This file describes the API that need to be implemented for the final demo. The system itself is made of three components: emotion recognition (images), music generation, and user interface. In turn, the music generation component contains two subcomponents --- the adjustment of the base song toward the emotion and the computer-assisted music generation.

**Emotion Recognition**

*emotion\_recognition\_model train(images, emotions)*

Given an annotated collection of images (for each image, we have an emotion, which is present on the image; only one emotion per image is used in this demo).

*emotion predict(image, emotion\_recognition\_model)*

Given a trained emotion recognition model and a new image, assign probabilities to each of the emotion classes and select the most probable emotion. For example, for an image of a beach, the model could very likely predict the following distribution:

{

“Anxiety” : 0.01,

“Sadness” : 0.01,

“Awe” : 0.2 ,

“Determination” : 0.05,

“Joy” : 0.3 ,

“Tranquility” : 0.4,

}

And the most probable emotion is

{

“Tranquility” : 0.4

}

The tranquility is the output of the image processing API.

**Music Generation**

*base\_song\_modulated modulate(base\_song, emotion)*

Given a .MIDI file with a base song (e.g. “Happy Birthday to You…”) and an emotion, this method adjusts the scale, tonality, and tempo of the base song to fit the emotion. We call this process an emotion-based modulation. For example, if the emotion is “sad” the music will be in minor form, not loud, and not fast as compared to the case when the emotion is “joy” or “determination”.

*music\_generation\_model train(songs)*

Given a collection of songs in .MIDI format, train a sequence model that can predict a .MIDI note for the prefix of .MIDI notes. The model captures transition probabilities between .MIDI notes.

*computer\_generated\_song generate\_song(modulated\_base\_song, music\_generation\_model)*

Given an emotion-modulated base song, which serves as a seed for the computerized music generation process, and a model trained to generate music, we produce a sequence of new .MIDI notes. For example, we can seed the generative process with a “Happy Birthday to You…” song modulated with the “Joy” emotion and complete it using a trained generative model toward the complete song.

**User Interface**

*bool upload\_image(image)*

Uploads an image and returns the error code upon completion (True for success, False for fail). Used as part of the submission form.

*int select\_base\_song(base\_songs)*

Select a song from a list of base songs to be modulated based on the emotions from uploaded images. Return the index of the selected base song. Used as part of the submission form.

*true play(movie)*

Play the movie made of uploaded images and with a computer-generated song in the background.