MIREX 2015 MUSIC/SPEECH CLASSIFICATION

Jimena Royo-Letelier **Romain Hennequin** **Manuel Moussallam**
Deezer
10 rue d'Athenes
75009 Paris, France
research@deezer.com

ABSTRACT

We give a short overview of the algorithm we proposed for the MIREX 2015 music/speech classification contest. The system is based on a deep convolutional neural network with Constant Q Transform spectrograms (CQT-grams) as input features.

Keywords: MIREX 2015, Music/Speech Classification, Music Information Retrieval, Deep networks.

1. ALGORITHM

Each unknown recording is partitioned in non-overlapping 5 second long samples. From each sample a CQT-gram is computed as explained in Section 2. These are used as input features to a classifier consisting in a convolutional neural network (see [2]), with a logistic regression at the top which yields a probability for each of the classes (speech and music). The mean probability of each class is computed over all the samples, and the recording is then assigned to the class with the highest mean probability.

2. FEATURES EXTRACTION

The input features are CQT-grams [3] computed with a 23 ms hop size, 6 octaves, 24 bins per octave and a minimum frequency of 80 Hz. The computation of CQT-grams is done using YAAFE [4]. In order to reduce the dynamics of the features, we compress it using $f(s) = \log_2(1 + C \cdot s)$, where C is a positive constant.

3. CLASSIFIER DETAILS AND TRAINING

Our classifier consists in a convolutional neural network with three convolutional and max-pooling layers followed by two fully-connected layers and a logistic regression at the end as illustrated in Figure 1. The first convolutional layers are aimed at recognizing speech/music patterns present in the CQT-grams. We expect the next layers to capture higher level attributes of audio recordings, yielding at the last layer a representation of the input data suitable for the classification with the logistic regression. We use Theano Python library [1] to build our classifier. This was pre-trained over a database independent from the contest's one (see Section 4), using a mini-batch stochastic gradient descent process over the negative log-likelihood of the input data. We decided not to train our classifier using the contest database since the training phase requires quite intensive computations that must be performed with a GPU in order to keep computation time reasonable.

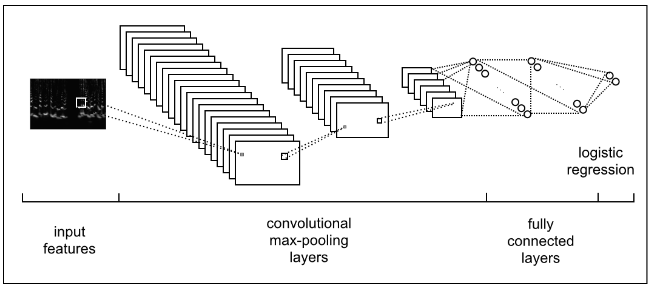

Figure 1. CNN classifier architecture.

4. DATABASE

We used a database consisting on about 130000 speech and music recordings equally distributed between the two classes. The database comes from the Deezer catalog and was annotated using metadata from the audio providers. In order to improve generalization of our system from our database to the contest's one, we enhance the database by applying a range of class-preserving audio transformations, such as noise addition, pitch shifting, equalization, etc. This permits to improve the diversity of our training database, which may be crucial for the classification of new unknown audio recordings.

5. REFERENCES

[1] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In Stéfan van der Walt and Jarrod Millman,

editors, *Proceedings of the 9th Python in Science Conference*, pages 3 – 10, 2010.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[3] J. C. Brown. Calculation of a constant Q spectral transform *Journal of the Acoustical Society of America* vol. 89, no. 1, pp. 425434, January 1991.

[4] B.Mathieu, S.Essid, T.Fillon, J.Prado, G.Richard YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software In *proceedings of the 11th ISMIR conference* , Utrecht, Netherlands, 2010.