Analyzing the Potential of Pre-Trained Embeddings for Audio Classification Tasks

Sascha Grollmisch
Institute for Media Technology
TU Ilmenau
Ilmenau, Germany
sascha.grollmisch@tu-ilmenau.de

Estefanía Cano
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
cano@idmt.fraunhofer.de

Christian Kehling
Institute for Media Technology
TU Ilmenau
Ilmenau, Germany
christian.kehling@tu-ilmenau.de

Michael Taenzer
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
tzr@idmt.fraunhofer.de

***Abstract*—In the context of deep learning, the availability of large amounts of training data can play a critical role in a model's performance. Recently, several models for audio classification have been pre-trained in a supervised or self-supervised fashion on large datasets to learn complex feature representations, so-called embeddings. These embeddings can then be extracted from smaller datasets and used to train subsequent classifiers. In the field of audio event detection (AED) for example, classifiers using these features have achieved high accuracy without the need of additional domain knowledge. This paper evaluates three state-of-the-art embeddings on six audio classification tasks from the fields of music information retrieval and industrial sound analysis. The embeddings are systematically evaluated by analyzing the influence on classification accuracy of classifier architecture, fusion methods for file-wise predictions, amount of training data, and initial training domain of the embeddings. To better understand the impact of the pre-training step, results are also compared with those acquired with models trained from scratch. On average, the OpenL3 embeddings performed best with a linear SVM classifier. For a reduced amount of training examples, OpenL3 outperforms the initial baseline.**

***Index Terms*—audio classification, transfer learning, audio embeddings, industrial sound analysis, music information retrieval**

I. Introduction

With the availability of large audio datasets in recent years (e.g. *AudioSet* [1]), many audio classification tasks based on deep learning techniques have seen improved classification accuracy. This has mostly occurred in scenarios where data is abundant or easily collected, such as speech or environmental sounds. However, for many audio classification tasks, large scale data collection is unrealistic. One such example in the field of Music Information Retrieval (MIR) is classification of non-western music (e.g. regional traditional music). For musicological analysis, automatic classification can be a powerful tool; however, performing annotations at large scale is restricted, among other things, by the amount of domain knowledge required for the annotations. A similar problem arises in the field of Industrial Sound Analysis (ISA) for acoustic quality control applications [2]. The goal is to assess the health of a given machine by analyzing the sound it produces. However, large amounts of training examples are very costly to obtain for every product, machine, and possible fault. With this in mind, this work focuses on training and evaluating a number of ISA and MIR classifiers, under the premise that only a small amount of annotated training data will be available.

This work has been supported by the German Research Foundation (BR 1333/20-1, CA 2096/1-1, AB 675/2-1)

Transfer Learning (TL) is a powerful technique for building classifiers for small datasets. The main idea behind TL is to pre-train models on tasks where data is abundant, and re-use the knowledge gained during training for tasks where data is limited [3]. There are two main TL approaches: In the first approach, a trained model (obtained with a large dataset) is fine-tuned on the task-specific dataset. In the second approach, learned feature representations, also called embeddings, are used to train additional classifiers on task-specific datasets. TL was shown to be a promising training strategy for a variety of research fields such as Image Classification [4], Natural Language Processing [5], Environmental Sound Classification (ESC) [6]–[8], and several MIR tasks like genre classification [9] and instrument recognition [10].

This work focuses on the second type of TL and analyzes the classification capabilities of learned embeddings for six MIR and ISA tasks. While the MIR tasks mostly deal with music signals that are predominantly harmonic, the ISA tasks deal with signals with transient- and noise-like characteristics. Evaluating both types of tasks allows us to get a broad overview of the performance of these embeddings on a wide variety of audio signals. Results obtained with the embeddings are compared to baseline systems where the classifiers are trained from scratch using only the task-specific dataset. Furthermore, the influence on performance of the choice of classifier, the number of training examples in the final task, the fusion technique for getting file-wise predictions, and the size of the embeddings is investigated.

II. Pre-trained Embeddings for Audio

Initial work on the use of embeddings for audio classification proposed using pre-trained image networks for classification [11], [12] due to the lack of large annotated audio datasets at the time. However, results were still below state-of-the-art performance for the evaluated tasks.

In [9], a VGG network architecture (originally proposed for image classification in [13]) was modified and trained

in a supervised fashion with audio data from the Million Song Dataset [14]. On all the evaluated tasks, the embeddings outperformed a baseline using Mel Frequency Cepstral Coefficients (MFCCs) as input representation. However, performance was still below the state-of-the-art (except for speech/music classification where nearly perfect classification was achieved with all methods).

The **VGGish** embeddings were initially proposed in [15] by training a modified VGG model using mel spectrograms as input. Google later published a trained model[1] using a very weakly labeled dataset which was a preliminary version of the YouTube-8M dataset [16]. Besides having weak labels, the dataset had overlapping tags and events which, while visible in the video, were not necessarily audible. The problems with overlapping tags and very weak labels were addressed with the release of *AudioSet* [1] which contains 2 million audio clips with a duration of 10 seconds each. The sound clips were manually labeled, and an ontology of 527 tags of audible events was proposed. While the labels were correct for the full audio clip, it could still happen that the annotated sounds were not present in some time frames.

The **Kumar** embeddings were proposed as a solution to weak labels in *AudioSet*. In [7], the authors trained a supervised convolutional neural network (CNN) with mel spectrograms as input on the *AudioSet* by pooling the embeddings over time for each file. This is referred to as *early fusion*. A linear Support Vector Machine (SVM) classifier was trained on these embeddings achieving 83.5% accuracy with fine-tuning of the model, and 82.8% without on an ESC task (*ESC50* dataset [17]). Global max pooling across time for each of the 1024 embedding dimensions performed better than average pooling. For this reason, global max pooling is used as the *early fusion* strategy in this work.

Instead of training with weak labels in a supervised fashion, an auxiliary task for creating an audio embedding model (*L3-Net*) was proposed in [6]. A video with matching or non-matching audio is fed into two separate CNN branches for video and audio feature extraction. The outputs of both networks are concatenated, and a fully connected neural network classifier is trained with the CNNs to detect whether the audio corresponds to the video or not. In this way, large amounts of unlabeled videos can be used for training the network in a self-supervised way. The pre-trained audio branch is used for extracting the audio embeddings. Training a linear SVM classifier on the audio embeddings achieved 79.3% on the *ESC50* dataset by summing the class likelihoods over each file, and picking the highest for file-wise prediction. This approach is referred to as *late fusion*.

The **OpenL3** embeddings were proposed in [8] as an extension to *L3-Net*. Different models were trained on the *AudioSet* in the same self-supervised way as *L3-Net* to evaluate the impact on classification accuracy of different design choices such as embedding size (512 vs. 6144), input representation (linear magnitude spectrogram, mel spectrograms with 128 and 256 bins), and trained domain (music vs. environmental audio). The **OpenL3** embedding models were compared to **VGGish** and *SoundNet* [18] embeddings for various ESC datasets, outperforming both. On the *ESC50* dataset, 79.82% classification accuracy was achieved with a feedforward neural network classifier with two hidden layers (512 and 128 units, respectively), and Rectified Linear Unit (ReLU) activation. Results show that the domain of the training data does not necessarily need to correspond to the final task domain.

[1]Pre-trained models and extraction methods:
VGGish: https://github.com/tensorflow/models/tree/master/research/audioset/vggish
Kumar: the FFT size was changed to 1024 to achieve the same results on *ESC50* as in [7]. https://github.com/anuragkr90/weak_feature_extractor
OpenL3: https://pypi.org/project/openl3/

TABLE I
DATASET DESCRIPTION FOR EACH TASK

Task	Classes	Files	Dur. (min)	Test set	Rand basel.
T1	5	185	70	no	20.0%
T2	5	50	310	no	20.0%
T3	4	7527	577	no	25.0%
T4	3	2378	42	yes	33.3%
T5	3	1521	10	yes	33.3%
T6	5	150	150	yes	20.0%

III. EXPERIMENTAL DESIGN

A. Tasks and Datasets

Three MIR and three ISA tasks were selected to cover a wide variety of application fields and signal characteristics. A description is provided below for each task, baseline model and dataset used. Furthermore, a short summary of all the datasets is presented in Table I. The dataset splits have been standardized for all presented experiments to get comparable results between all embeddings and networks trained from scratch. Therefore, the baseline results may differ slightly from the ones previously reported in the literature.

1) Task 1: Ensemble Size Classification in Music (T1): This task deals with counting the number of active instruments in a music recording. The ACMUS-MIR dataset,[2] which includes recordings of string music from the Colombian Andes, was selected for evaluation. The task was defined as a 5-class classification problem, covering solo recordings (one instrument) to large ensembles (five or more instruments). The best performing model reported in [19] is used as baseline, where the best classification result (80.7% file-wise accuracy) was obtained with a feedforward neural network. The dataset has been slightly extended since the initial publication. The same baseline system achieves 81.0% on version 1.1 of the dataset.

2) Task 2: Musical Instrument Family Recognition (T2): For the task of instrument family recognition, we use an in-house dataset called *DB-MTC* [20], which contains 50 commercial recordings of different composers of Western classical music. Each recording is a polyphonic piece of music composed for one instrument family and is, hence,

[2]ACMUS-MIR instrumental format dataset: https://zenodo.org/record/3268961

monotimbral. The five instrument families are woodwinds, brass, piano, vocal, and strings. As a baseline, we intended to use the instrument recognition CNN model proposed in [21]. However, this model showed a tendency to overfit on *DB-MTC* leading to a classification accuracy of 72%. Therefore, we removed two of the original four convolutional blocks to decrease the number of trainable parameters to 10%. With this modification, state-of-the-art performance was achieved with a file-wise accuracy of 94%. Full details about the baseline model can be found in the study website.[3]

3) Task 3: Speech Music Classification (T3): This task is an extended version of speech/music classification which considers four classes: speech, solo singing, choir, and instrumental music. The dataset of ethnomusicological field recordings from [22] is used here.[4] Initially, the model proposed in [22] was considered as a baseline. This model achieved a final accuracy of 94% with data enrichment from several speech/music datasets and augmentation via pitch shifting and time stretching. Additionally, [22] reports that a multilayer perceptron with 16 units trained on **VGGish** embeddings achieved 86.7%. For comparability between the other tasks of this paper, no additional augmentation methods or other datasets were used. The reduced amount of training data led to overfitting and unstable training with the initial model from [22].[3] Therefore, the smaller CNN architecture from [19] is used as baseline, leading to a file-wise accuracy of 88.6%.

4) Task 4: Classification of Operational States in Electric Engines (T4): Electric engines are used in a variety of products such as industrial fans and car seats. This task deals with classifying three operational states of such engines: "good", "heavy-load" and "broken". The *IDMT_ISA_ELECTRIC_ENGINE* dataset[5] is used for this task. As a baseline, the system using a feedforward neural network proposed in [2] is applied. The proposed baseline achieved an average file-wise accuracy of 97.2% on the test set with six different background noises while being trained only on recordings without background noise.

5) Task 5: Metal Surface Classification (T5): Metal balls inside ball bearings may suffer from abrasion leading to damaged surfaces. In this study, the *IDMT_ISA_METAL_BALL* dataset for surface detection is used, which includes three surface conditions: "eloxed", "coated", "broken".[6] As baseline, the feedforward neural network proposed in [2] is used. This model achieved 98.8% file-wise accuracy on the balanced test set.

6) Task 6: Plastic Material Classification (T6): Changes or faults inside plastic material products can potentially be detected by analyzing their acoustic response from being struck. The *IDMT_ISA_PUCKS*[7] dataset is used in this study, and includes the acoustic response of several plastic pucks printed from four different materials. Furthermore, recordings of background noise without any pucks were added as an additional class. Each one minute recording contains several hit events of the same puck. The exact times and quantity of these hit events during each recording are unknown. This characteristic distinguishes this task from the others since the relevant sound events are not audible during the entire file. The CNN proposed in [23] was trained without background noise and tested only on noisy recordings. Without additional task-specific pre-processing, the CNN achieved a file-wise accuracy of 91.5% at the highest background noise level.

[3]Detailed results: https://acmus-mir.github.io/publication/embeddings20/

[4]https://github.com/matijama/field-recording-db

[5]https://www.idmt.fraunhofer.de/en/publications/isa-electric-engine.html

[6]https://www.idmt.fraunhofer.de/en/publications/isa-metal-balls.html

[7]https://www.idmt.fraunhofer.de/en/publications/isa-pucks.html

B. Evaluated Embeddings and classifiers

VGGish, **Kumar**, and **OpenL3** embeddings were chosen for this study since they have already shown state-of-the-art performance on the *ESC50* dataset. **VGGish** are embeddings trained in a supervised fashion with weak labels, the **Kumar** embeddings are trained in a supervised fashion considering weak labels and with improved annotations from *AudioSet*, and **OpenL3** are self-supervised embeddings learned from *AudioSet*. Available source code and models were used to extract all the embeddings.[1] The published default values for sampling rate, window and hop sizes, as well as additional parameters have been kept in all experiments. The sampling rate for **VGGish** is only 16 kHz compared to 44.1 kHz (**Kumar**) and 48 kHz (**OpenL3**). **VGGish** are the smallest embeddings with 128 values, **Kumar** comprises 1024 values, and **OpenL3** can be extracted with 512 or 6144 values. We use **OpenL3** embeddings with 512 output values trained on music data with mel spectrogram (256 bins) as input since this configuration achieved the best accuracy over evaluated datasets [8]. All embeddings are normalized between 0 and 1 on the training set. The calculated normalization values are applied to the validation and test set.

The validity of the processing pipeline has been confirmed for all embeddings using *ESC50* dataset leading to results comparable to the ones reported in the initial papers. However, a tendency to overfit was observed during the training of the feed forward neural network classifier used in [8] (two hidden layers of size 512 and 128 as well as an output layer) on **OpenL3** embeddings. A modification to this classifier is proposed here, where dropout [24] of 0.5 was added between the hidden layers and before the classification layer. This led to an improved accuracy from the reported 79.8% to 81.35% on the *ESC50* dataset. This classifier is referred to as *D512* in the remainder of this paper. Additionally, a smaller non-linear classifier denoted as *D128* was used, containing one hidden layer (128 units), ReLU activation and dropout of 0.5 before the classification layer. *D128* has 20% of the parameters of *D512*. As a third classifier, the commonly used linear SVM [6], [7] was chosen with regularization parameter C set to 1 (default). These three classifiers were selected to evaluate the influence of classifier complexity on the overall performance. To achieve file-wise results, both *early fusion* (on a feature level) and *late fusion* (on a prediction level) methods are evaluated. For each task, the performance is compared with

TABLE II
CLASSIFICATION ACCUARCY IN PERCENT USING 80% OF THE DATASETS FOR TRAINING WITH VGGISH (V), OPENL3 (O) AND KUMAR (K) EMBEDDINGS

Emb.	T1	T2	T3	T4	T5	T6	Avg
Baseline	81.0	94.0	88.6	**95.7**	**99.1**	**89.1**	91.2
V-SVM	69.5	**100**	85.1	33.3	36.3	20.0	57.4
V-D128	75.0	**100**	88.8	33.3	33.9	20.0	58.4
V-D512	73.0	**100**	88.9	33.3	33.3	20.0	58.2
O-SVM	**84.5**	97.0	91.5	93.0	97.1	84.7	**91.3**
O-D128	82.0	96.0	**92.0**	87.8	96.8	80.0	89.3
O-D512	83.5	96.0	91.7	91.4	97.4	78.0	89.4
K-SVM	79.5	99.0	87.8	66.8	96.2	52.7	80.3
K-D128	75.5	97.0	89.1	73.9	94.4	43.3	78.9
K-D512	75.0	97.0	89.1	75.6	95.0	47.3	79.9

the baseline obtained by training a neural network solely on the task specific dataset.

C. Experiments

Three experiments were conducted in this study to assess the influence of different training configurations in classification performance. Each experiment was conducted twice to account for randomness during training, and with 10-fold cross validation if no separate tests set were defined (see Table I). *D128* and *D512* were trained for 500 epochs using Adam optimizer [25] with a learning rate of 0.001 and a batch size of 256. The training set was kept unbalanced applying class weights during training as reported to be the best method for *T1* in [19]. The test data was balanced randomly. Average file-wise accuracy over all repetitions and folds was used as evaluation metric.

The first experiment shows the performance of the embeddings compared to the baseline for all classifiers using *late fusion*. For this experiment, 80% of the data was used for training, 10% for validation and 10% for testing if no separate test set exists. Otherwise, 20% of the data was used for validation. All splits were performed on an audio file level to avoid having the same file in different sets. The baseline results may differ slightly from the originally reported since everything was retrained to obtain comparable results with the same processing pipeline.

The second experiment evaluates *early* and *late fusion* strategies to obtain file-wise classification accuracy. The same data distribution as in the first experiment is used here.

The third experiment analyzes the impact of the training domain using *late fusion* for all tasks by comparing **OpenL3** music and environmental embeddings. The influence of the embedding size was also evaluated using 512 and 6144 output values from **OpenL3**. Finally, the training data was reduced to 10% showing the implications of having a smaller annotated dataset for training a classifier.

IV. RESULTS

A. Embeddings & Classifiers

Over all tasks, an average accuracy of 91.2% was achieved with the baseline models in the first experiment (see Table II).

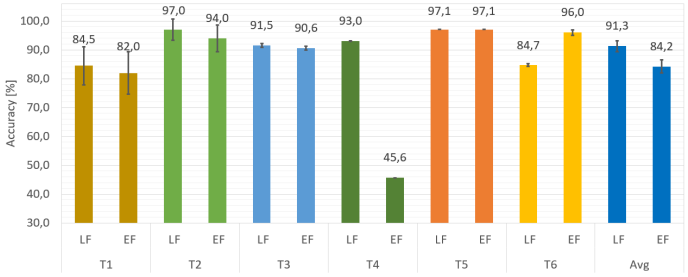

Fig. 1. Mean accuracy and standard deviation of late fusion (LF) and early fusion (EF) over all folds per task. Results averaged over all tasks are also shown (Avg).

While **VGGish** performs best on *T2*, the **OpenL3** embeddings achieved competitive results compared to the baseline over all tasks, in particular for the MIR tasks. While the **OpenL3** results are slightly worse for the ISA tasks compared to the baseline, the **VGGish** embeddings obtain relatively poor results in all ISA tasks. Even though the best performance for the embeddings on the *ESC50* dataset was reported using **Kumar** embeddings,[8] this cannot be observed for any of the evaluated tasks. The results of this experiment suggest that embeddings trained in a self-supervised fashion can be a powerful alternative to fully supervised training with large annotated datasets.

Surprisingly, the linear SVM performed better on average as a classifier than the two fully-connected networks. This is an interesting result since SVMs are fast to train, and easy to incorporate in real-world applications. Since **OpenL3** embeddings with a linear SVM showed the best results in this experiment, all the results reported in the remainder of this paper are reported for this embedding-classifier combination. Detailed figures and values for all combinations can be found online.[3]

B. Early and late fusion

Fig. 1 shows the performance of *early* and *late fusion* approaches for all tasks using **OpenL3** with SVM. *Late fusion* performs better or similar to *early fusion* for all tasks except *T6*. Showing the variability of each class at every time frame to the classifier seems to be a better choice when the classes are present in the majority of the recordings. For tasks such as *T6* where the class is only audible at certain time frames, it is beneficial to fuse the features for each file. The *early fusion* approach outperforms the baseline for *T6* indicating that the choice of fusion approach should be task-specific.

C. Trained Domain

In line with the findings in [8], the **OpenL3** embeddings trained on the music domain performed best (see Fig. 2). This shows that the initial training domain must not necessarily fit to the final task. The final size of the embeddings does not appear to affect the performance of the classifier, hinting that

[8]See results reported at https://github.com/karolpiczak/ESC-50

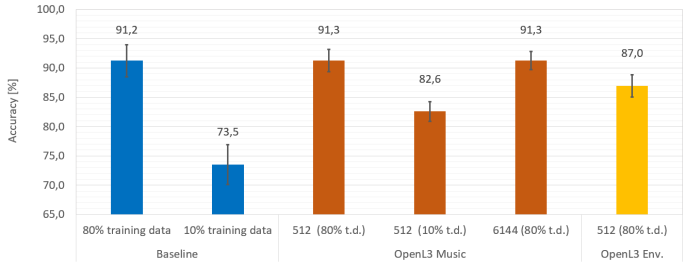

Fig. 2. Mean accuracy and standard deviation for all tasks using OpenL3 embeddings: initial training domain (music vs environmental), embedding size (512 vs 6144), and the amount of data used for training (80% vs 10%).

the different output sizes between the evaluated embeddings are not purely responsible for their performance.

Finally, reducing the number of training examples to 10% emphasizes the importance of pre-trained networks for small datasets. While performance is comparable when using the entire dataset (baseline: 91.2% vs OpenL3: 91.3%), the **OpenL3** embeddings clearly outperform the baseline when fewer training examples are available (baseline: 73.5% vs OpenL3: 82.6%).

V. Conclusions

From the evaluated embeddings, **OpenL3** performed best over all tasks and was on par with the baseline using models trained from scratch. This shows that self-supervised embeddings such as **OpenL3** are a promising alternative to pre-training models on large annotated datasets for getting descriptive features. Lowering the amount of training examples changed the performance in favor of pre-trained embeddings. Therefore, embeddings seem to be a good starting point for novel tasks with small datasets, and important for future research. In general, linear SVMs trained on embeddings performed better than non-linear dense models, making them a feasible choice for real-world applications. Even though the initial training domain does not necessarily need to fit the target domain, there was more room for improvement on ISA tasks compared to the baseline and MIR tasks. This suggests that noise-like signal are not fully covered in the evaluated embeddings. As future work, retraining one or several layers as well as combining the intermediate activations as reported in [9], [13], [18] can be a possible extension. Furthermore, the impact of capturing embedding changes over time is another interesting direction.

References

[1] 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*, New Orleans, USA, 2017, pp. 776–780.

[2] S. Grollmisch, J. Abeßer, J. Liebetrau, and H. Lukashevich, "Sounding Industry: Challenges and Datasets for Industrial Sound Analysis," in *EUSIPCO*, A Coruña, Spain, 2019.

[3] 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.

[4] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks," in *IEEE CVPR*, 2014, pp. 1717–1724.

[5] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified Language Model Pre-training for Natural Language Understanding and Generation," in *NeurIPS*, 2019.

[6] R. Arandjelovic and A. Zisserman, "Look, Listen and Learn," in *IEEE ICCV*, 2017, pp. 609–617.

[7] A. Kumar, M. Khadkevich, and C. Fugen, "Knowledge Transfer from Weakly Labeled Audio Using Convolutional Neural Network for Sound Events and Scenes," in *IEEE ICASSP*, 2018, pp. 326–330.

[8] J. Cramer, H.-h. Wu, J. Salamon, and J. P. Bello, "Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings," in *IEEE ICASSP*, Brighton, United Kingdom, 2019, pp. 3852–3856.

[9] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," *arXiv:1703.09179*, 2017.

[10] J. S. Gómez, J. Abeßer, and E. Cano, "Jazz Solo Instrument Classification With Convolutional Neural Networks, Source Separation, and Transfer Learning," in *ISMIR*, Paris, France, 2018.

[11] G. Gwardys and D. Grzywczak, "Deep Image Features in Music Information Retrieval," *Int. Journal of Electronics and Telecommunications*, vol. 60, no. 4, pp. 321–326, 2014.

[12] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Computer Science*, vol. 112, pp. 2048–2056, 2017.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Int. Conf. on Learning Representations (ICLR)*, San Diego, USA, 2015.

[14] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in *ISMIR*, 2011.

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

[16] S. Abu-El-Haija, N. Kothari, J. Lee, A. P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark," *arXiv:1609.08675*, 2016.

[17] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *23rd Annual ACM Conf. on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[18] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning Sound Representations from Unlabeled Video," *Advances in Neural Information Processing Systems*, pp. 892–900, 2016.

[19] S. Grollmisch, E. Cano, F. Mora-Ángel, and G. López Gil, "Ensemble size classification in Colombian Andean string music recordings," in *Int. Symposium of Computer Music Multidisciplinary Research (CMMR)*, Marseille, France, 2019.

[20] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, M. Müller, and H. Lukashevich, "Investigating CNN-Based Instrument Family Recognition for Western Classical Music Recordings," in *ISMIR*, Delft, The Netherlands, 2019.

[21] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, 2016.

[22] M. Marolt, C. Bohak, A. Kavčič, and M. Pesek, "Automatic Segmentation of Ethnomusicological Field Recordings," *Journal of Applied Science*, vol. 9, 2019.

[23] S. Grollmisch, D. Johnson, T. Krüger, and J. Liebetrau, "Plastic Material Classification using Neural Network based Audio Signal Analysis," in *Sensor and Measurement Science International (SMSI)*, Nürnberg, 2020.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Int. Conf. on Learning Representations (ICLR)*, San Diego, USA, 2015.