CHORDMOSAIC

First Author
Daming Wang
wdm1732418365@gmail.com

Second Author
Haolin Liu
haolinliu@uvic.ca

ABSTRACT

In this project, music understanding and visualization are combined to analyze songs and generate corresponding artwork. We present a system to automatically analyze song lyrics and audio chords to tag music and automatically visualize its mood and harmony. The system begins with some audio files, like WAV or MP3, and applies a music recognition API to recognize and process them. We use a lyrics-based auto-tagging using natural language processing (NLP) and a chord recognition method from audio using music information retrieval (MIR) techniques to preprocess the lyrics, tags, and chords. Then, multi-label classification will be used to generate labels into a text-to-image generation AI model. This will create an innovative picture of the song and describe the mood of the song. Moreover, we can analyze the chords and tones to output as Piet Mondrian-style color blocks based on the chords as an additional output result. In general, this project includes the processes of music analysis, deep learning, and generative art. The meaning of this project is to provide an efficient and accurate way to create a mood drawing or an album/song cover picture for songs.

1. INTRODUCTION

Music is an audio-visual experience in our daily lives that can enrich our lives and spirits. Music is also considered an auditory format of the artwork. However, another standard format of artwork that is often considered is painting and drawing. Therefore, our project wants to discuss the possibility of combining these two artworks together, making them feasible to convert from one way to another. Since we are focusing more on music and audio processing in this course, we will primarily work on converting the audio into drawings in some way. Some previous work on music information retrieval has tackled the classification of songs by genre, mood, and tags using audio signals or textual features [1][2]. However, combining lyric analysis and audio chord recognition to give generative visualizations is still a novel area. This project presents a unified system that can tag songs based on lyrics and chords and then use the tags to create visual art reflecting the song's mood and content.

This project aims to help listeners understand music more deeply by providing visual complements. For example, a song's lyrics usually suggest the main theme and emotions, while the chord progression represents the tonal mood. By combining these, we can generate an image, such as an album cover or dynamic visualization, that represents the song's mood and musical structure.

In general, this project aims to develop under three steps. The first step is to design a lyric analysis model for multi-label tag predictions on genre and themes and mood detection from raw lyrics. The second step is to develop an audio analysis model for automatic chord recognition from music audio and, finally, make a generative visualization module that creates artwork from the combined lyric and chord information. The purpose of this project is to create a visualization work from an auditory artwork.

2. RELATED WORK

2.1 Lyric-Based Auto Tagging and Mood Detection

Song lyrics have been used as textual data for music classification and tagging tasks. There are already some early studies that works on music auto-tagging as a multi-label text classification problem by applying techniques like bag-of-words, TF-IDF, and support vector machines to lyrics [1]. For example, Mayer et al. combined lyric features with audio features to classify genre [4], finding that a fusion of lyrics and audio improved accuracy [5]. Similarly, Mckay et al. evaluated lyrics versus audio and symbolic features for genre classification, nothing that lyrics alone can be indicative but often perform best when combined with audio [6]. Fell and Sporleder (2014) explored linguistic features for lyric-based music classification, highlighting the value of text analysis beyond simple word counts [2].

In addition to genre and tags, mood detection from lyrics has been another important topic for a long time. Hu and Downie (2010) showed that combining lyric test features with audio features can improve mood classification in music libraries [3]. Their paper shows that lyrics can provide many semantic contexts like happy, angry, romantic, etc. More recent deep learning approaches show NLP advances by using recurrent neural networks and attention mechanisms that can model lyric sequences to predict mood or theme. For example, Tsaptsinos (2017) introduced a hierarchical attention network for genre classification purely from lyrics [1], which achieves a good result by

Attribution: D. Wang and H. Liu, "ChordMosaic", in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

considering the song's structure from words to lines to segments in the model. Additionally, Delbouys et al. (2018) extended lyrics-based mood detection with deep learning by using a bimodal approach to predict continuous mood dimensions [7].

Overall, lyric-based auto-tagging is feasible because of large lyric datasets with annotated lyrics. Deep learning models like CNNs and RNNs can effectively extract meaningful features from lyrics for the tag extraction task [8]. In our work, we will try to build a model using a multi-label text classifier to assign mood and theme tags to lyrics, which will then be processed to the next stage of image generation.

2.2 Lyric-Based Auto Tagging and Mood Detection

Another aspect we need to consider is the automatic chord recognition (ACR) from audio. It is a well-established MIR task with over two decades of research [15]. The goal of our work is to transcribe the chord sequences like C major, G7 and A minor from a music audio signal. Fujishima's work in 1999 introduced the use of chroma features like pitch class profiles for real-time chord detection [9]. That gives a basic typical chord recognition pipeline. Firstly, we need to compute a time-frequency representation such as the short-time Fourier transform or Constant-Q transform [18]. Then map the spectrum to 12-dimensional chroma vectors by summing energy into the 12 pitch classes of the octave and finally applying pattern matching or machine learning to identify the chord at each time frame [11]. Early models rely more on template matching and heuristics, sometimes with Hidden Markov Models (HMMs) to enforce the temporal continuity of chords [18]. An HMM can smooth frame-wise chord predictions by modelling transition probabilities between chords. This can be formulated as finding the chord sequence $c_{1:T}$ that maximizes a joint probability with transition penalties [19]:

$$\hat{c}_{1:T} = \arg\max_{c_{1:T}} \prod_{t=1}^{T} P(x_t \mid c_t)\, P(c_t \mid c_{t-1}) \qquad (1)$$

where x_t is the feature (chroma) at time t, $P(x_t|c_t)$ is the likelihood of chord c_t generating that feature, and $P(c_t|c_{t-1})$ is the transition probability from the previous chord. Solutions to (1) are typically found via the Viterbi algorithm in chord HMM frameworks [18].

There are also some novel approaches that use machine learning and deep learning to improve chord recognition accuracy. Convolutional neural networks (CNNs) were applied to chromagrams to learn chord patterns robustly (Humphrey & Bello 2012) [12]. Recurrent neural networks (RNNs), including LSTMs, have been used to model temporal dependencies in chord sequences (Boulanger-Lewandowski et al. 2013) [10]. These network models see chord recognition as a sequence of labelling problems and output a probability distribution over chords at each time step. Sigtia et al. (2015) proposed a hybrid approach mixing an RNN and a temporal model for chord sequences to make it more accurate [13]. Additionally, in order to handle large chord vocabularies, Deng and Kwok (2017) presented an even-chance training scheme with deep models to balance chord class frequencies [20]. McVicar et al. (2014) mentioned that even advanced models benefit from good input representations [14], and Korzeniowski and Widmer (2016) developed a deep chroma extractor by learning an optimized chromagram from the raw audio using a neural network [15]. The chord recognition accuracy was improved by training the network to emphasize harmonic content and suppress noise. All these related works gave us the idea of using a pre-trained model or our own trained CNN/LSTM to output chord symbols from audio. We can then process the chord outputs to the next step.

2.3 Generative Music Visualization

When comes to the phase of converting music into visuals, we can use generative AI, particularly text-to-image synthesis. Models like OpenAI's DALL-E can create images from textual descriptions by using large transformer-based generative models [16]. Recent diffusion models like Stable Diffusion can provide further improved quality and fidelity of generated images via latent diffusion processes [17]. We can use the tags and labels from previous steps and implement API connections to generate images that align with the song's emotional tone.

In addition to lyric-based imagery, we can also use chord progressions to map musical features into colors or shapes. Here, we propose a Mondrian-style visualization for chords since his work is characterized by rectangular blocks of primary colors with varying sizes. Our suggestion is to design a simple algorithm to convert the chord sequence into a neoplasticist art form, echoing Mondrian's style.

Overall, by combining all of the ideas on related works, we aim to produce a composite visualization that reflects both the lyrical and musical content of the song.

3. METHODOLOGY

3.1 Pipeline Overview

The system architecture (Figure 1) consists of three main stages arranged in a pipeline. First, the Lyric Analysis stage takes the raw lyrics of a song as input and produces both a set of tags and an inferred mood description. Next, the Chord Recognition stage processes the song's audio to output the chord progression. The lyrics are analyzed by an NLP classifier to predict tags like mood and genre. The audio is analyzed by a chord recognition model to yield the chord sequence. Finally, the Generative Visualization stage uses the information from the first two stages to create visual content: an AI-generated image guided by lyrical themes and a Mondrain-style artwork guided by chord progression. The two visual elements can be combined or presented side by side as the final visualization of the music. The result is a composite visual representation of the song.

3.2 Tools & Libraries

3.2.1 Program Language & Libraries

The project is implemented in Python/Colab notebook. Python has a rich set of music information-related libraries, which makes it well-suited for the rapid development of machine learning and audio processing pipelines. We use both TensorFlow and PyTorch to develop deep learning models. Classic machine learning algorithms are implemented using Scikit-learn, a Python library with a wide range of ML algorithms. For analyzing musical audio, like extracting features and detecting chords and beats, we use Librosa and Madmom. Librosa is a Python package for music and audio analysis that provides the building blocks for MIR systems. It offers functions for computing spectrograms, chroma features, onset strength, etc. Madmom is an open-source audio signal processing library focused on MIR tasks: `https://librosa.org/doc/latest/index.html`. It includes methods for beat tracking and even a pre-trained deep-learning model for chord recognition: `https://github.com/CPJKU/madmom`. By using Madmom's chord detection module, we can obtain chord estimations from audio. Song lyric analysis is implemented with the Hugging Face Transformers library. Hugging Face Transformers is an open-source library providing thousands of pre-trained transformer models for NLP and audio tasks. For lyrics classification and mood detection, we can fine-tune pre-trained language models on song lyric data to predict mood labels or genres.

In order to create visualizations inspired by musical content, we can use text-to-image generation models. Specifically, we use OpenAI's DALL-E API and Stable Diffusion. By inputting descriptions of a song's mood, theme, or even transcriptions of its lyrics, the model can produce a corresponding image. These generative tools allow the project to visualize musical pieces in creative ways. For example, we can generate an album cover art from the song's emotional content.

For plotting and image handling, we use Matplotlib (if charts such as mood timelines or chord progression plots are involved) and Python Imaging Library for image processing tasks. These libraries help generate images with text or combine multiple visual elements to ensure the results are ready for analysis.

3.2.2 Open-Source Repositories

For chord recognition, we can use implementations like the chord recognition system by Korzeniowski et al. and others. For example, a complete chord recognition pipeline based on deep learning LSTMs is available: `https://github.com/krist311/chords-recognition`. We will use such repositories for initial models, adapting their code to our dataset. For lyrics classification, we take inspiration from open projects like a lyric emotion classification system that classifies song lyrics into mood categories (angry, happy, sad, relaxed) using NLP techniques: `https://github.com/wojtek11530/song_lyric_classification`. This repository can guide our feature extraction and model evaluation strategies. In the generative AI part, we can follow the official Stable Diffusion GitHub repository: `https://github.com/CompVis/stable-diffusion`. By using these open-source repositories, we can have faster implementation progress and focus on the integration of these components rather than building everything from scratch.

3.3 Datasets

Our methodology involves multiple sub-tasks including lyrics mood classification, chord recognition, etc. Each stage requires specialized datasets.

3.3.1 Lyrics Classification & Mood Detection Datasets

We use two primary datasets that provide lyrics, audio, and metadata for a large number of songs, enabling the training of lyric-based classifiers and mood detection models: Music4All and Spotify dataset.

Music4All is a music database that contains a large variety of information per track: metadata (artist, album), user tags, genre labels, 30-second audio clips, and full lyrics for each song. Music4All is a large-scale dataset aimed at MIR tasks, and it satisfies key requirements like diversity of genres and inclusion of lyrics. We use Music4All for its lyrics and associated tags, which include genre and possibly mood indications, to train our lyric classification models.

Spotify is the world's largest music media platform and, therefore, has very rich music metadata information. We can retrieve data from artists, albums or shows and also search for Spotify content. Besides, there are many examples and usage cases on the Internet showing how to use them properly. This dataset is mainly used for acquiring the most basic music information. The tags and possibly mood indications in Music4All are lacking here.

3.3.2 Chord Recognition Datasets

For training and evaluating the chord recognition component, we will mainly use two datasets: Chordify (CASD) and Choco.

Chordify Annotator Subjectivity Dataset is a dataset released by Chordify that contains multiple annotations per song, highlighting subjective differences in chord perception. We incorporate CASD primarily to evaluate the robustness and agreement of our chord recognition system. This dataset allows us to measure if our algorithm's output falls within the range of human annotations.

ChoCo provides 20K+ timed chord annotations of scores and tracks that were integrated, standardized, and semantically enriched from a number of repositories and databases for a variety of genres and styles, which is really convenient for us to implement and integrate with our project.

3.3.3 Integration and New Dataset Construction

For the new dataset construction, since we are not in the coding phase, we can't guarantee what the constructed dataset will look like since our input song files will contain lots of information, including tags, mood, lyrics, chords, etc. We will keep updating this part as the project continues.

4. TIMELINE

4.1 Literature Review & Data Setup

This should be done within a week. We should gather datasets for lyrics with tags or annotations. The basic dataset setup should be done with the correct implementation of using and requiring data from datasets. Ideally, we can get from Music4all and Spotify API to gather the needed information.

4.2 Lyric Analysis Module

This should take about 1-2 weeks. We need to develop a multi-label lyric classifier, which includes preprocessing lyrics and training the model on a labelled dataset. Also, implement or fine-tune a sentiment or mood detection model. We should have a working script that takes songs as input and outputs predicted tags and moods from this song.

4.3 Chord Recognition Module

Similar to phase 2, we should do this in 1-2 weeks. We will need to implement a chord extraction model. We could integrate a pre-trained model from Madmom or some Github programs. By the end of this session, we should get a chord sequence from an audio file input.

4.4 Generative Visualization Module

This part should take about a week. We would integrate the text-to-image API and produce Mondrian-style paintings as output image files. By the end of this session, all jobs should be closely done, and the program should be working fine on test samples.

4.5 Integration & Testing

Take about half of a week. Final tuning and checking whether the program is running accurately or not. Test songs can vary from many sources and different genres. The output should match our input song's mood.

4.6 Evaluation

Take about half of a week. Finish the testing phase and evaluate the results qualitatively and quantitatively if possible. Possibly conduct a small user study or ask peers to guess the mood of the song from the image to see if it correlates. Measure the lyric tag accuracy and chord accuracy as well to make sure that all the components meet acceptable performance.

4.7 Finalization

Finish the final report and possibly a demo video of the system. If possible, embed the program into an interactive UI software or website if we have time. The final stage should take around a week.

5. TEAM CONTRIBUTIONS

5.1 Daming Wang

Daming Wang mainly charges for the audio chord recognition model implementation and the Mondrian-style art generation and is responsible for generative AI integration and overall system integration (APIs). He is also responsible for overall coding quality and bug fixing. Besides, he will also contribute to the final report but write less content and also check the grammar.

5.2 Haolin Liu

Haolin Liu will lead lyric-based research and development, as well as the testing and evaluation phases. She will also be responsible for writing reports and writing most of the content.

6. REFERENCES

[1] A. Tsaptsinos, "Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network," in *Proc. Intl. Society for Music Information Retrieval (ISMIR)*, pp. 553–559, 2017.

[2] M. Fell and C. Sporleder, "Lyrics-based Analysis and Classification of Music," in *Proc. COLING*, pp. 620–631, 2014.

[3] X. Hu and J. S. Downie, "Improving Mood Classification in Music Digital Libraries by Combining Lyrics and Audio," in *Proc. ACM/IEEE Joint Conf. Digital Libraries*, pp. 159–168, 2010.

[4] R. Mayer, R. Neumayer, and A. Rauber, "Combination of Audio and Lyrics Features for Genre Classification in Digital Audio Collections," in *Proc. ACM Int. Conf. on Multimedia*, pp. 159–168, 2008.

[5] R. Mayer and A. Rauber, "Musical Genre Classification by Ensembles of Audio and Lyrics Features," in *Proc. ISMIR*, pp. 675–680, 2011.

[6] C. McKay, J. A. Burgoyne, J. Hockman, J. B. L. Smith, G. Vigliensoni, and I. Fujinaga, "Evaluating the Genre Classification Performance of Lyrical Features Relative to Audio, Symbolic and Cultural Features," in *Proc. ISMIR*, pp. 213–218, 2010.

[7] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, "Music Mood Detection Based on Audio and Lyrics with Deep Neural Nets," in *Proc. ISMIR*, pp. 475–481, 2018.

[8] K. Choi, G. Fazekas, and M. Sandler, "Automatic Tagging Using Deep Convolutional Neural Networks," arXiv:1606.00298, 2016.

[9] T. Fujishima, "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music," in *Proc. Int. Computer Music Conf. (ICMC)*, pp. 464–467, 1999.

[10] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio Chord Recognition with Recurrent Neural Networks," in *Proc. ISMIR*, pp. 335–340, 2013.

[11] T. Cho and J. P. Bello, "On the Relative Importance of Individual Components of Chord Recognition Systems," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 22, no. 2, pp. 477–492, 2014.

[12] E. J. Humphrey and J. P. Bello, "Rethinking Automatic Chord Recognition with Convolutional Neural Networks," in *Proc. Intl. Conf. on Machine Learning and Applications (ICMLA)*, pp. 357–362, 2012.

[13] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio Chord Recognition with a Hybrid Recurrent Neural Network," in *Proc. ISMIR*, pp. 127–133, 2015.

[14] M. McVicar, R. Santos-Rodríguez, Y. Ni, and T. De Bie, "Automatic Chord Estimation from Audio: A Review of the State of the Art," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 22, no. 2, pp. 556–575, 2014.

[15] F. Korzeniowski and G. Widmer, "Feature Learning for Chord Recognition: The Deep Chroma Extractor," in *Proc. ISMIR*, pp. 37–43, 2016.

[16] A. Ramesh *et al.*, "Zero-Shot Text-to-Image Generation," arXiv:2102.12092, 2021.

[17] R. Rombach *et al.*, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

[18] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 Years of Automatic Chord Recognition from Audio," in *Proc. Intl. Conf. on Music Information Retrieval (ISMIR)*, [Location], pp. [pages], [Year].

[19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

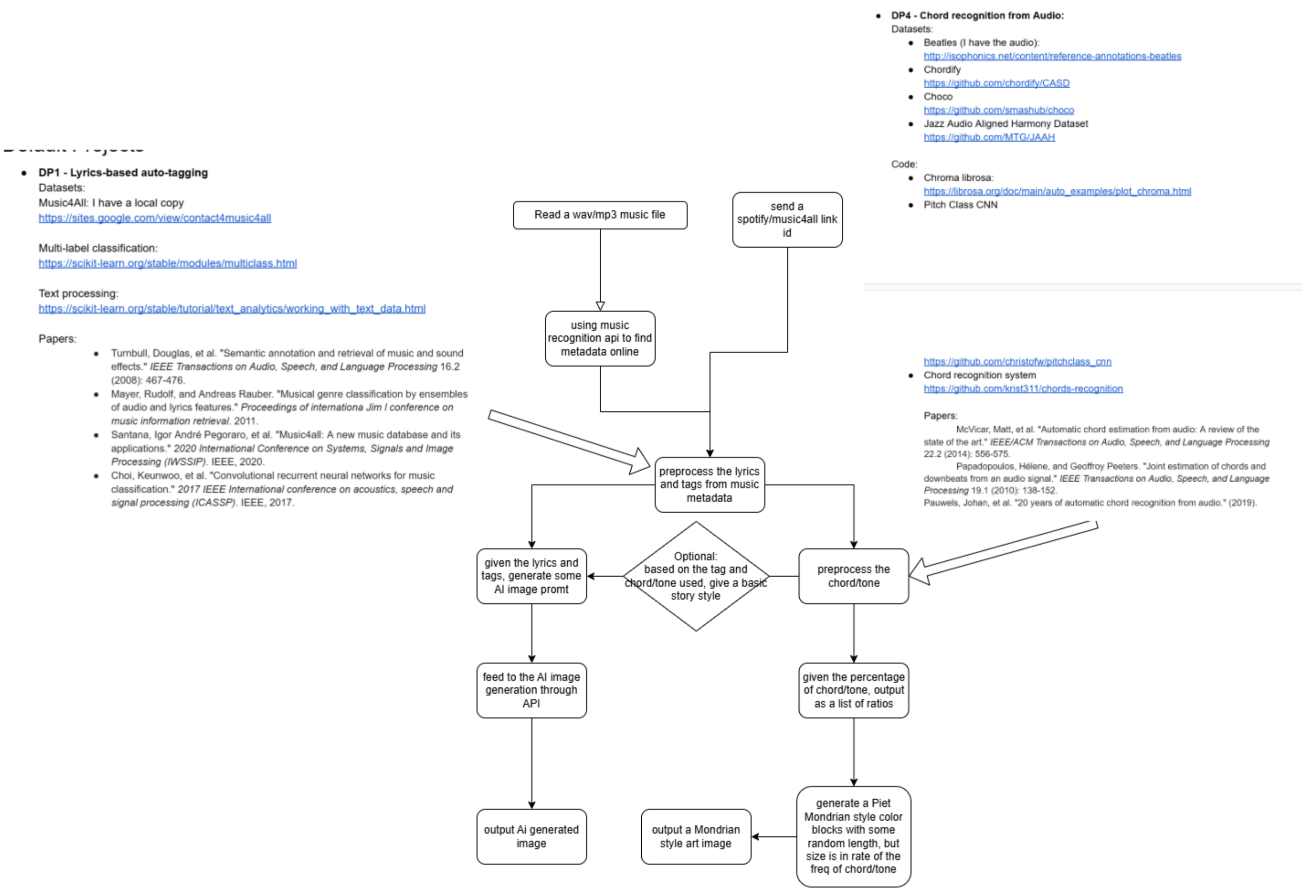
[20] J. Deng and Y.-K. Kwok, "Large vocabulary automatic chord estimation using bidirectional

7. FIGURES AND EQUATIONS

7.1 Equations

$$\hat{c}_{1:T} = \arg\max_{c_{1:T}} \prod_{t=1}^{T} P(x_t \mid c_t)\, P(c_t \mid c_{t-1}) \qquad (1)$$

7.2 Figures



Default Projects

- **DP1 - Lyrics-based auto-tagging**
  Datasets:
  Music4All: I have a local copy
  https://sites.google.com/view/contact4music4all

  Multi-label classification:
  https://scikit-learn.org/stable/modules/multiclass.html

  Text processing:
  https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

  Papers:
  - Turnbull, Douglas, et al. "Semantic annotation and retrieval of music and sound effects." *IEEE Transactions on Audio, Speech, and Language Processing* 16.2 (2008): 467-476.
  - Mayer, Rudolf, and Andreas Rauber. "Musical genre classification by ensembles of audio and lyrics features." *Proceedings of internationa Jim l conference on music information retrieval*. 2011.
  - Santana, Igor André Pegoraro, et al. "Music4all: A new music database and its applications." *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2020.
  - Choi, Keunwoo, et al. "Convolutional recurrent neural networks for music classification." *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017.

- **DP4 - Chord recognition from Audio:**
  Datasets:
  - Beatles (I have the audio):
    http://isophonics.net/content/reference-annotations-beatles
  - Chordify
    https://github.com/chordify/CASD
  - Choco
    https://github.com/smashub/choco
  - Jazz Audio Aligned Harmony Dataset
    https://github.com/MTG/JAAH

  Code:
  - Chroma librosa:
    https://librosa.org/doc/main/auto_examples/plot_chroma.html
  - Pitch Class CNN

    https://github.com/christofw/pitchclass_cnn
  - Chord recognition system
    https://github.com/krist311/chords-recognition

  Papers:
  McVicar, Matt, et al. "Automatic chord estimation from audio: A review of the state of the art." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.2 (2014): 556-575.
  Papadopoulos, Hélene, and Geoffroy Peeters. "Joint estimation of chords and downbeats from an audio signal." *IEEE Transactions on Audio, Speech, and Language Processing* 19.1 (2010): 138-152.
  Pauwels, Johan, et al. "20 years of automatic chord recognition from audio." (2019).

Read a wav/mp3 music file
send a spotify/music4all link id
using music recognition api to find metadata online
preprocess the lyrics and tags from music metadata
given the lyrics and tags, generate some AI image promt
Optional: based on the tag and chord/tone used, give a basic story style
preprocess the chord/tone
feed to the AI image generation through API
given the percentage of chord/tone, output as a list of ratios
output Ai generated image
output a Mondrian style art image
generate a Piet Mondrian style color blocks with some random length, but size is in rate of the freq of chord/tone

Figure 1. Project Pipeline