Towards Automatic Cover Song Detection with Parallel Convolutional Neural Networks

IEEE WNYISPW 2016 - Marko Stamenovic - November 18, 2016

Introduction

- A cover song, by definition, is a new performance or recording of a previously recorded, commercially released song. It may be by the original artist themselves or a different artist altogether and can vary from the original in unpredictable ways including key, arrangement, instrumentation, timbre and more.

- We propose a neural architecture to learn a relationship ***S***, feature extraction functions f_1 and f_2 and a comparison function ***g***, between two query songs ***A*** and ***B*** to classify them as either cover or non cover song pair such that:

$$V_A, V_B = f_1(A), f_2(B)$$
$$S = g\,(V_A, V_B)$$

Data Preprocessing

Dataset

- Our raw data consists of 64 kbps 22.5 kHz MP3's scraped from 7-Digital preview clips of songs in the Second Hand Song Dataset (SHS), which is a subset of the Million Song Dataset (MSD).
- The SHS contains a training set of 12,960 unique songs, divided into 4,128 cliques, or version-groups of the same original song.
- We use a frequency resolution of one half-tone per frequency bin spanning 7 octaves from C1 (32Hz) to B7 (3951 Hz). 25,000 unique cover song pairs.

Feature Extraction

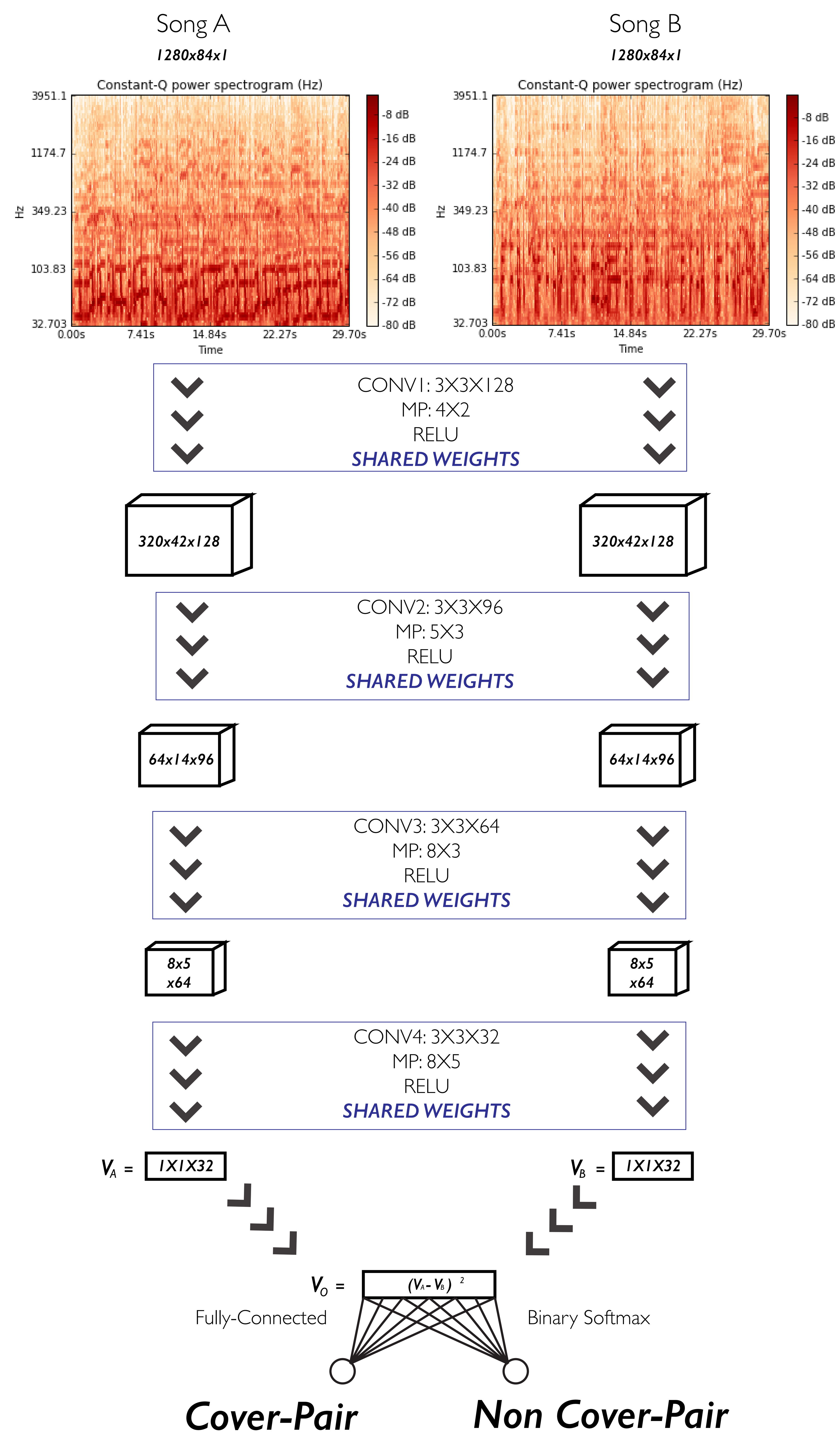
- We use a time-frequency log spectral representation of the audio based on the Constant Q Transform (CQT).
- The CQT is a transform with a logarithmic frequency resolution, mirroring the Western music scale and the human auditory perception of music.
- We use a time resolution corresponding to approximately 0.23 seconds per time frame.
- We use a frequency resolution of one half-tone per frequency bin spanning 7 octaves from C1 (32Hz) to B7 (3951 Hz).

Objective Function

- Our objective function is a binary cross-entropy loss ***h(P,Q)*** on the softmax of the affine transformation ***Q*** of the pointwise squared difference V_o of the extracted feature vectors V_A, V_B and the true labels ***P*** where:

$$V_o = (V_A - V_B)^2$$
$$Q = softmax(\mathbf{W}V_o + B)$$
$$h\,(P,Q) = -\sum P \log(Q)$$

Network Architecture



Evaluation

Filter Layout	Precision
128, 96, 64, 32	65.0%
96, 64, 32, 24	61.9%
8, 8, 16, 24	60.0%
8, 12, 32, 64	58.0%
128, 256, 128, 64	52.5%

Table 1. System precision for various network configurations

- The system was evaluated on a test set of 1,000 held out cover song pairs on five different architectures as shown above. Precision as reported above refers to the amount of correct predictions divided by the entire test set.

- In each case, the network was trained using Tensorflow with a Nvidia Titan Black GPU for 5 epochs and a minibatch size of 16 using the ADAM optimizer to update weights.

- To prevent overfitting, a dropout factor of 0.5 was used on the fully connected layers in addition to an l2 regularization lambda factor of 0.005.

- The network was easily prone to overfitting with additional filters added above the mo, as shown in the results.

Conclusion and Future Work

- We show that 1) it is possible to train a neural network to predict binary classification of cover song pairs, 2) provide an novel network for evaluating cover song pairs and 3) provide code to reproduce our experiments at ***https://github.com/markostam/coversongs-dual-convnet***.

- There are many avenues for future work including testing different configurations of the network including: different optimizers, depths, filter shapes, loss functions and configurations.

- We would also like to test a 3-legged convolutional architecture in which each leg is a separate pipeline for an original song, cover, and non-cover in order to drastically increase the training set size and thus combat overfitting.

Acknowledgements

The author would like to thank Brian Lee for his insightful conversations, Colin Raffel for his help tracking down the Second Hand Songs data, the Recurse Center for providing a supportive working environment and the University of Rochester Dept. of Electrical Engineering.