Nathan Villicaña-Shaw

November 18th, 2014

Final Proposal for Final Project

**Overview :**

Dexter and I are working together to create a program that analyzes audio files for percussive sounds. It extracts labeled samples from the given .wav file automatically. The program has five steps in its implementation. Preprocessing, onset detection, sample extraction, classification, and exportation.

**Specifics :**

A few things happen during the preprocessing stage. After the program checks that the file passed into the program is a .wav file. The first preprocessing step reduces the signal to mono. We check the signal for DC bias, if any is present it is removed. Lastly, we half-wave rectify the signal and then normalize it. We will experiment with down sampling the signal but at the moment we keep it at its native sample rate.

After the preprocessing is taken care of we check for onsets in the file. This is one of the weak links in our code as of the writing of this proposal. At the moment we are using a simple local max function to find the peaks but it is far too sensitive and really not the most effect algorithm for finding onsets, we will experiment with different options. We want the onset portion of the code to be on the sensitive side, as any false positives will be thrown out in the classification stage. We don’t want so many false positives that it cuts our samples short, which is the problem in our code as at the moment. The onset function returns a list that contains flags that represent onsets which are used to extract samples from the master file.

Based on the returned list of flagged onsets we extract samples for analysis. We will want the samples to be on the larger size to ensure that we don’t cut off any important sonic material. We start the sample from 100-300ms before the onset is detected and either go up to the next onset or 2 seconds which ever comes first. There is a chance the samples will overlap but this is fine. A list of arrays are returned where each array contains a sample.

Now that we have our sampled data we initiate our classification section. The purpose of this code is to determine if the sample is percussive in nature or tonal. If the sample is percussive the algorithm then determines if it falls into any of its known categories, we are shooting for Kick Drum and Snare first but are planning to expand the code to claps, high hats, cymbals and other common sample categories. Although it might change, we are planning on using HFC, one-band RMS, three-Band RMS, crest factor, temporal centroid, spectral temproid, spectral kurtosis, spectral skewness, spectral roll off, spectral flux and zero crossing rate for our features. We remove DC bias from our samples, normalize them and lastly they are saved with names corresponding to their master file and sample type. If the algorithm can’t determine the category of the percussive sound but knows the sample is not tonal the program will export the sample as unknown. This way we can go and manually label the samples for use as further training data and feature tweaking.

**Progress/Timeline :**

In my machine learning homework I implemented SVM for my supervised learning and figured I would expand upon that code. Not wanting to reinvent the wheel I decided to do some research before settling on my features and firm implementation. I was lucky enough to stumble on quite a few papers written on drum recognition and extraction. Using their workflow as a guide our code is based on a conglomeration of these papers. I am not worried about the project, it all seems somewhat strait forward, just lots of tweaking.