Automatic Music Generation Based on Emotions

Bingcheng Li, Caiyi (Giselle) Deng, Divit Karmiani

**1. Introduction**

Music or sound is an important part of a human’s cognition. The right music according to the context of a video accentuates the correct perception of the message put forward. However, the cost of music production is still high. Using AI to generate music upon detection of the emotion behind the video will increase the return on investment especially music for short byte size videos. Applications range from entertainment to online education.

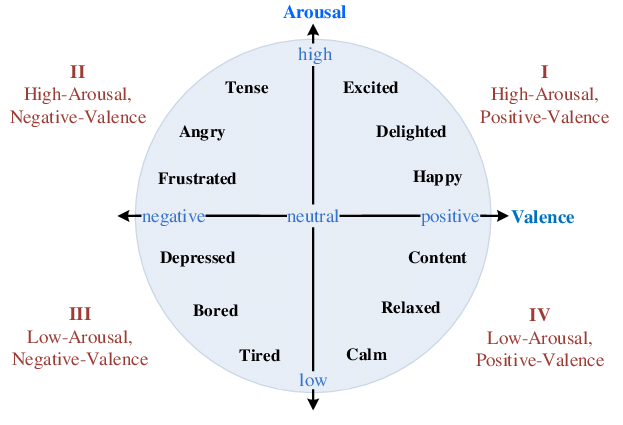
AI seems to be a potential aspect in the future of the music production industry. AI models that train on previously recorded song clips can raise concerns about its ethical implications of copyrights and style transfers. But such style transfers are already part of the natural creative process. Furthermore, innovation can be only done with experimentation in the studio; AI accelerates this innovation by creating millions of combinations of notes, chords or octaves.

Our goal is to create background music for a video which has characters speaking and expressing their emotions through their voice and facial expressions. Judging the context of the video correctly via emotions and changing background music accordingly throughout the video determines the success of this project.

**2. Data Collection**

We used the online data set the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)to train the emotion recognition model. This data set contains both audio and video, with 24 professional actors (12 female, 12 male) saying two statements in a neutral North American accent, and it includes eight emotions, which are natural, calm, happy, sad, angry, fearful, disgust, and surprised. In addition, each expression is produced in two levels of emotional intensity (normal, strong), with each actor performing 60 trials of speech. The audio model used the RAVDESS’s Speech file with 1440 files, while the visual model used the RAVDESS’s Video Speech data with 2880 files.

For music generation using AI, we used the Panda et al. TAFFC Dataset - 2018, also popularly known as the 4Q Audio Emotion dataset or Music Emotion Recognition dataset. This is a public dataset consisting of 900 audio clips (mp3 files each of duration of 30 seconds) which have been annotated into 4 quadrants of Arousal-Valence 2 dimensional chart according to the Russell model. These 900 audio clips are divided into 4 different folders for each quadrant.



*Russell model: Arousal Valence quadrants*

**Other Datasets we considered but discarded later**

We had 2 approaches to follow for music generation.

1. Use raw mp3 or wav files and train models like JukeBox and Wavenet architectures to generate new music.
2. Use MIDI formatted files of these audio and use recurrent neural networks like LSTM or convolutional neural networks to train.

JukeBox and Wavenet architectures require computationally intensive training and thus a higher GPU requirement. Therefore due to lack of resources we discarded the first approach.

However we found out that there is only one dataset VGMIDI which have corresponding emotions annotated. Thus to maintain the plausibility of adding more data for training, we used mp3 files which are annotated according to the 4Q.

As for testing, we recorded video from movie clips with different emotions. Most of the videos were around 5 seconds. After getting a test video for all emotions, we combined these movie clips into a single video to test the model.

For the demo, we wrote down a script that we spoke with different emotions.

**3. Modeling**

For the audio emotion recognition model, we first tried to use random forest and 1D convolutional neural network (CNN) by calculating 30 Mel Frequency Cepstral Coefficients (MFCC) on audio data. However, the test accuracy with the traditional ML method was only about 50%. The validation accuracy with the 1D CNN model is about 55%. When testing with film clips, both models performed poorly where they always predicted test samples as sad or neutral emotion. After getting feedback from the poster session, we realized that adding features variety and tuning the CNN model could increase the audio model performance. We referred to the Speech Emotion Recognition by Shivam Burnwal to rebuild our model. Instead of only using the original audio data, we created synthetic data samples by adding noise, stretching, and changing pitch of original audio to make the model invariant to perturbations and enhance generalization. With the data sample tripled, we also calculated more features. For each audio sample, we calculated 13 MFCCs to get its cepstral representations, a chroma vector to find energy of each pitch class, its root mean square to get energy representation over time, and the spectral centroid to find center of gravity of magnitude spectrum. The total number of features from each audio is 27. Based on the referred model, we have a 1D CNN model with 4 hidden layers with 256, 256, 128, and 64 hidden neurons in the layers. MaxPooling1D is also performed on each hidden layer to reduce the dimensions of the feature maps. After testing with different parameters, we decided to have 35 epochs and set the minimum learning rate to 0.0001.

For the visual emotion recognition model,we built 4 hidden layers 1D CNN and 512 neurons in each layer. For the input layers, we have extracted key frames from the video using OpenCV and parsed out facial landmarks locations with dlib. To featurize the landmarks, we calculated the Euclidean distance between every pair of facial landmark coordinates and input it as a 1D array. The model’s architecture is inspired by a similar classifier using facial landmarks as input. We have experimented with different architectures. In the end, four hidden layers with 512 neurons in each layer, batch normalization with a 2% dropout rate worked the best. Based on observation from the accuracy and loss graph, we trained for 60 epochs with a learning rate of 0.001.

As the visual emotion recognition model performed better in predicting emotion, we assigned different weights (75% on visual and 25% on audio) when combining the prediction from two models.

For music generation, we used prettyMIDI and fluidsynth to read and analyze MIDI files. These MIDI files had notes, chords and octaves of only a single instrument, Acoustic Piano. The visual and audio models recognized 8 emotions which were Happy, Surprise, Angry, Fear, Disgust, Sad, Calm, Neutral. These emotions were mapped according to the Arousal Valence Map in the following manner.

1. Q1: Positive valence, positive arousal(Happy, Surprise)
2. Q2: Negative valence, positive arousal(Anger, Disgust, Fear)
3. Q3: Negative valence, negative arousal(Sad)
4. Q4: Positive valence, negative arousal(Calm, Neutral)

4 LSTM models were trained on 4 sets of datasets or the 4 quadrants in the Arousal Valence map. 4 best models were saved.

**4. Results**

The audio emotion recognition model has a test accuracy around 80% and a validation accuracy is about 60% with 35 epochs. Among all emotions, the precision on anger, calm and surprise are above 60%, while the remaining emotions have precision from 43%~55%. As the number of epochs increases, the training loss continues to decrease to around 0.4 while the testing loss (in this case, validation loss) decreases to about 1.4. The model is slightly overfitting.

The visual emotion recognition model achieves 92%+ validation accuracy on the dataset overall, with 60 epochs. It works pretty well with the RAVDESS dataset. However, considering that all the actors show up right in the middle of the screen with similar size of faces, which rarely happens in real life, the model seems to be overfitting to the training set and doesn’t work that well in real-life situations.

After the emotion is detected and mapped to the Arousal Valence map, the corresponding best model is picked to predict the music.

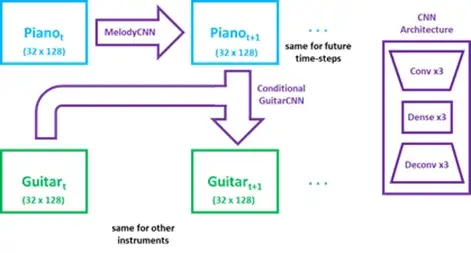
A sequence of notes are predicted after providing some random initial notes to the picked trained LSTM neural network. The length of the notes to be predicted can be set according to us or as the length of the video.

An output.mid(MIDI output file) of length 240 notes(arbitrary) is generated and can be downloaded.

**5. Next steps**

The Ravdess dataset we used to train the emotion recognition model has significant limitations. The actors simply repeat the same sentence in different emotions, so they exaggerate their facial expression and change their tone deliberately. In real life, human conversations are more complicated than a simple sentence and human facial expressions could have multiple interpretations. For instance, a man can speak happily with a frown. If we have more time to work on, we will collect data from real life to add data diversity. Moreover, videos in the Ravdess have all actors facing the camera, so our emotion recognition model performs poorly on video with people not facing the camera directly. Taking video from different angels could help the model learn human emotions that fit better in real life scenarios.

For music generation, we used a single instrument RNN model as we considered a dataset of MIDI files generated from only one instrument Acoustic piano. This was easy and clean to implement but the output music was far from ideal and had limited utility. Therefore, it is obvious that the next steps would be to capture the dependencies between multiple instruments and use a multiple RNN model or better a Convolutional Neural network capable of generating complementary music from different instruments. In order to generate multi instrument tracks compatible with each other, we can use 2 CNNs, one for melody generation for piano instruments, and one for conditional harmony for other instruments. The architecture for this might be something like this:



**6. Team Members Contribution**

Bingcheng: I was responsible for building the visual emotion recognition model and I worked on finding the suitable dataset for emotion recognition. After the visual model was built, I merged the visual prediction to the demo script. When a video was parsed into the demo script, the visual model tried to capture as many frames as possible, and the visual prediction was made based on the facial landmark of the captured frames.

Caiyi:I worked on building the audio emotion recognition model, where I did research on finding the appropriate emotion recognition dataset. After the audio model was built, I also worked on combining the audio prediction with the visual prediction. A video was fed into the demo script, and I converted the video (.mov) file to audio (.wav) file. Then, the audio file was segmented in every 3 seconds with 50% overlap. The audio model made predictions based on the small segments. I also recorded film clips of eight emotions from YouTube for model testing.

Divit: I was responsible for taking the emotion predicted by the visual and audio based emotion recognition models and generating music. I collected the relevant datasets required for music generation which were annotated according to the Arousal Valence map. Once the mp3 files were converted to wav files, I used software to convert them to MIDI files. After the conversion, I converted music files into a sequence of notes as input and the next note as the output. These were fed into a LSTM model. A random sequence of notes are provided to generate further music.