Evaluation of CNN-based Automatic Music Tagging Models

Minz Won **Andres Ferraro** **Dmitry Bogdanov** **Xavier Serra**
Music Technology Group
Universitat Pompeu Fabra, Barcelona
{firstname.lastname}@upf.edu

ABSTRACT

Recent advances in deep learning accelerated the development of content-based automatic music tagging systems. Music information retrieval (MIR) researchers proposed various architecture designs, mainly based on convolutional neural networks (CNNs), that achieve state-of-the-art results in this multi-label binary classification task. However, due to the differences in experimental setups followed by researchers, such as using different dataset splits and software versions for evaluation, it is difficult to compare the proposed architectures directly with each other. To facilitate further research, in this paper we conduct a consistent evaluation of different music tagging models on three datasets (MagnaTagATune, Million Song Dataset, and MTG-Jamendo) and provide reference results using common evaluation metrics (ROC-AUC and PR-AUC). Furthermore, all the models are evaluated with perturbed inputs to investigate the generalization capabilities concerning time stretch, pitch shift, dynamic range compression, and addition of white noise. For reproducibility, we provide the PyTorch implementations with the pre-trained models.

1. INTRODUCTION

Automatic music tagging is a multi-label binary classification task that aims to predict relevant tags for a given song. Typically these tags carry useful semantic music information that can later be used for other applications such as music recommendation or music retrieval. To tackle the problem of music tagging, recent studies in music information retrieval (MIR) adopted deep neural networks, mostly based on convolutional neural networks (CNNs), motivated by their huge success in other domains (e.g., computer vision, natural language processing). The introduction of deep learning helped to break the previous glass ceiling in the performance of music tagging systems and MIR researchers started actively proposing ingenious architecture designs. As a result, the hand-crafted feature-based approaches were replaced by data-driven feature learning approaches in most recent automatic music tagging research.

To maximize the advantages of CNNs, a deep fully convolutional network (FCN) was proposed for music tagging [1]. It uses a stack of 3×3 rectangular filters followed by a max-pooling layer. As an alternative, the Musicnn [2], also a Mel-spectrogram-based CNN, tried to incorporate domain knowledge into its filter designs so that the model can capture timbral characteristics and temporal patterns using vertical filters and horizontal filters, respectively. Sample-level CNN [3] pursued an assumption-free end-to-end model by applying 1D convolution directly to a raw audio waveform, and the following research [4] improved the performance by adding squeeze-and-excitation blocks [5]. Different from images, however, music is sequential. For this reason, a convolutional recurrent neural network (CRNN) [6] was proposed to extract local features using CNNs and summarize them with recurrent neural networks (RNNs). Another sequence modeling approach [7] adapted the self-attention mechanism [8] to summarize the temporal sequence of the extracted local features by CNNs. Finally, Harmonic CNN [9] used a harmonically stacked trainable representation to preserve spectro-temporal locality in convolution layers.

Unfortunately, due to different experimental setups followed by the authors of these approaches when reporting results (e.g., dataset splits, library versions, computing environments, and optimization methods), it is difficult to compare the proposed architectures directly with each other. In this paper, we address this issue and report experimental results for various state-of-the-art music tagging models using three different datasets (MagnaTagATune, Million Song Dataset, and MTG-Jamendo dataset) with a consistent experimental setup. In addition, we conduct experiments to assess the robustness of these architectures against four different types of deformations [10] and determine their generalization abilities. For the reproducibility, we provide PyTorch implementations for all the models considered and their pre-trained models. [1]

The paper is organized as follows. Section 2 describes automatic music tagging tasks with details of multiple instance problem, evaluation metrics, and dataset descriptions. Section 3 elaborates on the model designs. We report model performances and their generalization capabilities in Section 4 and Section 5, respectively. Finally, Section 6 concludes the paper.

[1] https://github.com/minzwon/sota-music-tagging-models/

2. AUTOMATIC MUSIC TAGGING

2.1 Multiple Instance Problem

Various semantic information characterizing music, such as genres, subgenres, moods, instruments, decades, and languages, can be expressed in the form of music tags. Automatic content-based music tagging is a task that aims to predict such relevant tags for a given song based on its acoustic characteristics. However, depending on the tag, relevant characteristics are not necessarily predominant in the entire music recording. For example, when a song has a tag *female vocal*, it does not imply that the female vocal appears in every time segment of the song. In essence, this is a multiple instance problem [11]. A song can have many acoustic characteristics (instances) that describe it, but often only specific characteristics (instances) are responsible for the associated music tag. In most cases, we do not have time precise instance-level annotations because the precise labeling can be laborsome, therefore a music tag associated with a song is simply applied to all music excerpts (instances) of the song during training [12].

There are two approaches to handle the multiple instance problem in music tagging. One is to train the model on full songs and produce song-level predictions from a song-level input. From a given song-level input, the model has to predict relevant tags. Another one is to train the model on short audio chunks (instances) and generate chunk-level predictions, which can be later aggregated (e.g., majority vote, average) in the evaluation phase. In this paper, we refer to these two approaches as song-level training and chunk-level training, respectively.

2.2 Evaluation Metrics

A common evaluation metric of binary decision problems is the area under receiver operating characteristic curve (ROC-AUC). However, the area under precision-recall curve (PR-AUC) can be more informative for evaluation on highly skewed datasets [13]. Hence, we use both macro ROC-AUC and PR-AUC to evaluate all considered music tagging models with both scores being averaged across all the tags the models operate with. For our robustness studies, we report the variance of ROC-AUC and PR-AUC scores obtained on perturbed audio inputs.

2.3 Datasets

MagnaTagATune (MTAT) [14] is one of the most commonly used datasets for benchmarking automatic music tagging systems. It contains multi-label annotations by genre, mood, and instrumentation for 25,877 audio segments, each 30s long. The audio is in the MP3 format (32 Kbps bitrate and 16 kHz sample rate). Originally the dataset is split into 16 folders, and commonly the first 12 folders are used for training, the 13th for validation, and the last three are used for testing. Only 50 most frequent tags are typically used for the task. The top 50 tags include genre and instrumentation labels, as well as decades (e.g., '80s' and '90s') and moods.

In our work, we follow the same data split and we use the top 50 tags to be consistent with results reported in the majority of previous studies. [2] However, we have noted that the performances reported in some studies are inconsistent, as their authors discarded audio segments without any associated tags (leading to slightly higher values of performance). Also, some of the previous studies are reusing those reports unintentionally compare incompatible performance values, which is one of the main motivations of our work.

Million Song Dataset (MSD) [15] is a dataset of audio features for one million songs, partially expanded by the MIR community with crowdsourced tags from *Last.fm* as well as a mapping to 30s audio preview segments originally obtained from *7digital*. [3] In total, this subset of the dataset contains 241,904 annotated song segments and it is commonly used for benchmarking on a larger scale. The tags cover genre, instrumentation, moods and decades. The audio segments vary in quality, being encoded as MP3s with a bitrate from 64 to 128 Kbps and sample rate of 22 kHz or 44 kHz.

Again, in our study we follow the dataset split commonly used by researchers [4] [2, 3] and use only 50 most frequent tags for consistency with previous studies. This split includes 201,680 songs for training, 11,774 for validation and 28,435 for testing. Unfortunately, similarly to MTAT, some previous studies report inconsistent comparisons of automatic tagging approaches, as there also exists a slightly different split, [5] containing audio segments of shorter duration and on which lower performance values were reported. Note that the tag annotations available for this dataset are inherently noisy as they come from a free-form social tagging application for music enthusiasts and are used without any preprocessing intended to improve the quality of tags [16].

MTG-Jamendo Dataset [17] contains audio for 55,701 full songs and is built using music publicly available on the *Jamendo* [6] music platform under Creative Commons licenses. The minimum duration of each song is 30s, and they are provided in the MP3 (320 Kbps bitrate). Thus, this dataset contains significantly larger audio segments with higher encoding quality than MTAT and MSD. The tracks in the dataset are annotated by 692 different tags covering genres, instrumentation, moods and themes. All tags were originally provided by the artists submitting music to Jamendo, but they were preprocessed with the goal of tag cleaning by the creators of the dataset.

Multiple splits of the data are provided for training, validation and test. In this work we use the split-0 and 50 most frequent tags. [7] As this dataset has been released only recently, not many studies are reporting the performance of the models using it yet. Yet, it is a useful addition for the evaluation methodologies followed by researchers in order to better assess the generalization of their models.

[2] https://github.com/jongpillee/music_dataset_split/tree/master/MTAT_split

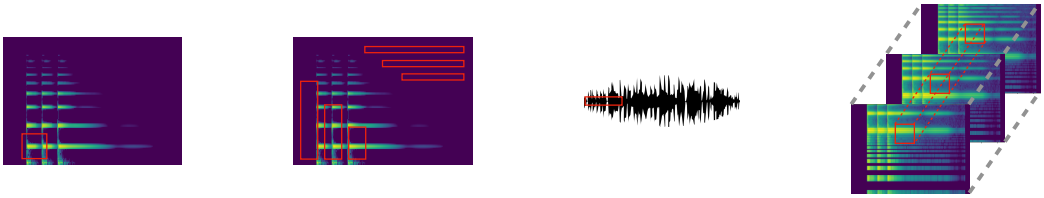
[3] https://www.7digital.com

[4] https://github.com/jongpillee/music_dataset_split

[5] https://github.com/keunwoochoi/MSD_split_for_tagging

[6] https://jamendo.com

[7] https://github.com/MTG/mtg-jamendo-dataset



(a) (b) (c) (d)

Figure 1: Shapes of the first convolution filters and input representations of different models. (a) FCN, CRNN, self-attention, and short-chunk CNN with a Mel spectrogram input (b) Musicnn with a Mel spectrogram input (c) sample-level CNN with a raw audio input (d) Harmonic CNN with a stacked harmonic tensor.

Model	input length	# of Mel	training
FCN	29.1s	96	song-level
FCN (128)*	29.1s	128	song-level
Musicnn	3s	96	chunk-level
Musicnn (128)*	3s	128	chunk-level
Sample-level CNN	3.69s	-	chunk-level
CRNN	29.1s	96	song-level
CRNN (128)*	29.1s	128	song-level
Self-attention	15s	128	chunk-level
Harmonic CNN	5s	128	chunk-level
Short-chunk CNN	3.69s	128	chunk-level

Table 1: Experimental setups of different models.

3. MODELS

We here describe the architectures of all models considered in this study. *Essentia* [18] and *Librosa* [19] libraries are used for loading audio files and extracting Mel spectrograms, respectively. We re-sampled the audio to 16 kHz sample rate. Table 1 shows different input lengths and the number of Mel bands. Since some models have a smaller number of Mel bands in their original implementations, we further experimented with a larger number of Mel bands for the fair comparison (marked with *). All Mel-spectrogram-based approaches used 512-point FFT with a 50% overlap. For training the models, we used a unified optimization method: a mixture of scheduled ADAM [20] and stochastic gradient descent (SGD), introduced in [7]. The best model is selected based on the training loss. In the evaluation, chunk-level models average predictions over 16 chunks to perform the final prediction. Note that FCN and CRNN are song-level training models for MTAT and MSD but they perform as chunk-level training models for the MTG-Jamendo dataset since the songs in the dataset are longer than 29.1s which is the size of receptive fields of both models.

3.1 Fully Convolutional Network

A fully convolutional network (FCN) is a variant of CNN that consists of only convolutional layers without any fully-connected layers. A FCN for music tagging [1] uses Mel spectrogram inputs. In the preprocessing step, a 29.1s audio segment is converted to a 96×1366 Mel spectrogram. It is then used as an input and is passed through 4 convolutional layers. Each convolutional layer uses homogeneous 3×3 2D filters (Figure 1-(a)) followed by a max-pooling layer. Different sizes of strides are used for max-pooling layers ((2, 4), (4, 5), (3, 8), (4, 8)) to increase the size of receptive fields to cover the entire input Mel spectrogram (96×1366). In the original paper, FCN was trained with a song-level training method since the track durations in MTAT correspond to the size of the receptive field. However, this is not the case for MTG-Jamendo containing longer tracks, where chunk-level (29.1s) training is applied.

3.2 Musicnn

The Musicnn [2] model also uses Mel spectrograms as its inputs. Different from previously proposed models, the architecture design choices in Musicnn rely on some intuition from the music domain knowledge. The first convolutional layer of Musicnn consists of vertical and horizontal filters. Vertical filters are designed to capture pitch-invariant timbral features (bottom-left of Figure 1-(b)): e.g., 38×7 filter captures sub-band information of short period of time. To enforce the pitch-invariancy, the following max-pooling layer pools the maximum value across the frequency axis. Horizontal filters, on the other hand, capture temporal the energy envelope of the audio. After the mean-pooling across the frequency axis of input Mel spectrograms, horizontally long filters (e.g., 1×165) capture the temporal energy patterns (top-right of Figure 1-(b)). The extracted timbral and temporal features are concatenated in the channel, then the following 1D convolutional layers summarize them to predict relevant tags. Different from FCN, the Musicnn only uses short audio excerpts (3s) as its inputs during training, i.e., chunk-level training.

3.3 Sample-level CNN

Sample-level CNN [3] tackles the automatic music tagging problem in an end-to-end fashion. It takes raw audio waveforms as its inputs. Sample-level CNN is simpler and deeper than Mel spectrogram-based approaches. It consists of ten 1D convolutional layers with 1×3 filters and 1×3 max-poolings (Figure 1-(c)). Trained front-end filters perform similar to the process of deriving Mel spectrograms and the back-end convolution layers summarize them. We also considered a variation of sample-level CNN [4] with squeeze-and-excitation (SE) [5] blocks. Sample-level CNN and its variant with SE blocks also use short audio excerpts (3.69s) for the chunk-level training.

3.4 Convolutional Recurrent Neural Network

Convolutional recurrent neural network (CRNN) [6] uses Mel spectrogram inputs. A CRNN can be described as a combination of CNNs and RNNs. The CNN front end extracts local features and the RNN back end summarizes them temporally. Since RNNs are more flexible than CNNs for summarizing sequential information, it can be beneficial to use RNNs for predicting tags that may be affected by global structures (e.g., moods/themes). Four convolutional layers with 3×3 2D filters are used in the front end and two-layer RNNs with gated recurrent units (GRU) are used in the back end. Long music excerpts (29.1s) are used as inputs of CRNN.

3.5 Self-attention

The self-attention-based music tagging model [7] shares the same intuition as CRNN to extract local features with CNNs and summarize them with sequence models. The only difference is that the self-attention mechanism is used instead of the RNNs for the temporal summarization back end. Motivated by its huge success in natural language processing [8], the authors adapted the Transformer encoder, which is a deep stack of self-attention layers, for automatic music tagging. 15s-long audio excerpts are used for training the self-attention model.

3.6 Harmonic CNN

Harmonic CNN [9] takes advantage of trainable band-pass filters and harmonically stacked time-frequency representation inputs. Trainable filters (mainly trainable bandwidths) bring more flexibility to the model. And harmonically stacked representation preserves spectro-temporal locality while keeping the harmonic structures through the channel of the input tensor in the first convolution layer (Figure 1-(d)) as introduced in [21]. The number of trainable frequency bands is set to 128 and the number of harmonics considered for stacking is 6. Chunk-level training with 5s audio segments is performed.

3.7 Short-chunk CNN

According to the previous work [9], a simple 2D CNN with 3×3 filters can already claim exceptional results when it is trained with short chunks of audio, i.e., chunk-level training. It is a very prevalent type of CNN (sometimes referred as *vgg*-like) but, to the best of our knowledge, there are no references for this architecture design in music tagging research. Hence, we implemented a 7-layer CNN with a fully-connected layer and its extension with residual connections [22]. Different from FCN, it uses a smaller size of max-pooling (2×2) because the input segment is way shorter than the song-level inputs (29.1s). We used 128 Mel bins so that 7 max-pooling layers can summarize them into a single dimension ($2^7 = 128$). It uses 3.69s audio excerpts, hence we call this model "short-chunk CNN" in this paper to differentiate it from FCN.

4. RESULTS

We report ROC-AUC and PR-AUC of all implemented models using three datasets in Table 2. In general, models trained with short audio excerpts (Musicnn, variants of sample-level CNN, Self-attention, Harmonic CNN, variants of short-chunk CNN) outperform other models trained with relatively longer audio segments (FCN, CRNN). Training with short chunks (instances) is noisier: e.g., an audio excerpt can have a tag *guitar* if a guitar appears in the song even though the selected excerpt doesn't include guitar sound in it. However, one can expect a much larger number of examples during the training (e.g., 25,877 tracks $\times$ 16 chunks = 414,032 examples). We suspect this brings the performance gain when the model is trained with short chunk-level examples. Furthermore, most of the top 50 tags in the three datasets can be identified only with a short audio excerpt (e.g., instruments, genres). Thus, the model does not need a long sequence of audio to perform its binary classification task. For the top 50 tags in each dataset we experimented with, it is more beneficial to use chunk-level training with short audio excerpts than the song-level training.

Short-chunk CNN, short-chunk CNN with residual connections, and Harmonic CNN showed the best results for every dataset. These three models are trained on short audio excerpts (3.69s or 5s) and they use 3×3 convolutional filters followed by 2×2 max-poolings. FCN uses similar filters, but larger max-poolings which increase its size of the receptive field to fit long audio segments (29.1s). We conclude that smaller max-poolings with shorter audio excerpts work better for CNNs with 3×3 filters.

Musicnn shows competitive results in MTAT. However, other models (sample-level + SE, self-attention) outperform Musicnn on larger datasets (MSD and MTG-Jamendo). This confirms an intuition that domain knowledge can be beneficial for relatively small datasets, reported in [2]. However, the design choices for parameters of Musicnn restricts the power of the model when it is trained on larger datasets.

For the sequential models, self-attention outperforms the CRNN. Different from self-attention mechanisms, RNNs with long sequence inputs suffer from vanishing gradient problems. Self-attention mechanism alleviates the problems by providing direct paths between all time steps. According to the reported visualizations [7], self-attention performs well for pinpointing relevant short-time acoustic features in the audio sequence, but it was difficult to determine if the model learned long-time characteristics properly. To determine such abilities, some tags related to a global structure have to be cherry-picked and evaluated.

Since FCN, Musicnn, and CRNN use Mel spectrogram inputs with 96 Mel bands, there can be relative disadvantages when they are compared with other models using 128 Mel bands. For the fair comparison, we experimented with FCN, Musicnn, and CRNN using 128 Mel bands. A larger number of Mel bands did not show any significant impacts on the performances. Since each architecture design was optimized for a smaller number of Mel bands, simply increasing the size of input Mel bands cannot guarantee the optimized performance of the models.

Methods	MTAT		MSD		MTG-Jamendo	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
FCN [1]	0.9005	0.4295	0.8744	0.2970	0.8255	0.2801
FCN (with 128 Mel bins)	0.8994	0.4236	-	-	-	-
Musicnn [2]	0.9106	0.4493	0.8803	0.2983	0.8226	0.2713
Musicnn (with 128 Mel bins)	0.9092	0.4546	-	-	-	-
Sample-level [3]	0.9058	0.4422	0.8789	0.2959	0.8208	0.2742
Sample-level + SE [4]	0.9103	0.4520	0.8838	0.3109	0.8233	0.2784
CRNN [6]	0.8722	0.3625	0.8499	0.2469	0.7978	0.2358
CRNN (with 128 Mel bins)	0.8703	0.3601	-	-	-	-
Self-attention [7]	0.9077	0.4445	0.8810	0.3103	0.8261	0.2883
Harmonic CNN [9]	0.9127	0.4611	**0.8898**	**0.3298**	0.8322	0.2956
Short-chunk CNN	0.9126	0.4590	0.8883	0.3251	**0.8324**	**0.2976**
Short-chunk CNN + Res	**0.9129**	**0.4614**	**0.8898**	0.3280	0.8316	0.2951

Table 2: Performances of state-of-the-art models.

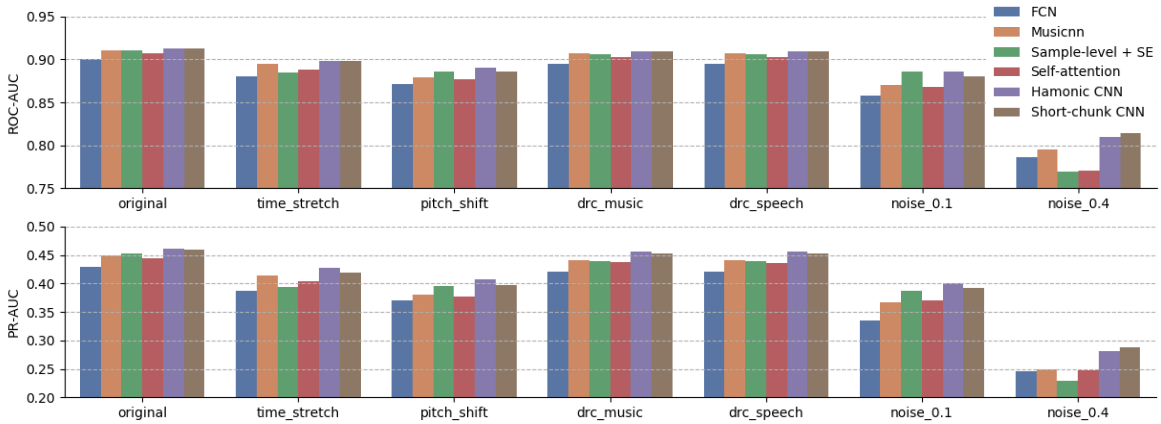


Figure 2: Evaluations metrics with perturbed audio inputs. Dynamic range compression is shortened as "drc" in the plot.

5. ROBUSTNESS STUDIES

5.1 Input Deformations

To further investigate the performance of different state-of-the-art models, we conducted robustness studies. If a pre-trained model has good generalization abilities, the prediction of the model should not be sensitive against small perturbations in the input audio. By applying four different audio deformations to the test set (pitch shift, time stretch, dynamic range compression, and addition of white noise), we intended to determine the generalization abilities of the models. Note that we applied these four deformations only to the test set, which means that the models have never been exposed to the same deformations during training. All employed deformations are based on an existing music data augmentation framework (MUDA) [8] [10]:

Pitch shift by $n \in \{-1, 1\}$ semitones.

Time stretch by $\gamma \in \{2^{-1/2}, 2^{1/2}\}$.

Dynamic range compression following *speech* and *music (standard)* settings of Dolby E standards [23].

White noise addition $x_{mixed} = (1 - \alpha) \cdot x + \alpha \cdot x_{noise}$ where $\alpha \in \{0.1, 0.4\}$.

5.2 Robustness results

Figure 2 shows performances of each model under various input deformations. Here we tested FCN, Musicnn, sample-level + SE, self-attention, Harmonic CNN, and short-chunk CNN. We followed the original input preprocessing of each model because a larger of Mel bands did not show significant effects in Section 4. CRNN is not included due to its relatively low performance. Dynamic range compression was the least influential and the white noise addition (0.4) was the most critical among the four different perturbations considered. Musicnn is robust against time stretching but it is relatively vulnerable against pitch shift. We suspect the max pooling layer over frequency axis hinders the Musicnn from learning generalized representations. Harmonic CNN and short-chunk CNN were the two best models with original data. However, Harmonic CNN showed better generalization abilities against input deformations except for the white noise addition (0.4). Sample-level CNN with SE blocks showed good performance with a small amount of noise (0.1), but it could not generalize when this amount was increased (0.4).

[8] https://github.com/bmcfee/muda

6. CONCLUSION

In this paper, we revisit state-of-the-art automatic music tagging models and report their performances with a consistent experimental setup. In general, short-chunk-based approaches showed better results than models trained with larger input segments (FCN, CRNN). The design choices followed by Musicnn could show good performance on a small dataset, but it restricted the model from learning more information on larger datasets. Sequential models (CRNN, self-attention) showed competitive results but could not outperform other models since most of tags in the datasets do not require long sequences for their identification. Interestingly, the best performing model is a simple CNN with 3×3 filters trained on short audio excerpts (short-chunk CNN). Although the original design choice of the CNN is from computer vision, it outperformed other methods except for Harmonic CNN.

We further assessed generalization abilities of models by testing perturbed inputs. We could observe a different ranking of the models in terms of their performance on each deformation. This implies that the ROC-AUC and PR-AUC scores are not enough to evaluate music tagging models. In our experiment, Harmonic CNN and short-chunk CNN consistently report better scores than other models. Specifically, Harmonic CNN showed the best generalization abilities against every deformation types except for a heavy white noise addition.

We expect the reported results to be a useful reference for further research in automatic music tagging. More detailed exploration of the deformations for robustness tests should be done in the future work. Also, the efficacy of data augmentation in music tagging has to be further investigated.

Acknowledgments

This work was funded by the predoctoral grant MDM-2015-0502-17-2 from the Spanish Ministry of Economy and Competitiveness linked to the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502). Also we acknowledge this research has been supported by Kakao Corp.

7. REFERENCES

[1] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv preprint arXiv:1606.00298*, 2016.

[2] J. Pons Puig, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018; 2018 Sep 23-27; Paris, France. p. 637-44.* International Society for Music Information Retrieval (ISMIR), 2018.

[3] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *arXiv preprint arXiv:1703.01789*, 2017.

[4] T. Kim, J. Lee, and J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 366–370.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[6] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.

[7] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," *arXiv preprint arXiv:1906.04972*, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] M. Won, S. Chun, , O. Nieto, and X. Serra, "Data-driven harmonic filters for audio representation learning," *2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2020.

[10] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation." in *ISMIR*, vol. 2015, 2015, pp. 248–254.

[11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[12] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[13] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[14] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging." in *ISMIR*, 2009, pp. 387–392.

[15] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," 2011.

[16] K. Choi, G. Fazekas, K. Cho, and M. Sandler, "The effects of noisy labels on deep convolutional neural networks for music classification," *arXiv preprint arXiv:1706.02361*, 2017.

[17] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," 2019.

[18] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, J. Salamon, J. R. Zapata González, X. Serra *et al.*, "Essentia: An audio analysis library for music information retrieval," in *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.* International Society for Music Information Retrieval (ISMIR), 2013.

[19] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[21] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music." in *ISMIR*, 2017, pp. 63–70.

[22] 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*, 2016, pp. 770–778.

[23] E. Dolby, "Standards and practices for authoring dolby digital and dolby e bitstreams," *Dolby Labortories, Inc*, 2002.