Multi Modal Analysis of songs for emotion detection

Bikram Pandit Radhika Tekade Shubham Patil
Oregon State University
{panditb, tekader, patilsh}@oregonstate.edu

Abstract

Music Emotion Recognition (MER) has grown in popularity in recent years, fueled by the necessity to analyze massive collections of music files automatically, a key task for streaming services, for example. In this paper, we attempt the classification a song based on the emotion associated with it into mood classes.

We have developed a multi-model neural network consisting of CNN and transformers, a novel approach to classifying audio data and also taking an advantage of lyrics to improve the accuracy. Our results showed that our model had 34% improvement as compared to the prior multi-model and uni-model approaches.

1. Introduction

Music is a powerful tool for expressing our feelings, and it is commonly used as a sort of therapy to help us cope with difficult moments in our lives. Music easily reflects feelings and moods; for example, we like to listen to powerful music when exercising; similarly, a delightful and calming song can help us relax when frightened or fatigued. Keeping this in mind, music streaming giants consider mood or emotions a significant factor in classifying songs and building their recommendation system for the user. It also helps users know the spirit of any song beforehand if they are too lazy to listen to the song completely. Hence, Music Emotion Recognition (MER) has been a rapidly expanding topic of research in recent years. In the attempt to contribute to this field of study, our motivation is to provide the user with a system that can play various tracks based on the current feelings of the user.

Sentiment Analysis. Over the years, various techniques were investigated to automatically recognize the sentiments of music with a wide range of datasets and features. Traditionally, the majority of song emotion identification has focused on audio signals of songs [4]. Features like Mel-Frequency Cepstral Coefficients (MFCC) and chords are retrieved from audio information to create emotion classifiers. It turned out to be inadequate to perfectly simulate the perception capability of a human ear [17]. Then came the insight of joining music and lyrics to analyze the mood after a psychological investigation of how the human brain processes these modalities independently [2]. Approaches to use only lyrics turned out to be particularly tough due to difficulty in feature extraction for emotion labeling that stemmed from complications involved with disambiguating effects from text [19].

Multi-modal Analysis.As the internet is burgeoning, music-related online pages and social tags are becoming increasingly valuable for gathering data. Song suggestion systems that extensively depend on afore-said data store information like the song name, description, genre, etcetera from web-scraping instead of actually analyzing the track. [dataset wala point] On the other hand, the single-mode data can only capture a portion of the object's features. With data categorized simply using single-mode data, a significant amount of information is lost. Thus, an increasing number of researchers have begun to focus on multi-modal fusion technology [8].

Here, we focus on building a multi-modal neural network in which two different models will be used to train vocal and lyrical data separately. We developed a late-fusion architecture where the output from two models is combined together and passed down to a classifier. The model is trained to classify the input into four quadrants. A quadrant is a dimensional approach for annotating an emotional music database in which emotions are represented in dimensional space [7]. Russell suggested a model that has two dimensions: valence and arousal. We're employing a Thayer model (based on Russell's circumplex model) that's commonly utilized in dimensional approaches for musical definition.

Firstly, we collected 14,858 songs from Deezer's mood detection database. Then, we labeled each song into quadrants and went through a pre-processing and feature extraction step. We extracted a high-definition Mel-spectrogram from audio and used CNN to train the model. Similarly, lyrics were pre-processed using non-trainable GloVe embedding and then passed down to a transformer to train another model. The outputs from each of these models were

stacked together into a classifier that has four classes as the final output.

We have not only experimented with transformers for lyrics but also used LSTMs and GRUs. However, the former produced higher accuracy. As far as late-fusion is concerned we also tried concatenating two modalities at different network depths. We found that fusing the output from convolution and from the transformer at lower depths was mostly effective.

The contributions of this paper are to-

- Demonstrate how utilizing the song to create a Mel-spectrogram is a better idea than just the song's sound
- Show how multi-modal analysis of songs helps us achieve better accuracy for emotion detection

2. Related Work

Multimodal Analysis in deep learning. Despite tremendous advancement, uni-modal learning still falls short of covering all components of human learning. When several senses are involved in information processing, multi-modal learning helps to grasp and analyze it better. As the name suggests, its objective is to develop models that can handle and integrate data using various modalities like image, video, text, audio, body gestures, facial expressions, and physiological signals. In their paper [18], Summaira et al. provide a detailed review of historical and current baseline methodologies and an in-depth assessment of recent breakthroughs in the given domain. Various DL models are classified into distinct application groups and fully discussed in a variety of media. Chein et al., in their paper [20], conducted a survey study of two modalities viz image and text. They established that feature embedding techniques and objective function design are superior to auto-encoders, generative adversarial nets, and their variants.

Sentiment Analysis using audio data. The audio signals of songs are the area of much, if not all, of the present research on song emotion recognition. Previous emotion recognition systems have proven that convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long-short term memory models (LSTMs) all have advantages. Previous research has revealed that there is still a lot of room for improvement when it comes to audio feature representation and model architecture options. A number of algorithms have been developed to classify songs based on their acoustic characteristics, for example [1, 9, 12].Hidden Markov models(HMMs) [15] were employed in traditional attempts to recognizing emotion from speech [16]. Note length, velocity, and beat note density were used in the symbolic realm using logistic regression and handmade features. Using a deep learning approach, the key was used, and an average of 20 dimensions of Mel-frequency cepstral coefficient(MFCC) vectors were used in the audio domain. This audio, on the other hand, had no chorus and was solely composed of pop piano music [6].Li and Ogihara [11] extract information about the song's timbre, pitch, and rhythm from the music signals. Palanisamy et al. (2020) [13], offers preliminary research on employing CNN-based models for audio categorization, in which CNNs learn from spectrograms by visualizing gradients.

Sentiment Analysis using text data. Several deep learning approaches, like RNN, CNN, and transformers, have been used to tackle the issue of sentiment analysis, and they have proven to outperform classic machine-learning methods. One major difference between the paper cited here and our approach is the multi-modal aspect of our work. Cliche came up with a novel Twitter sentiment classifier using CNN and LSTMs. The algorithm has shown a notable improvement in performance after combining the two. The paper by Devlin et al. [3], popularly known for introducing the BERT model (bidirectional transformers), shows state-of-the-art results on the plethora of NLP tasks while being conceptually simple. We found out that even we were able to achieve the best accuracy using the transformer model. The paper [14] by Patel et al. shows more than 85 % accuracy on the IMDB movie review dataset using RNN as that model is most popular for opinion mining. We did not implement RNN for the simple reason of task discrepancy. Convolutional LSTM deep neural network architecture [7] exhibits a cut above in correct interpretations over KNN, SVM, and Random Forest classifiers. LSTM integrated with Emotional Intelligence (EI) and attention mechanisms in the paper [5] by Huang et al. is an inventive technique that not only proved to be effective on real-world datasets but also introduced higher-level abstraction and emotion's modulation effect.

Sentiment Analysis using both audio and lyrics There has been a prior work that uses a similar concept to ours. However, their work [2] seems shallow as compared to ours in terms of feature extraction and their model architecture. While extracting the Mel-spectrogram we didn't consider diminishing any audio feature so we used a high-resolution spectrogram and instead wanted CNN to downsample since it helps in learning meaningful features. In contrast, they have used a low-resolution Mel-spectrogram. Similarly, we have used GloVe embedding with 200 dimension vectors and a vocabulary size of 50,000 while they have trained embedding solely on available lyrics, which in general doesn't contain a wide range of vocabulary. Their best accuracy on four-quadrant classification is 40% while ours is 55% on the same dataset.

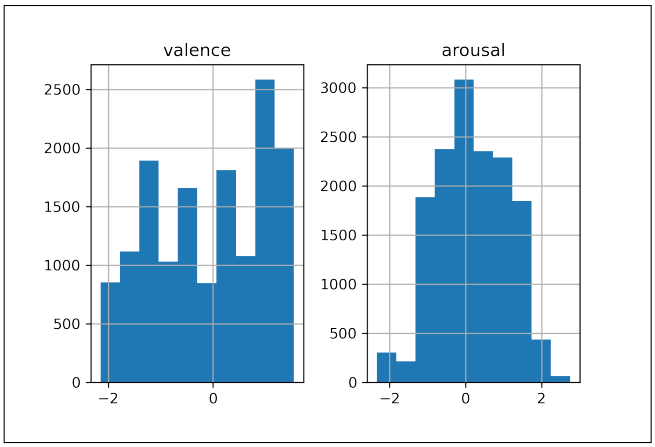

Figure 1. Histogram of valence and arousal data of 14,585 songs.

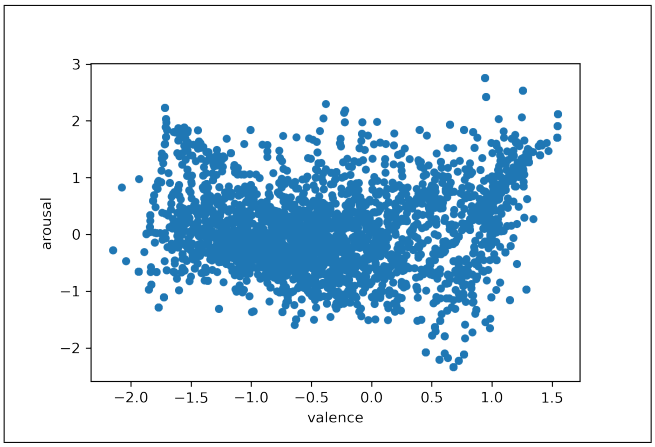

Figure 2. Scatter plot of valence and arousal data of 14,585 songs.

3. Methodology

3.1. Dataset

We collected a dataset from Deezer that consists of a song ID, track's name, artist's name, and corresponding valence and arousal annotations. While Deezer provided API to get track links from IDs, most song IDs didn't result in any song as IDs were obsolete. To overcome this, we queried songs by track name and artist name as we found them to be unique.

As far as lyrics is concerned, unfortunately, Deezer's didn't have any APIs due to copyright difficulties. Nevertheless, lyrics could be viewed on their website using the song ID that we had. Because of this reason, we developed a web scraper that was able to fetch lyrics for every song from the website. However, since not every song had lyrics in their database, from 18,000 songs, we were able to collect 14,585 songs with lyrics.

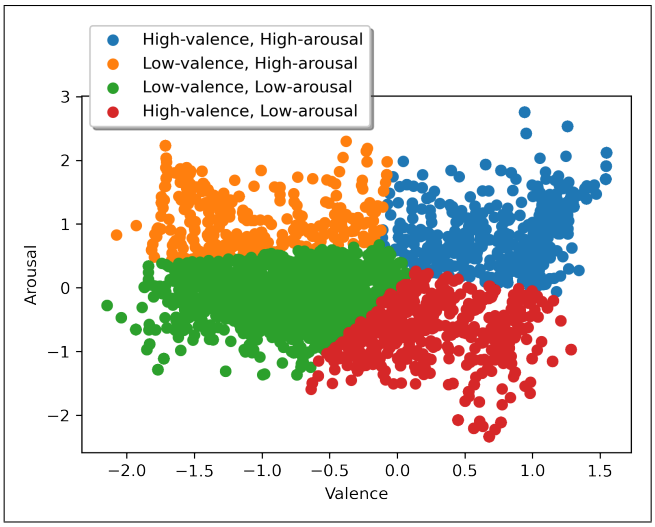

Figure 3. K-means clustering of valence and arousal data.

Quadrants	Valence	Arousal	Class names
Q1	High	High	HV-HA
Q2	Low	High	LV-HA
Q3	Low	Low	LV-LA
Q4	High	Low	HV-LA

Table 1. Labeling classes from given valence and arousal

3.2. Data labeling

We labeled our data into four classes (quadrants) using valence and arousal. From histogram and scatter plot as shown in figure 1 and figure 2, we analyzed that we have a nearly equal number of negative and positive data. Next, we labeled these data using k-means clustering (figure 3) and named our classes as shown in the table [1].

3.3. Feature engineering for Audio

We resampled mp3 songs to fix the sample rate of 22050 with a 30-second duration in a wav format. Then, we had multiple choices to extract feature vectors from audio and with our research we found Mel-spectrogram to have a rich representation of any waveform. We also converted Mel-spectrogram back to audio to confirm this holds true. In fact, we tried to use a higher value for parameters used for conversion so that high-order audio resolution is preserved. Furthermore, we also did a few experiments with audio augmentation such as pitch shifting and time masking. Our Mel-spectrogram was 2-dimensional, representing amplitude at various frequencies over time whose final size was 64×1292. While this is a common technique to preprocess audio for training, prior works use smaller-sized spectrograms that usually make training faster but worsen

the performance.

3.4. Feature engineering for Lyrics

We cleaned the lyrics by eliminating non-alphabetic characters. We further pre-processed lyrics by lemmatization, de-contraction, and converting slang to formal phrases, for example, "I've" became "I have", "gonna" became "going to", "standin" became "standing" etc. We experimented with lyrics both by keeping and removing stop-words. For the augmentation part, we used back-translation from Arabic and French languages. Before passing it to a network, we used a pre-trained GloVe vocabulary size of 50,000 and we looked up each word of a lyric in vocabulary to convert it to a positional value. After padding, the final size of each lyric as a vector was $1 \times$ max_sequence_length where max_sequence_length was the maximum word count in a lyrics. When compared to prior work, they didn't focus on lyrics pre-processing and didn't do augmentation at all.

3.5. Network Architecture

Audio Only. We added a singleton dimension (1D channel) to the two-dimensional Mel-spectrogram so that we can apply 2D convolution over both frequency and time axis. Hence, the input to the first layer was $1 \times 64 \times 1292$. Figure 4 depicts the architecture of a convolutional neural network (ConvNet) [10] . It is comprised of three 2-dimensional convolution blocks, each containing a convolution layer, ReLU activation, and an average pooling layer. Each convolution block produced 64, 32, and 16 feature maps by convolving over frequency and time axis with the stride of 1 and kernel size of 3×3. For every convolution block, average pooling was performed on each of the feature maps with a kernel size of 4×4 after the output from ReLU activation. Finally, the output from the last convolution block was flattened into a size 1×320 and connected to two fully-connected layers producing four outputs.

Lyrics Only. We fed each lyric vector to a pre-trained GloVe embedding layer. The output from the layer is the representation of each lyric as word vectors with respect to other word vectors in a space where semantically similar words are close to each other. Hence, we have a final 2D vector for each input. Next, we also needed positional information of these words so we used sine and cosine functions for positional encoding.

For network architectural choice we experimented with self-attention with several networks, such as CNN, GRU, LSTM, and transformer. The output from these models was of size 1×512 (512 is the output dimension of the transformer) which was connected to two fully-connected layers to produce an output of size 4.

The prior work did not take advantage of pre-trained state-of-the-art models and didn't use the self-attention technique which we used in our problem.

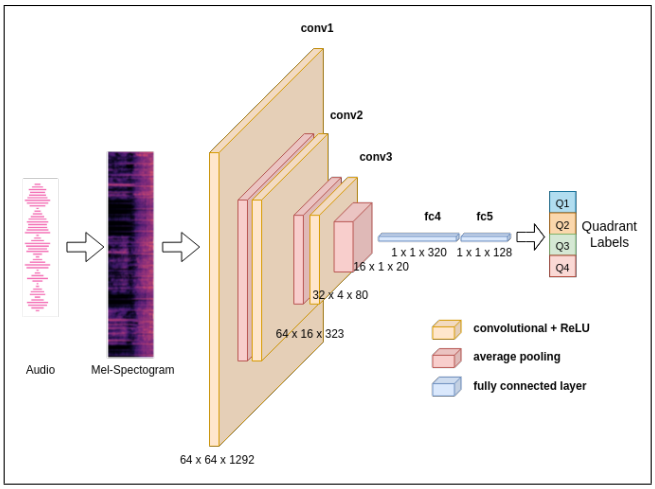

Figure 4. CNN model.

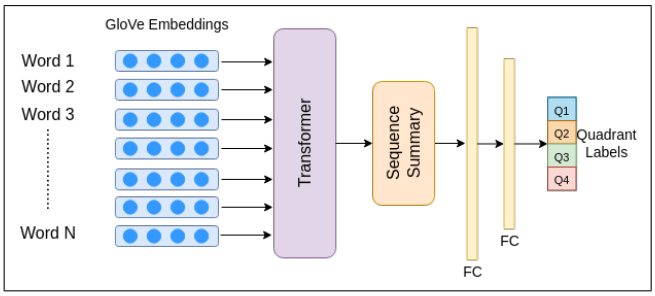

Figure 5. Transformer model.

Fusion. The late-fusion model was constructed using the existing uni-models as explained earlier. Since in late-fusion is all about concatenating the output from two different modalities, we had experimented to concatenating at various depths (Figure: 6). For an instance, in one experiment, we concatenated output directly from the convolution layer and transformer, then added a fully-connected layer afterward. In this case, the size of output after concatenation was 320 (output dimension from the audio model) plus 512 (output dimension from the lyric model) which was 812. Next, we stacked fully-connected layers that took concatenated input of size 812 and produced 4 outputs.

In another experiment, before concatenation, we first passed it fully-connected layers for two different modalities that produced an output of size 8 and then concatenated. Further, we connected it to one fully-connected layer to produce 4 outputs. In contrast, the prior work doesn't seem to use variable depths which in our case we found to change accuracy by a reasonable factor.

4. Results

We ran experiments on Tesla V100 GPU, Tesla K80 on Google Colab, and Apple silicon GPU. The main library

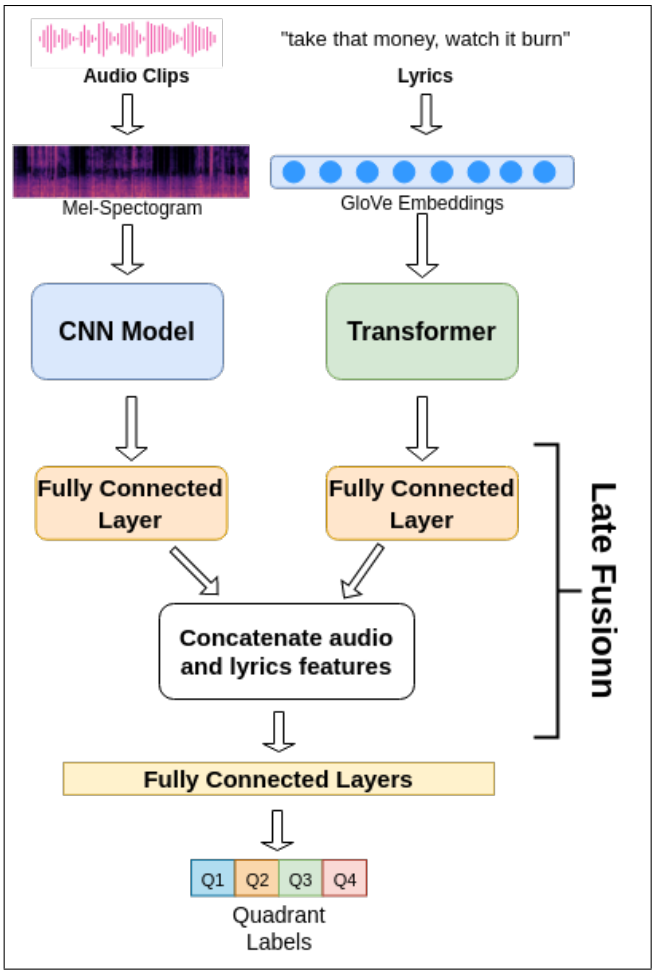

Figure 6. Fusion.

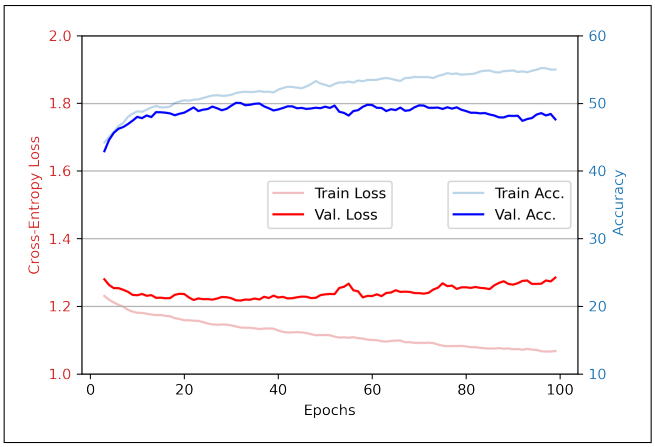

Figure 7. Accuracy and loss plot on audio-net

we used was Pytorch versions 1.11 and a nightly version (on Apple silicon) and supporting torchvision, torchaudio, and torchtext libraries. All test results were reported in files and visualized on a tensorboard. We compiled graphs by exporting results from the tensorboard.

We used cross-entropy as the loss function and the Adam optimizer. We used a batch of 256 but were limited to 128 for multi-model training due to limited resources. For every experiment, we found a learning rate of 0.0003 and weight decay of 0.003 to be optimal. We chose the Adam optimizer as it requires fewer parameters for tuning performance and it has adaptive learning nature.

From 14,585 songs, we used 20% of it for validation and 10% for the test. We evaluated our model based on the accuracy obtained from the test set. Also, we inspected the confusion matrix to get an idea about which classes were misclassified.

For audio modality, since we had a high-resolution Mel-spectrogram we wanted a model to learn with a smaller representation so fewer convolution blocks didn't produce a good result. Stacking high number convolution blocks helped in optimization but led to over-fitting. With all experiments, the best trend was observed for three convolution blocks as shown in figure 7.

For lyric modality, we experimented with CNN, LSTM, GRUs, and transformers. For each of these networks, we used trainable embeddings, GloVe embeddings with frozen weights (non-trainable), and GloVe embedding that is trainable. The non-trainable GloVe embeddings with transformer produced the highest accuracy of 47% as compared to others as shown in table 2. We think since the transformer has a self-attention mechanism, is not limited to sequence length, and tends to perform better in general, it produced the best result for the lyric net as shown in figure 8.

For multi-net, we used the best of individual modalities' settings as a baseline. It is not necessarily always true that the best individual modalities would produce the best performance combined but it tends to work for our problem. We found that combining the result from the audio net and lyrics directly from the convolution layer and transformer and concatenating outputs to dense layers to produce 4 outputs was the best late-fusion setting which produced an accuracy of 55%. Accuracy-loss plot for this setting is shown in figure 9.

In comparison with prior work on late-fusion technique [2] (paper) and https://github.com/SeungHeonDoh/Music_Emotion_Recognition (implementation), they had multi-net test accuracy of 41% which means we have achieved 34% of improvement in generalization of mood classes from songs. We think that the primary components that were different from prior work were data preprocessing, augmentation, use of pretrained embedding, and a transformer.

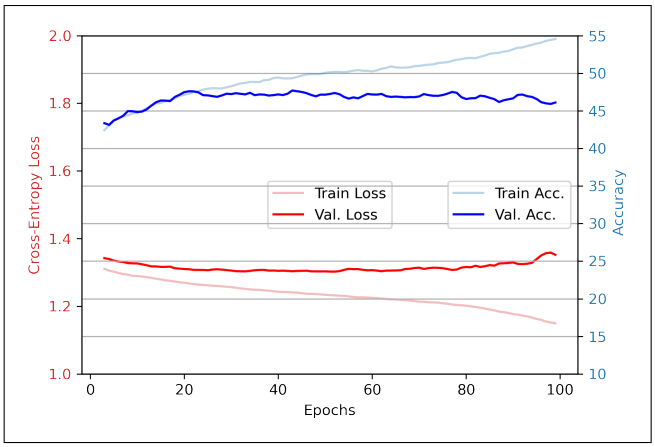

Figure 8. Accuracy and loss plot on lyric-net

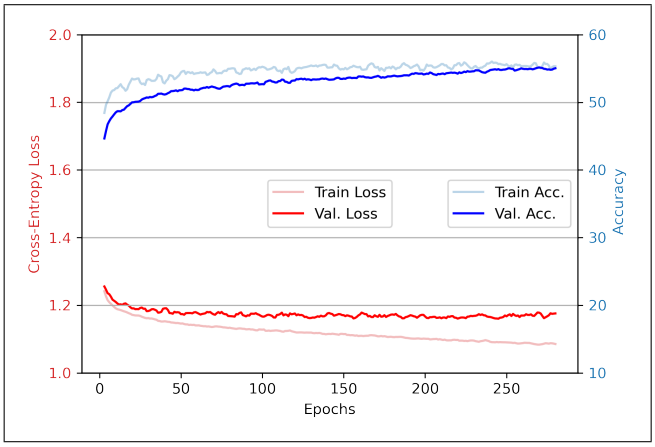

Figure 9. Accuracy and loss plot on multi-net

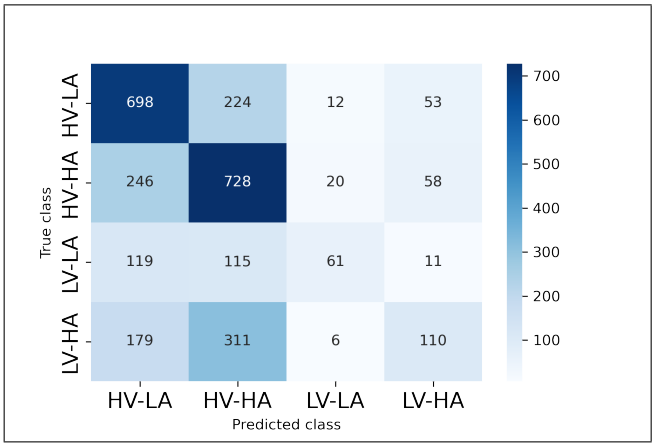

Figure 10. Confusion matrix produced on multi-net with optimal settings

Modality	Models used	Accuracy
audio	CNN	50%
lyrics	CNN	46%
	GRU	43%
	LSTM	45%
	Transformer	47%
bimodel (ours)	CNN (audio) + Transformer (lyrics)	**55**%
bimodel (prior)	CNN (audio) + CNN (lyrics)	41%

Table 2. Accuracies of the different tested approaches.

5. Conclusions

In machine learning, most of the data have multiple semantic meanings which individually is not sufficient to explain it accurately. Inspired by nature and how human interprets things, there is a necessity to combine these modalities or meaning in different dimensions so that model performs more naturally. We have taken an example of a song with audio and textual representation and used the same idea to show how the multi-model approach beats the uni-model.

Our work can be one of the starting points to try various experiments and improve accuracy. Figure [10] shows that our model was not able to generalize songs that have low valence or mostly negative valence. One reason could be because of unbalanced data or biased data. To overcome this, one could use the same model as a baseline to experiment with the balanced and diverse dataset. It is also worthwhile to use pre-trained networks such as ResNet or VggNet, which are the benchmark for image classification. Furthermore, Mel-spectrogram can be interpreted as sequential data over time which might be a convincing reason to use RNN and transformers.

The limitation of our work is that our multi-model implementation only works with English songs. Classifying new songs wouldn't be a problem for our audio net, however, we need multilingual embeddings for the lyric net to understand them. Another limitation is, that since we are using transformers, we might require a large number of annotated songs, which might be hard to collect and harder because song emotion might be subjective in nature. Lastly, although we have used full song lyrics, we have used only considered the first 30 seconds of audio of songs, which might not be sufficient to learn the context. For an instance, a 30-second excerpt might contain mostly music without vocals.
[1]

[1] www.github.com/bikcrum/SongMultimodelAnalysis

References

[1] David Bainbridge, Sally Jo Cunningham, and J Stephen Downie. Analysis of queries to a wizard-of-oz mir system: Challenging assumptions about what people really want. In ISMIR, 2003.

[2] Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. Music mood detection based on audio and lyrics with deep neural net, 2018. i, ii, v

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018. ii

[4] Yajie Hu, Xiaoou Chen, and Deshun Yang. Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In Keiji Hirata, George Tzanetakis, and Kazuyoshi Yoshii, editors, ISMIR, pages 123–128. International Society for Music Information Retrieval, 2009. i

[5] Faliang Huang, Xuelong Li, Changan Yuan, Shichao Zhang, Jilian Zhang, and Shaojie Qiao. Attention-emotion-enhanced convolutional lstm for sentiment analysis. IEEE Transactions on Neural Networks and Learning Systems, PP:1–14, 02 2021. ii

[6] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. arXiv preprint arXiv:2108.01374, 2021. ii

[7] Serhat Hızlısoy, Serdar Yildirim, and Zekeriya Tüfekci. Music emotion recognition using convolutional long short term memory deep neural networks. Engineering Science and Technology an International Journal, 24:760–767, 11 2020. i, ii

[8] Xiaosong Jia. A music emotion classification model based on the improved convolutional neural network. Computational Intelligence and Neuroscience, 2022:6749622, Feb 2022. i

[9] Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer. A music search engine built upon audio-based and web-based similarity measures. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 447–454, 2007. ii

[10] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In Proceedings of 2010 IEEE international symposium on circuits and systems, pages 253–256. IEEE, 2010. iv

[11] Tao Li and Mitsunori Ogihara. Detecting emotion in music. 2003. ii

[12] Beth Logan, Daniel PW Ellis, and Adam Berenzweig. Toward evaluation techniques for music similarity. 2003. ii

[13] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking cnn models for audio classification. arXiv preprint arXiv:2007.11154, 2020. ii

[14] Alpna Patel and Arvind Kumar Tiwari. Sentiment analysis by using recurrent neural network (february 8, 2019). feb 2019. ii

[15] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. ieee assp magazine, 3(1):4–16, 1986. ii

[16] Björn Schuller, Gerhard Rigoll, and Manfred Lang. Hidden markov model-based speech emotion recognition. In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., volume 2, pages II–1. Ieee, 2003. ii

[17] Mehmet Cenk Sezgin, Bilge Gunsel, and Gunes Karabulut Kurt. Perceptual audio features for emotion detection. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1):16, May 2012. i

[18] Jabeen Summaira, Xi Li, Amin Muhammad Shoib, Songyuan Li, and Jabbar Abdul. Recent advances and trends in multimodal deep learning: A review. arXiv preprint arXiv:2105.11087, 2021. ii

[19] Xing Wang, Chen Xiaoou, Deshun Yang, and Yuqian Wu. Music emotion classification of chinese songs based on lyrics using tf*idf and rhyme. pages 765–770, 01 2011. i

[20] Fangyi Zhu, Zhanyu Ma, Xiaoxu Li, Guang Chen, Jen-Tzung Chien, Jing-Hao Xue, and Jun Guo. Image-text dual neural network with decision strategy for small-sample image classification. Neurocomputing, 328:182–188, 2019. ii