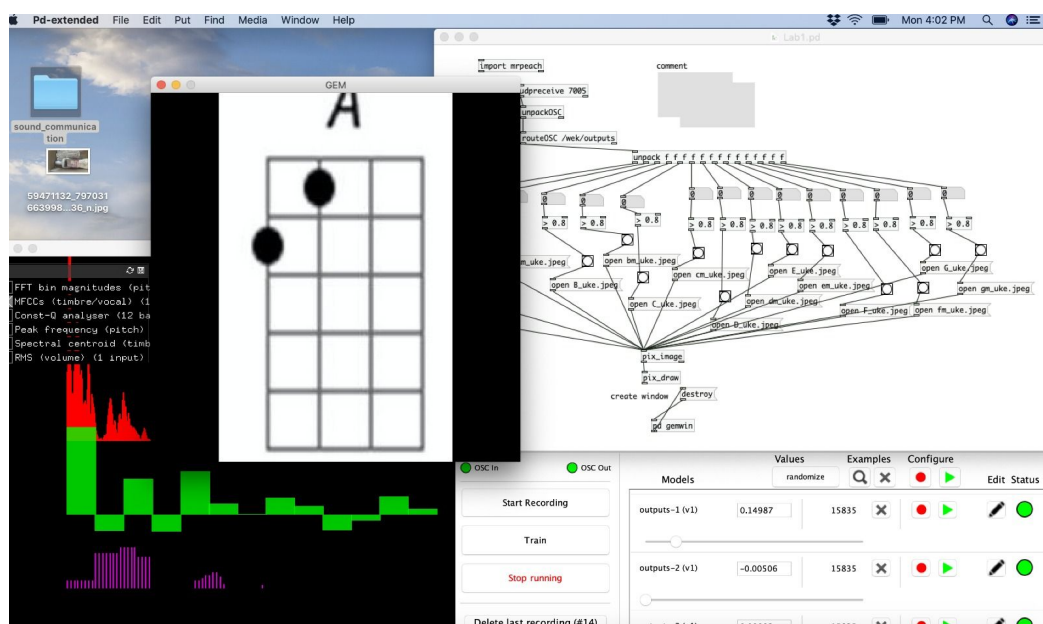


In this first assignment, Various_Audio_Inputs has been used to take 13 channels MFCC information to train ANN models in Wekinator to recognize various chords played on ukulele. The models trained in Wekinator are utilized by a Puredata patch which links the trained models to images of ukulele chords. The intention has been to create an application that could be utilized in music lessons or by amateur musicians to facilitate the learning of ukulele chord tabs with a more fluid interface.

The ultimate goal and use for this particular type of application would be as follows: A teacher/player strums a chord on ukulele, and chord tab is displayed on the screen of the monitor responding in real time to the chord being played. This would prevent music teachers from having to stopping the flow of a demonstration to remind a student of the chord as they play together, and improves the students experience by allowing the student to play along and master chords more easily and in time with the teacher/other players.

After Various_Audio_Inputs is used to record and train models, it is utilized by the Pd patch to listen through the computer's microphone in real time. The models trained by wekinator output a "score" in real time of which chord is most likely being played. When one of these scores goes beyond the threshold ($> .8$ for example), a bang is triggered that opens the file for the appropriate chord chart. A new chord chart is only loaded in the window when another bang linked to the output score passes the threshold for another model.



MFCC information represents the harmonics that correspond to the timbre produced by a particular instrument. Played chords on ukulele should be represented differently in MFCC signals, and should be distinguishable from one another. However one problem encountered in this Lab was that, when the first chord played triggers an appropriate output image, the following chord changes are not recognized by the gemhead which loads the images, despite wekinator and the patch itself reflecting that the scores produced are indeed passing the indicated thresholds. Originally 14 chords were attempted to be trained using wekinator, however this number was decreased to 3 chords in order to attempt to debug this problem. However this problem persisted with only 3 chords/models as well. Another attempt to debug included lowering the thresholds significantly ($>.2$) which still did not lead to appropriate image changes.

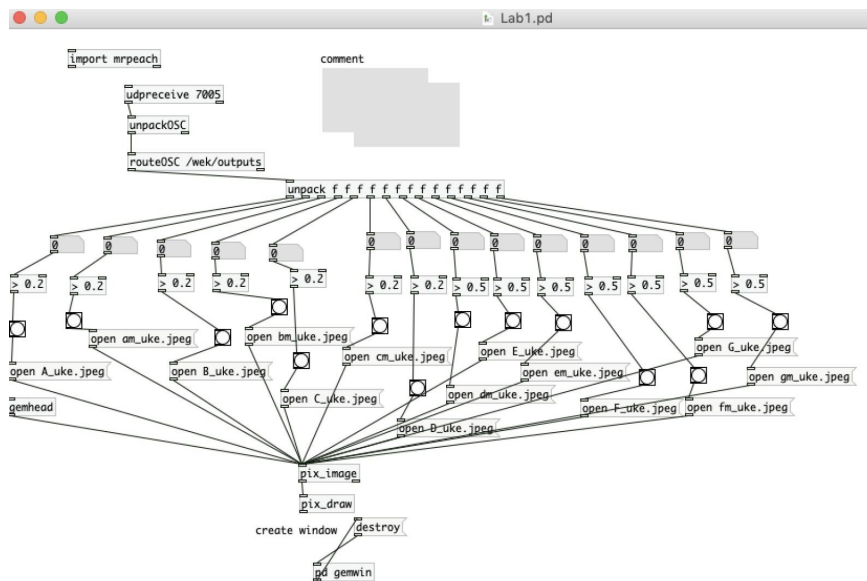


Fig1: First Pd patch which attempted to discern 14 different chords

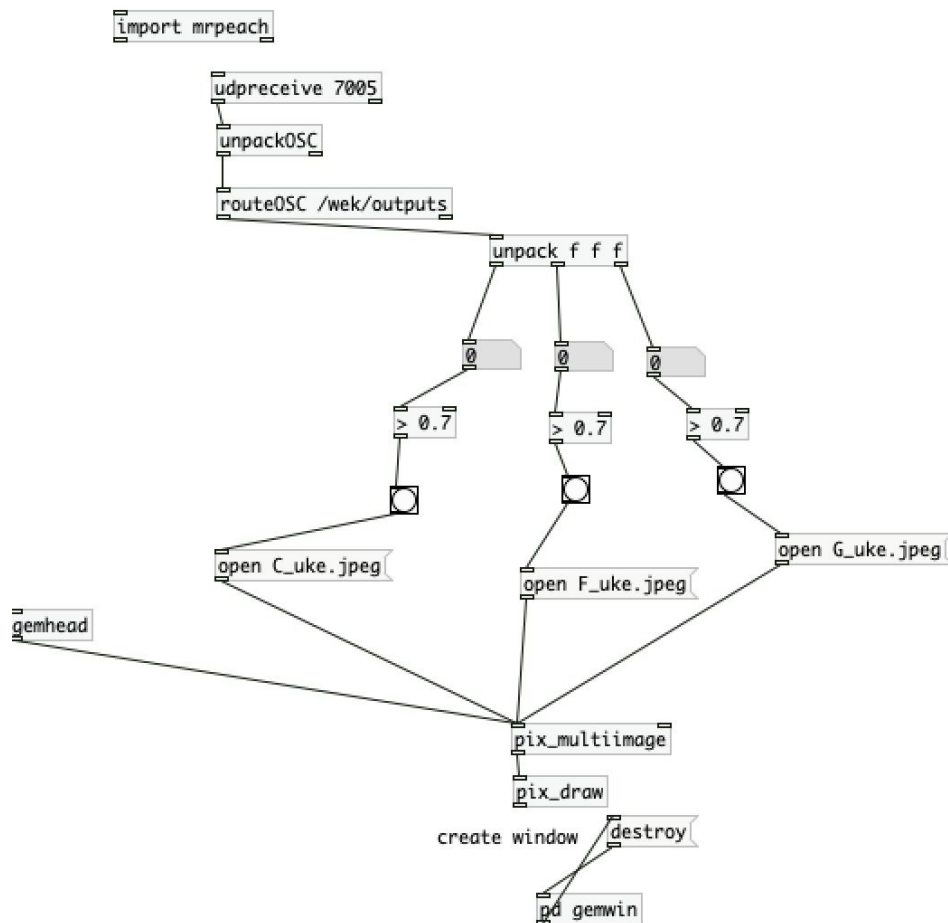


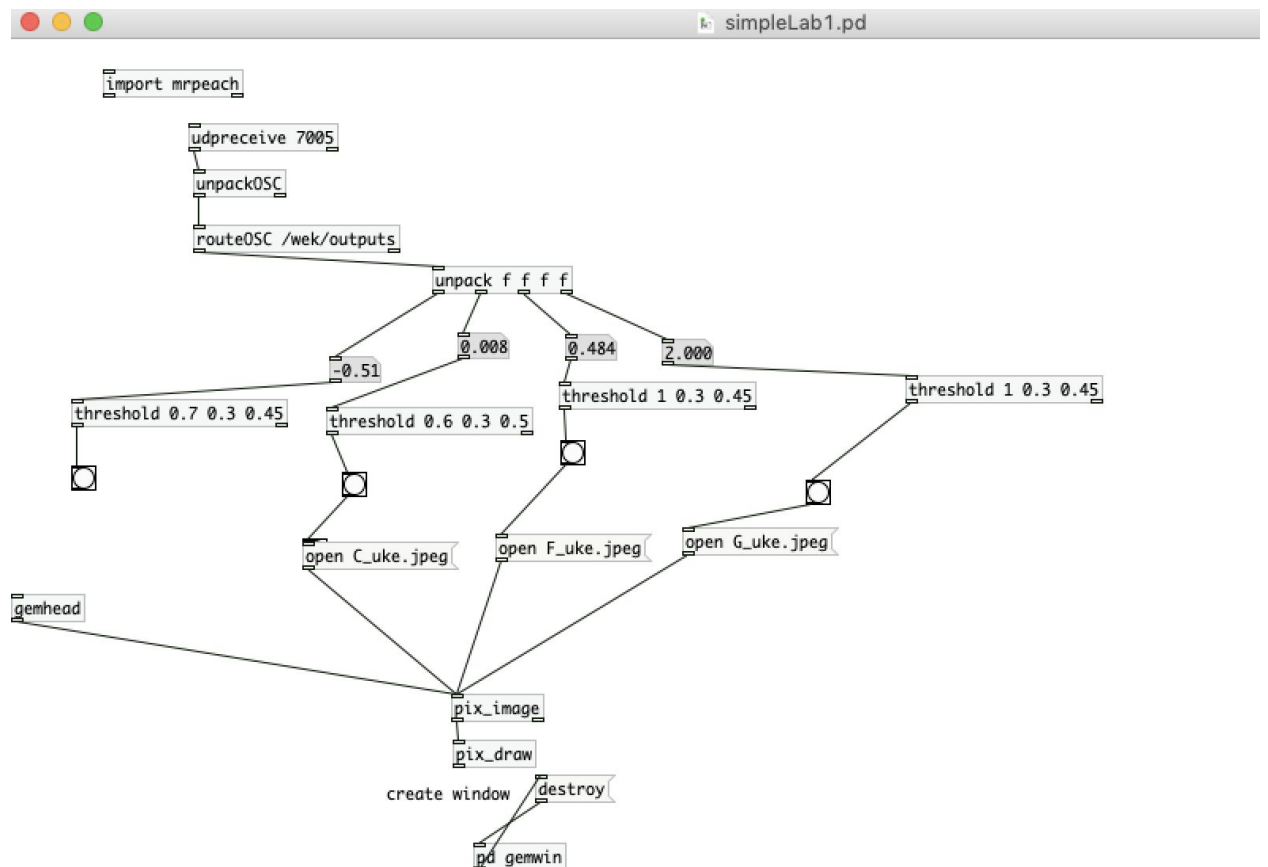
Fig2: Second Pd patch which attempted to discern 3 chords.

Possible causes could stem from the fact that pixelated images are expensive to load in Puredata. The `pix_image` object was the first object used to try and display the images, however another object (`pix_multiimage`) was also tried as its intended use it so load images more quickly. However in the current patch did not change the behavior of the application. It is also possible that the MFCC information as input data for the models may not be the best input for this type of task. Ukulele is a small instrument with limited differentiation of harmonics as the played chords often are configured/fingered in a way that does not produce the same richness of tone as even a guitar (For example, the voicing of an F and C chord on ukulele share the bottom note sounded as a “C”, which is not the case on guitar). So it is possible that the MFCC harmonic information received by the models to train the chords just does not have the right features to distinguish between played chords. Perhaps more training could fix this.

After Feedback from class:

Suggestions in class included training the silence and experimenting with a different threshold function!

The improved patch then includes 4 models, one model for silence and 3 for the main chords (C, F, and G)



These improvements led to the patch working (mostly) correctly! There is a lot of noise in distinguishing between the chords and improvements would be to include a lot more training to strengthen the models. When run the patch will mostly display the chord being played, however flicker between other the other chords causing some chord charts to be displayed that are not what is actually being played on the ukulele. There are probably some more sophisticated/precise ways of programming this type of program in puredata that are able to filter this type of noise, however more training as stated above might also be a solution.