training data

The results for UrbanSound8K and ESC-50 are shown in the top and bottom plots of Figure 4. For the former, improvements in accuracy exhibit diminishing returns after training the embedding with 13M samples (at around 77%) while for the latter we see diminishing returns after 40M samples (at around 79%). For a resource constrained training scenario, the results suggest that at least 40M samples should be used to train the L^3-Net embedding.

Training Subset	Music Test Acc.	Env. Test Acc
Music	**77.04%**	64.82%
Env.	62.93%	**78.08%**

Table 1. Accuracy of M256 L^3-Net models on the AVC test set.

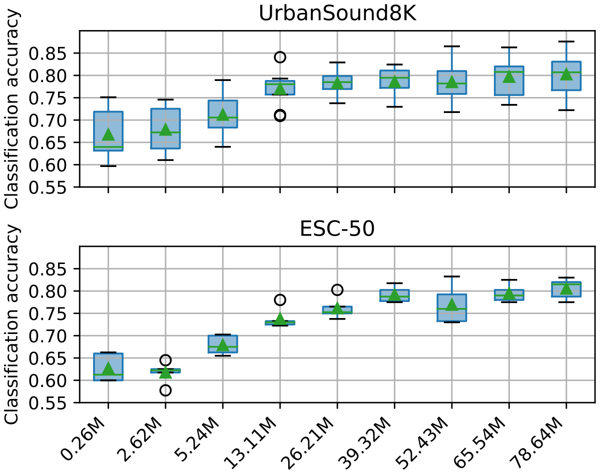

Fig. 4. Classification accuracy vs. number of training samples used to train the embedding model.

4.4. Embedding type: L^3-Net, SoundNet and VGGish

Finally, we compare our best overall L^3-Net embedding model variant (M256 trained on Music) with SoundNet and VGGish, shown in Figure 5. We see that L^3-Net performs best on all three datasets, resulting in a mean classification accuracy of 78.23%, 79.82% and 97% on UrbanSound8K, ESC-50 and DCASE 2013 SCD respectively (the best performance on UrbanSound8K overall is obtained with L3-M256/Env with an accuracy of 79.34%). The improvement over the alternative embeddings is statistically significant with respect to UrbanSound8K and ESC-50: L^3-Net outperforms VGGish on the two datasets by 4.85 and 6.28 points respectively and outperforms SoundNet on the two datasets by 9.48 and 32.16 points respectively. Furthermore, compared to VGGish, L^3-Net has an order of magnitude less parameters (4.7M vs 62M) and is trained on significantly less data (296K vs 70M videos). Both advantages are highly beneficial for scenarios with constrained resources.

The mean classification accuracy obtained by the best L^3-Net variant on each dataset is presented in Table 2 along with the accuracies obtained using SoundNet and VGGish embeddings. Apart from the fact that the Mel-based L^3-Net variants consistently outperform SoundNet and VGGish, it is worth highlighting that by using Mel-based L^3-Net embeddings we are able to train a simple 2-layer MLP that matches the state-of-the-art performance on UrbanSound8K [2], arguably the most challenging of the three datasets.

5. CONCLUSION

In this paper we elucidate the relative importance and impact of different design and training choices on the efficacy of deep audio embeddings, in particular L^3-Net. Our key findings are:

- Using sufficient training data has the largest impact on the efficacy of the embedding for downstream tasks. For L^3-Net, using less than 40M training samples results in a sub-optimal embedding, after which we see improvements with diminishing returns. Since L^3-Net does not require any labeled data, all that is needed is a large video dataset.
- Domain-informed design choices still matter. Using an input representation better suited for audio convnets (Mel spectrograms) outperforms a vanilla audio representation.
- Matching the audio content domain between the embedding and downstream task is not necessarily helpful. Our results suggest it might be more important to use content that is best suited to the embedding training paradigm.
- L^3-Net consistently outperforms VGGish and SoundNet on environmental sound classification. In particular, the model has 10x less parameters compared to VGGish and can be trained using 100x less data while not requiring labels, making it attractive both for general purpose use and for deployment scenarios with constrained resources.

Embedding Model	UrbanSound8K Test Accuracy
L^3-Net M256/Env	**79.34%**
VGGish	73.43%
SoundNet	68.80%
Embedding Model	ESC50 Test Accuracy
L^3-Net M256/Mus	**79.82%**
VGGish	73.54%
SoundNet	47.66%
Embedding Model	DCASE 2013 SCD Test Accuracy
L^3-Net M256/Mus	**97%**
VGGish	93%
SoundNet	76%

Table 2. Test classification accuracy of the best L^3-Net embedding compared to VGGish and SoundNet on each dataset.

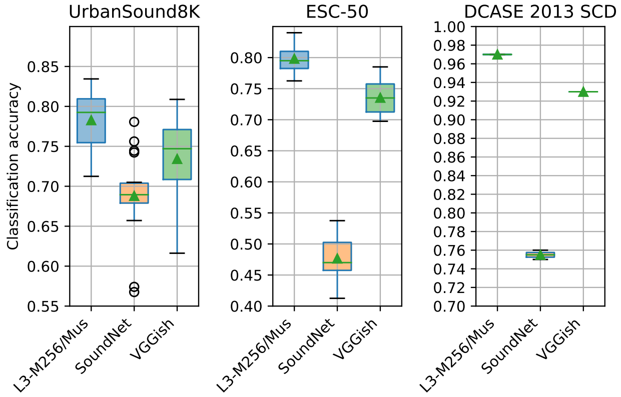

Fig. 5. Classification accuracy using different audio embeddings.

Pre-trained versions of the L^3-Net variants studied in this work are made freely available online[3] for the community to experiment with. For research reproduciblity, the code for running our experiments is also available online[4].

[3] https://github.com/marl/openl3

[4] https://github.com/marl/l3embedding

6. REFERENCES

[1] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM Int. Conf. on Multimedia*. ACM, 2014, pp. 1041–1044.

[2] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[3] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[4] R. Arandjelović and A. Zisserman, "Look, listen and learn," in *IEEE ICCV*, 2017, pp. 609–617.

[5] R. Arandjelović and A. Zisserman, "Objects that sound," in *ECCV*, Munich, Germany, Sep. 2018, pp. 451–466.

[6] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J.F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R.A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *IEEE ICASSP*, New Orleans, LA, USA, Mar. 2017, pp. 131–135.

[7] J. F. Gemmeke, D. P. W. 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 *IEEE ICASSP*, 2017, pp. 776–780.

[8] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE MLSP*, 2015, pp. 1–6.

[9] A. Jansen, J. F. Gemmeke, D. P. W. Ellis, X. Liu, W. Lawrence, and D. Freedman, "Large-scale audio event discovery in one million youtube videos," in *IEEE ICASSP*, 2017, pp. 786–790.

[10] A. Kumar and B. Raj, "Deep CNN framework for audio event recognition using weakly labeled web data," in *Machine Learning for Audio Sig. Proc. Workshop, Conf. on Neural Info. Proc. Sys.*, Long Beach, CA, USA, Dec. 2017.

[11] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[12] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *IEEE ICASSP*, Calgary, Canada, Apr. 2018.

[13] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[14] M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. P. Bello, and O. Nov, "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. 1, 2017.

[15] B. Kim and B. Pardo, "I-sed: an interactive sound event detector," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 2017, pp. 553–557.

[16] B. Kim, "Leveraging user input and feedback for interactive sound event detection and annotation," in *23rd Int. Conf. on Intelligent User Interfaces*. ACM, 2018, pp. 671–672.

[17] B. Kim and B. Pardo, "A human-in-the-loop system for sound event detection and annotation," *ACM Trans. on Interactive Intelligent Systems (TiiS)*, vol. 8, no. 2, pp. 13, 2018.

[18] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *ACM Int. Conf. on Multimedia*, 2015, pp. 1015–1018.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[20] E. J. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open dataset for multiple instrument recognition," in *ISMIR*, 2018.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[22] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[23] S. Pini, O. B. Ahmed, M. Cornia, L. Baraldi, R. Cucchiara, and B. Huet, "Modeling multimodal cues in a deep learning-based framework for emotion recognition in the wild," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ACM, 2017, pp. 536–543.

[24] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, "audeep: Unsupervised learning of representations from audio with deep recurrent neural networks," *J. of Machine Learning Research*, vol. 18, no. 1, pp. 6340–6344, 2017.

[25] S. Chen, Q. Jin, J. Zhao, and S. Wang, "Multimodal multi-task learning for dimensional and continuous emotion recognition," in *Proceedings of the 7th Annual Workshop on Audio/Visual Emotion Challenge*. ACM, 2017, pp. 19–26.

[26] Z. Zhang, J. Wu, Q. Li, Z. Huang, J. Traer, J. H. McDermott, J. B. Tenenbaum, and W. T. Freeman, "Generative modeling of audible shapes for object perception," in *IEEE ICCV*, 2017.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009, pp. 248–255.

[28] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[29] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. of the Acoustical Soc. of America*, vol. 8, no. 3, pp. 185–190, 1937.

[30] B. McFee, C. Jacoby, and E. Humphrey, "pescador," Mar. 2017.

[31] D. Kingma and J.L. Ba, "Adam: A method for stochastic optimization," in *ICLR*, San Diego, CA, USA, May 2015.

[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Conf. on Operating Sys. Design and Implementation*, Berkeley, CA, USA, 2016, pp. 265–283.

[33] K. Choi, D. Joo, and J. Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," in *Machine Learning for Music Discovery Workshop, 34th Int. Conf. on Machine Learning (ICML)*, Sydney, Australia, Aug. 2017.

[34] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge university engineering department*, vol. 3, pp. 175, 2002.

[35] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.