194.039 VU Intelligent Audio and Music Analysis WS 2018/19 – Exercise 3

Final submission deadline: February 29, 2020, 23:59.
Contact: peter.knees@tuwien.ac.at (Part A), schindler@ifs.tuwien.ac.at (Part B)

For both parts, follow the code structure from Assignments 1 and 2 and reuse as much code as possible. Again, use Python (or IPython/Jupyter Notebook), pyTorch, and madmom or librosa. Submit a zip file containing your solutions to both Part A and Part B via TUWEL. Do not include the audio datasets in your submission but rather use a variable that can be easily set to point to a local copy of the data.

Part A - Semantic Music Tagging (50%)

Following the approach by Choi et al. [2016] "Automatic Tagging Using Deep Convolutional Neural Networks" (https://arxiv.org/pdf/1606.00298.pdf), a CNN architecture shall be implemented to predict semantic categories for snippets of music pieces. As suggested, for experiments, the MagnaTagATune dataset will be used.

1. From the MagnaTagATune dataset Website, download the *Clip metadata*, the *Tag annotations*, and the *Audio data* files.
2. Prepare the data in a way that allows to train the network and test performance, e.g., by generating a training/validation(dev)/test split.
   Important: Use the information given in the Clip metadata file to ensure that *no clips extracted from the same track end up in different splits*.
3. Filter the tags such that only the 50 most frequent tags in the dataset remain.
4. Implement the approach outlined by Choi et al. [2016] based on Mel-spectrograms. You can find methods for transformations of spectrograms to the Mel-scale in both librosa and madmom. Note that the FCN-4 architecture described in the paper lacks implementation details. Instead follow the MusicTaggerCNN KERAS reference implementation by the same authors to reproduce the architecture using pyTorch.
5. Train the network using ADAM optimization and binary cross-entropy as loss function. Consider reducing the amount of data in a reasonable way if necessary.
6. Use the test set to estimate performance of the trained network using AUC (implemented in sklearn).
7. Compare your results to the numbers reported by Choi et al. [2016] and comment on you main findings.

Part B - Auditory Scene Classification (50%)

Your task is to implement a solution to an auditory scene detection challenge, precisely the DCASE 2016 Acoustic Scene Classification task. Details about the challenge are provided on the task website.

- You are free in the strategy that you apply here and can also reuse and modify your implementation of Part A, e.g., by modifying the architecture to handle clips of length 30 secs.
- Follow the given evaluation strategies of the task, in particular wrt. development and evaluation datasets and cross validation settings.
- Consider reducing the amount of data in a reasonable way if necessary.
- Compare your results to the numbers reported on the task website and comment on you main findings.

Remark: The goal is not to outperform the state of the art, but to experiment with a classification task in the general audio domain. Therefore you can apply your existing solutions from the music domain and reflect upon the capabilities and limitations of your approach.