CSC475/575 Project Brainstorming Spring 2026 - George Tzanetakis

Use this shared document to write ideas about projects as well as to express interests in joining a particular project group. More information about ideas for projects as well as the typical structure can be found in Appendix B of the MIR textbook. Feel free to edit this document to add your own suggestions and ideas for projects and express potential interest. You can express interest in more than one potential project.

For the Spring 2025 offering I would like to also suggest two additional general types of projects:

- Pedagogical exposition: These projects will explore using effective visuals and sounds explain in an intuitive way a concept related to MIR
- Re-implementation: These projects will take existing code for a particular language or environment and re-implement/verify it works in a different environment. For example one could take an algorithm written in MATLAB and re-implement it in Python or an algorithm written in Python and re-implement it in Julia.

This is a long document where you will find many examples of project description from previous years. There is no problem with selecting one of these ideas and reimplementing it/exploring it for your project. If that's the case make sure you add it as a proposal. The default projects are good choices if you want to have more specific, concrete milestones and not work on a more open ended project. Feel free to add your name to any project proposal you are interested in. Eventually you will have to choose a single project/group but at this stage it helps to have an idea of what interest there is for each proposal. A project can be selected by multiple groups.

Example project idea

- Chipify (THIS IS AN EXAMPLE - please add your own)
  In this project the goal will be to convert a music audio recording to a rendering using an old school 8-bit sound generator. You can read more about chiptunes at: https://en.wikipedia.org/wiki/Chiptune. A software simulator of a simple software sound generator will be written. The music recording will be analyzed for extracting the dominant melody, beats and chords and the extracted information will be mapped to appropriate control signals for the sound generator.

Originator: George Tzanetakis (georges@email)
Potential interest: Student name 1 (student1@uvic.ca), Student name 2 (student2@uvic.ca)

Default Projects

These are projects that are well-defined, with clear resources, and milestones. If you are having a hard time choosing a project, you are not aiming for a high grade in the course, and you want to have a structured experience. These projects are the best choices for you and your group.

- **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.

- **DP2 - Beat Tracking and Tempo Estimation:**
  Datasets:
  https://github.com/TempoBeatDownbeat/gtzan_tempo_beat (I can provide the GTZAN dataset)
  https://github.com/GiantSteps/giantsteps-tempo-dataset

  Code:
  https://github.com/mjhydri/1D-StateSpace
  https://pypi.org/project/madmom/
  https://github.com/marsyas/marsgeyas

  Papers:
  - Gouyon, Fabien, et al. "An experimental comparison of audio tempo induction algorithms." *IEEE Transactions on Audio, Speech, and Language Processing* 14.5 (2006): 1832-1844.
  - Percival, Graham, and George Tzanetakis. "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.12 (2014): 1765-1776.
  - Böck, Sebastian, and Markus Schedl. "Enhanced beat tracking with context-aware neural networks." *Proc. Int. Conf. Digital Audio Effects*. 2011.

- **DP3 - Content-based visualization and browsing of sound effects**
  Datasets:
  - FMA - https://github.com/mdeff/fma
  - MTG Jamendo: https://mtg.github.io/mtg-jamendo-dataset/
  - Music4All https://sites.google.com/view/contact4music4all

  Code:
  - Kadenze online course notebooks for clustering and visualization: https://github.com/gtzan/mir_program_kadenze/tree/master/course2/session6

  Papers:
  - Pampalk, Elias, Andreas Rauber, and Dieter Merkl. "Content-based organization and visualization of music archives." *Proceedings of the tenth ACM international conference on Multimedia*. 2002.
  - Khulusi, Richard, et al. "A survey on visualizations for musical data." *Computer Graphics Forum*. Vol. 39. No. 6. 2020.
  - Pampalk, Elias, and Masataka Goto. "MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling." *ISMIR*. 2006.

- **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).

Project summary proposals 2026

Please provide a title and short description of a project you would like to work on, and/or add your name (email) to a project proposed by someone else.

- Chipify (THIS IS AN EXAMPLE - please add your own)
  In this project the goal will be to convert a music audio recording to a rendering using an old school 8-bit sound generator. You can read more about chiptunes at: https://en.wikipedia.org/wiki/Chiptune. A software simulator of a simple software sound generator will be written. The music recording will be analyzed for extracting the dominant melody, beats and chords and the extracted information will be mapped to appropriate control signals for the sound generator.

  Originator: George Tzanetakis (georges@email)
  Potential interest: Student name 1 (student1@uvic.ca), Student name 2 (student2@uvic.ca)

- Classification/Detection of GenAI Vocals
  Goal: Using isolated vocals, or full tracks, to identify whether GenAI platforms like Suno and Udio were used to generate the song. This could extend to segmentation for different sections of the track or components (vocals, guitar, drums, etc)
  Inspiration: Spotify is removing 75,000,000 “spammy” GenAI tracks from their platform https://newsroom.spotify.com/2025-09-25/spotify-strengthens-ai-protections/
  Feasibility: Anecdotally, people can tell the difference between GenAI and non-GenAI music, at least for true positives. This could mean that some data mining could lead to an effective classifier. Deezer is using a new tool to identify these generated tracks https://newsroom-deezer.com/2025/01/deezer-deploys-cutting-edge-ai-detection-tool-for-music-streaming/

  Originator: Nathan Pannell (npannell@uvic.ca)
  Potential Interest: Ryan Dreher (rdreher@uvic.ca)

-

Project Summary Proposals from previous offerings

- Chipify (THIS IS AN EXAMPLE - please add your own)
  In this project the goal will be to convert a music audio recording to a rendering using an old school 8-bit sound generator. You can read more about chiptunes at: https://en.wikipedia.org/wiki/Chiptune. A software simulator of a simple software sound generator will be written. The music recording will be analyzed for extracting the dominant melody, beats and chords and the extracted information will be mapped to appropriate control signals for the sound generator.

  Originator: George Tzanetakis (georges@email)
  Potential interest: Student name 1 (student1@uvic.ca), Student name 2 (student2@uvic.ca)

- **DIY DJ**
  The goal of this project is to build a program that allows users to easily create DJ mixes from multiple songs and/or music recordings. This may be done by analyzing each provided recording and looking for several identifiers such as BPM/tempo, key + chords, beat structure/similarity and combining them optimally with audio splicing. An additional goal for this implementation could be to adjust/match the BPM of differing music recordings, as this will allow a smoother transition for beat matching: https://en.wikipedia.org/wiki/Beatmatching

  Originator: Kevin Bao (kevinshbao@gmail.com)
  Potential interest: Dean Golan (deangogolan@gmail.com)

- **Convolution/IR-based Silent Vocoder/Talkbox**
  Vocoders work by filtering/forming one audio signal through another. Talkboxes work by projecting an audio signal into the mouth (in mono). Silent lip reading techniques allow us to simulate this effect with convolution. Using the IR of the mouth itself, create a convolution processor that can replicate something like a vocoding/talkbox effect - in stereo.
  Originator: Matthew Davidson (mcd@uvic.ca)

- **Digital MultiFX Guitar Pedal (Group formed)**
  The goal of this project is to build a program that allows users to create smart (adaptive) digital multiFX guitar pedals. The program would take key and tempo as input and apply

dynamic effects via DSP to a real-time signal from the user's guitar (signal-program I/O via iRig HD2). Effects would also adapt via real time signal information, such as energy of the plucked string. Ideally, new digital effects could be created to achieve a unique sound by way of MIR techniques.
Similar work: Tonebridge, Feature extraction, AmpliTube
Group members: Sonia Rosenberger (sonia@soniarosenberger.com), Hoksolinvan Chhun (chhunhoksolinvan.prof@gmail.com), Lachlan Lunny (wl.lunny@gmail.com)
Additional interest: George Wiseman (georgeawiseman@gmail.com)

- **Perceptual Stereo Audio Visualizer**
  The goal of this project is to create a visual representation of music that represents stereo information in a way that mirrors perception. Stereo left-right would be mapped to the x-axis and frequency would be mapped to the y-axis, with the possibility of using colours and brightness to convey additional information. Stem separation could be used to identify distinct elements. The end result would ideally be something like in this video: https://www.youtube.com/watch?v=TEjOdqZFvhY although perhaps with a different art style.

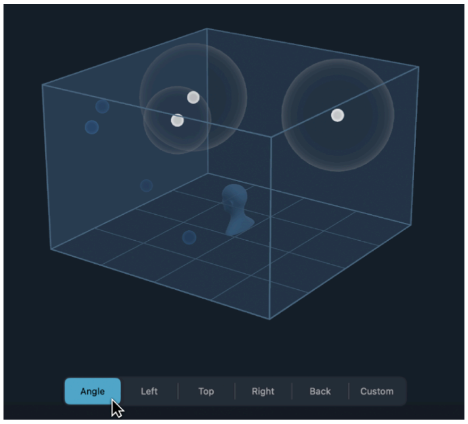
  Originator: Mckinley Wood (mwood@dhdev.ca)
  Potential interest: Kian Dunn, Isaac Wilson,

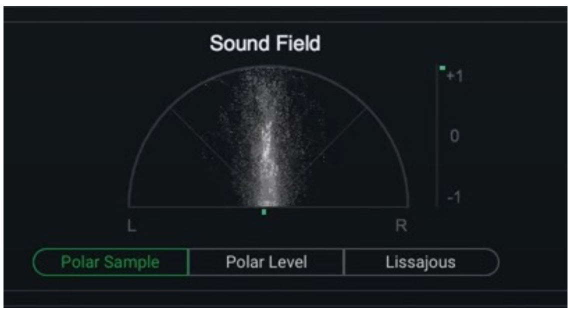
- **Interactive Visualizer in Godot engine**
  Perform some kind of stem-splitting or sound source separation to separate either an audio track or real-time mic input into multiple concurrent audio streams, which are then displayed independently with bar-chart (I think that's called a VU meter) or sine wave visualizers. I'm familiar with the Godot game engine and feel it would work great for the GUI/visualizer aspect of the project.
  Originator: Owen Lutwyche (owenoel333@gmail.com)
  Potential interest: George Wiseman (georgeawiseman@gmail.com), Lyden Buttnor (lyden9@gmail.com), Onur Kilic (kantasonurkilic@uvic.ca)

- **Voice Training Karaoke Game**
  This project aims to develop an interactive game that evaluates singing accuracy based on pitch and timing. The goal is to use stem splitting and pitch analysis and matching to create a karaoke track, and a pitch guide that the singer is trying to match based on the original recording. The game will provide real time visual feedback through a ball following a pitch graph changing color based on the singers accuracy (not set in stone). I am hoping to integrate Fmod and Unity in this project to gain experience in audio for video games.
  Originator: Isaac Wilson (isaacbwilson@hotmail.ca)
  Potential interest: Ryan Hasman (ryanhasman@uvic.ca), Christina Shen(christina.shen@icloud.com) Greyson (greyson.mueller@gmail.com)

- **Converting music to images/art**
  I don't have a very specific project idea, but generally I think it would be interesting to analyze a song or playlist and convert that to art, by choosing different colours, art styles, etc., to create a prompt for an AI image generator. For example this could be used to generate cover images for people's playlists based on the songs the playlist contains.
  Originator: Molly Stewart (mollystewart@uvic.ca)
  Potential interest: Jagjeevan Nahal, Will Satterthwaite (williamsatterthwaite1@gmail.com), Kevin Bao (kevinshbao@gmail.com)
  Dean Golan (deangogolan@gmail.com)

- **Mashify - Creating a mashup of two songs**
  We are trying to create a program that takes two songs and mixes them together in a way that sounds good. It will extract the vocals, instrumentals, key, and BPM from each song and combine the isolated parts into a mashup of both songs. If possible, we would also use machine learning to label different parts of the song (ex. Chorus, bridge, intro) to create a more cohesive mashup.
  Originator: Justin Nguyen (justinator999@gmail.com)
  Potential interest: Jim Duco (jimduco@uvic.ca)

- **AR Spatial Audio Visualizer**
  The idea is to make a spatial audio (or ambisonic) visualizer similar to this:
  

  But decentralize the visual information, so instead of spheres there is something like:
  

  I would like to explore using Unreal engine or Unity to implement it in AR so that a sound designer working on a VR film or game could listen to their mix with a headset on and see where in space the sound is coming from with a scatter-plot-like visualisation similar to the second image above.

  This paper may be of interest: Developing Tools for Audio Visualization and Data Sonification in Unreal Engine. The images are from Apple's Dolby Atmos Object Panner and Izotope's Insight 2.

  Originator: Stella Drinkwater (spd@uvic.ca)
  Potential interest: Matangi Priya (mpriya@uvic.ca),
  Ryan Hasman (ryanhasman@uvic.ca),
  Adin Nelson (adinnelson@uvic.ca)

- **Pacing for Runners**

  A variation of the TempoRun project from a previous year. I would like to have the input be the runners tempo for various paces. Songs can then be found that share the same tempos. Then a runner can also input their workout with their goals paces and it can create a playlist that paces the runner through their session.

Originator: Elijah Larmer (eliageon4@gmail.com)

Potential interest:

Ryan Hasman (ryanhasman@uvic.ca)
Aileen Klassen (aileenklassen@uvic.ca)

- **Music Genre Classification Using Swin Transformers with Mel-Spectrogram Feature Fusion and using Conformal Prediction as a post-processing module.**

  This project explores music genre classification using deep learning techniques. The GTZAN Music Genre Dataset is pre-processed by converting audio signals into Mel-spectrograms, representing time-frequency features as images. A Swin Transformer-based model is trained on both raw images and Mel-spectrogram representations, leveraging feature fusion to enhance classification performance. Additionally, conformal prediction is applied as a post-processing module to provide reliable uncertainty estimates and improve the robustness of genre predictions.

  Originator: Sarah Firouzabadi (sarahfirouzabadi@gmail.com)
  Potential interest: Student name

- **Nier Automata Tone Filter**

  In the game Nier: Automata, PlatinumGame studio developed a tone filter plugin with Wwise for a chipify effect. When the player is doing hacking minigame, the BGM of the game gradually gets mixed with a synthesized 8-bit like sidetrack.

  This effect is different from a regular downsampled/quantized bit crusher effect where you just try to downsample audio to make it sounds like coming out of a crappy 90s console. Instead, you are using many ultra-sharp bandpass parallel IIR filters in parallel to extract the tone, then distort and synthesize the sidetrack with what you get.

  https://themaister.net/blog/2019/02/23/recreating-the-tone-filter-from-nierautomata/

  Luckily, The Maister from Kronos Group has already done lots of experiments and found out the best way to do it. The plan is to re-implement what Maister has done in Graphite as a toy project in JUCE, from the filters to the SIMD optimization.

  Base Deliverables:

  Parallel IIR filter design

VST3 with GUI

Chiptune Synthesizer

Optional (Stretch Goals):

SIMD (for 48 IIR filters)

Unity native plugin w/ demo scene

Misc:

This project is intended to be done solo. However, if you are interested in joining me, you are welcome to talk. Ideally, you have some experience writing low level C++ codes and know some basic music theory, you know how to compile things with JUCE Framework, and you have experience working with C++ GUI frameworks like OpenFramework, IMGUI or best of all, JUCE itself. The workload will likely be considerably heavier than the example projects.

Originator: Zhixin Fang (zhixinfang@uvic.ca)

Potential Interest:

- **SP-13 Detecting AI generated audio [from Resources … Spring 2024 below]**
  Probably use a classification ML model to identify AI generated sound. This makes the code base pretty centralized which is why I intended to do it solo, but if you're interested and you have experience working with classification models, sound processing and python for data science, HMU.
  Datasets:
  https://ai.meta.com/datasets/dfdc/
  https://www.kaggle.com/c/deepfake-detection-challenge
  https://keithito.com/LJ-Speech-Dataset/
  https://zenodo.org/records/5642694
  Code/Resources:
  https://github.com/RUB-SysSec/WaveFake
  https://github.com/MarkHershey/AudioDeepFakeDetection?tab=readme-ov-file
  Papers:
  Khalid, Hasam, et al. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset." *arXiv preprint arXiv:2108.05080* (2021).Khalid, Hasam, et al. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset." *arXiv preprint arXiv:2108.05080* (2021).
  Wijethunga, R. L. M. A. P. C., et al. "Deepfake audio detection: a deep learning based solution for group conversations." *2020 2nd International Conference on Advancements in Computing (ICAC)*. Vol. 1. IEEE, 2020.

Khanjani, Zahra, Gabrielle Watson, and Vandana P. Janeja. "How deep are the fakes? focusing on audio deepfake: A survey." *arXiv preprint arXiv:2111.14203* (2021).
Hamza, Ameer, et al. "Deepfake audio detection via MFCC features using machine learning." *IEEE Access* 10 (2022): 134018-134028.

"Originator" : Brandon (brandonrogers@uvic.ca)
Potential interest: Owen Lutwyche (owenoel333@gmail.com) Carter Conboy (carterconboy@uvic.ca)

- **Solo Finder**
  The Idea here is an application that can take a song as input and automatically determine if the song contains an instrumental solo. If the song does contain a solo then the application would create a timestamp and instrument label for the solo.

  Originator: Ryan West (ryanwest@uvic.ca)
  Interest: Joel Chamberlain (joel.chamberlain38@gmail.com)

- **Cardio Matcher**
  A derivative of tempo run, in this idea we take the beat of the song you're listening to and match it to your heartrate (via a fitness tracker). Ideal for longer music or instrumental so you're able to accurately match the intensity of your workout.

  Originator: Jordan Prasad (jordyjordanp@gmail.com)
  Interest: Jayant Sapra (jayantsapra@uvic.ca)
  Interest: Blaize McGovern (blaizemcgovern@gmail.com)
  Interest: Adin Nelson (adinnelson@uvic.ca)

- **Music generative that played on Blockchain**

Build a platform/system that music enthusiasts can create and sell their music piece on a decentralized system like Bitcoin,Ethereum. To apply our knowledge in this course, we will extract some music features and encode into a vector with some randomized factor (for example, the input is a humming melody, the output is the seed for a generative model, the instrument type,...). Then, we will help users publish it to NFT and build an NFT player for that. A music piece can sell, transfer decentralized with Bitcoin tx: https://ordinals.hiro.so/inscription/231209aff698647145949baa888ebc19e0f412bac046b49f823066cc91c93fa8i0

So there are some works, I can think of:

- Music feature extraction and encode
- Music generator from melody
- Encode data above (chiptune and several music parameter)
- Blockchain NFT creation (smart contract and stuff)
- NFT browser player visualization (everything is stored in blockchain, not server!)

So, the project is quite similar with music generator and Chipify, but we 21will focus on adapting technique to blockchain world (small, efficient data)
Originator: Quoc Dung Van (vvstdung89@gmail.com)

- **Graphical Music Visualizer**

Create a program which accepts input audio (music) and plays it back over a visualizer. As well, it has multiple modes for visualizing the sound file (ex. different style of visualizer for rock music, electronic music, etc.), making various psychedelic, geometric, and multicoloured patterns based on the mode and the structure of the sounds. The visualizer(s) will incorporate signal processing and explore different visual triggers, such pitch changes and sudden volume shifts to synchronize animations with the music's dynamics.

Originator: Cyrus Parsons (cyrus123live@gmail.com)

Interested - Michael Schmidt (michaelschmidt@uvic.ca), Nick Baker (nickbaker9941@gmail.com)

– PixelTone – An Interactive 8-Bit Music Sequencer & Pixel Art Melody Creator

The PixelTone 8-bit Music Sequencer is a creative tool that allows users to compose music or visually explore 8-bit melodies through pixel art. Users can click on a grid-based interface to place notes, which are then played back in a loop, mimicking the classic chiptune sounds of retro video games. Alternatively, users can create pixel-based patterns without a musical goal and discover unexpected melodies from their visual compositions.

Originator: Shawn Zhang(leijiezhang@uvic.ca)
Interested: Peter Yu(peteryu0131@gmail.com)
Leon Yi(qq1071757405@gmail.com)

- **Geometry Dash like Rhythm game level generator**

Create a program that takes a song file and generates a level based on the song's properties (BPM, frequencies used, mood, etc).
This project will have 2 parts:

- a python program that takes in a song file and outputs a file of the level generated
- a game program (probably Unity) that allows you to play the level.

The generator will customize several things for the level such as: speed, obstacle positioning, background/foreground/level, animations, etc.
We will have to make a basic version of the geometry dash that parses the generated output and allows you to play.
Originator: Ryan Hasman (ryanhasman@uvic.ca)
Potential interest: George Wiseman (georgeawiseman@gmail.com) Onur Kilic (kantasonurkilic@uvic.ca) Adam James (af34james@gmail.com)

- **Timbre-based voice trainer**
  This project is a component of a larger application that supports people undergoing voice training in developing their preferred vocal timbre by mimicking a synthesized (or otherwise modified) version of their voice with any desired timbral modifications. Semantic timbral descriptors (e.g., *breathier, huskier, higher*) will be given by the user to apply to a recording of their voice, and the resulting output can be tweaked further using additional descriptors. The goal of the final product is to be a speech pathology tool to help people looking to explore what their voice is capable of (e.g., trans people seeking a more feminine/masculine/neutral voice, people with language impairments, or voice actors) do so in a healthy, intentional way.

  Originator: Isaiah Doyle (isaiahdoyle56@gmail.com)
  Potential interest:
  Aileen Klassen (aileenklassen@uvic.ca)
  Elijah Larmer (eliageon4@gmail.com)

- R&B Song Grader
  The goal of the project is to take a song and grade it on how R&B it is. In order to do so, we will be breaking it down into its subgenres and giving it a grade. May extend the scope to include other genres, in which songs would be graded to how well they fulfill a certain genre.

  Originator: Natasha Au-Duong (natashaad132@yahoo.com)
  Potential interest: Michelle Chen(mish.pyc@gmail.com), Sara Subedi (ssubedi@uvic.ca)
  Hannah Konings (hannahkonings@uvic.ca)

- Interactive game generated from input
  The main ideas of the project are to create a browser-accessible game that is generated from an audio input, probably audio upload. Additional work is different audio visualization methods that sync with the tempo and beat of the selected/uploaded song. A stretch goal is Spotify integration to make a wider range of music selection available. We are considering using JS, Godot, or Pygame; an early goal would be to decide features and select the best tool to meet them.

  Originator: noahedjacobsen@uvic.ca
  Potential interest: danigallegdup@uvic.ca,
  Potential Interest: Erfan Golpour (golpourerfan@gmail.com)
  Weiting Ye(atuony0312@gmail.com)

- **Chord Recognition from Audio**
  The goal of this project is to develop a system that detects and recognizes musical chords from audio recordings. The project will involve analyzing the harmonic content of audio signals, identifying chord structures, and implementing chord classification techniques. We may also explore adding a visual representation of the detected chords, such as pitch class profiles or progression timelines, depending on feasibility and project scope. (We plan to extend the project with further features to enhance accuracy, interactivity, or functionality as we refine our approach.)

  Originators: Aran Cinar (arancinar@gmail.com), Anthony Ho (AnthonyH3925@gmail.com), Swoyam Rajbhandari (swoyamrajbhandari@gmail.com), JungJoo Yoon (5akfl5@gmail.com)
  Potential interest:
- **Guitar Auto Tuner**
  The idea of this project is to allow anyone to "solo" over a song, regardless of their guitar playing ability. Given an input file and a key, this app will auto tune the input forcing it into the given key. Tempo and rhythm will be up to the guitarist, while correct note choice will be managed by the program. Multiple methods of auto tuning will be explored using this method in an attempt to see which style produces the best results. If time permits, live signal sampling could be utilized instead of an input file.

  Originator: Alexmerkl50@gmail.com (solo)

- **vocal deep-fake:**
  get a tiny sample of a person's voice and make a system to generate every speech with that person's voice. This video describes how we can build it. Here are the papers:
  https://arxiv.org/abs/1806.04558
  https://arxiv.org/abs/1912.05566

Originator : Jay Milishia (jaymilishia@uvic.ca)
Potential interest: Adi Lal (adithyalal@uvic.ca)
Youngchan Song (youngchan0518@gmail.com)

Need inspiration? Here are some project descriptions from previous semesters. As mentioned above, there is no problem with selecting one of these ideas and reimplementing it/exploring it for your project. If that's the case make sure you add it as a proposal for Spring 2025.

- **Sinusoid Incremental Game:** We could demonstrate the concept of creating complex audio signals using a linear combination of sinusoids by creating an incremental game (think cookie clicker, adventure capitalist, antimatter dimensions etc.) The basic concept would be the player purchases sinusoids that then generate currency which can be used to buy additional sinusoids which will eventually stack together into some complex audio signal, with more currency being generated based on different aspects of the combined

- **Music Preference with Complexity:** The topic of "what makes people prefer one song over another" is hugely complex, but one component may be the level of redundancy (i.e. repeated patterns). It has been hypothesised that people prefer music with an intermediate amount of redundancy: we enjoy finding patterns, but don't want to be bored. This project would attempt to test this idea; we would use audio processing to calculate a metric of redundancy from a piece of music, and would then use data mining to calculate preference metrics likely from Spotify, Apple Music, et al. Investigating the interaction between these two variables would hopefully give some insight into the hypothesis. Controlling for variables would obviously be an important aspect of this project and it will take some thought to determine how to do this properly. As a fallback plan, creating a way of determining musical redundancy would itself be an interesting project with various potential uses. signal. Off the top of my head there could be upgrades which change the amplitude/phase of a particular sinusoid, a prestige mechanic that increases your sample rate, bonuses for having harmonic signals etc. Would probably be easiest to create in Unity or with HTM5/CSS/JS

-

- **Effect Matcher:** I had an idea for a program that would take a song and find what sort of effects the instrument was using, and try to match the effects. This way you could potentially get a guitar to sound like the guitar in some other song. This is a kind of problem I encounter in my casual guitar playing.

- **Automated Clonehero/Guitar hero track maker**, There are plenty of games that let you make custom charts for tracks. I think it would be cool to make a tool that can automatically chart songs into a simple kind of sheet music. This is a video of how it is done manually Basics of Charting Part 3: Charting /

- **3D Audio Visualizer/Platform Game in Unreal Engine 4:** I would like to explore audio signal extraction in Unreal Engine 4 and attempt to create a 3D audio visualizer, and I think this would be a great opportunity to do some game design and figure out creative ways to make a video game using the audio samples. Right off the top of my head I figured we could create a platformer where we navigate a level that changes dynamically to the music.The minimum viable product would be a 3D Audio Visualizer, but I hope to extend beyond that and make a platform game with dynamically moving platforms based on the song that is playing.

- **Writing a short song by humming a melody:** The idea is that a user hums a random melody. After that the program analyzes the notes and identifies some chords. Then it proposes some chord progressions. The user selects 1 of them. The program converts the input (voice) into a melody played by an instrument (piano/guitar/flute/etc.) and the program combines the melody and the chords into a short complete song.

- **Virtual Concert with Landscape/Effects that Change to the Music**
  Due to COVID-19, virtual concerts have begun to be held as bigger events than ever before. Fortnite recently launched a series of virtual concerts called Soundwave after a successful virtual concert of a Japanese artist brought in many Japanese players. Roblox has hosted multiple virtual concerts, including big name artists such as Lil Nas X. Personally, the first virtual concert I attended was Secret Sky, hosted by Porter Robinson. The idea would be to create a beautiful landscape, like in Secret Sky, for the audience that changes in tune and emotion with the music. For a simpler version, perhaps creating a landscape view that users can watch while listening to music.

- **Make the music become happy or sad**: Basically, the program identifies chords, then makes them major/minor (probably there are other techniques). The program may need to change some of the chords to make the music sound smoothly. It also needs to change the tempo, and other aspects of the song, which we can do some more research on.

- **Guitar Tab Creation**
  Given monophonic audio as input, we'd like to employ machine learning to determine how a certain note is being played (because there are many ways to play a specific note

depending on the string and fret). The hope is then to generate a guitar tab, ideally all in real time (for the most part).

- **Advanced Guitar Tuner**: Often you tune each string, but when you play a chord it still sounds out of tune. Guitar tuners are good at tuning a single note, but do not account for the complexity of intonating an entire chord voicing, nor do they account for guitars with poor intonation. This program would isolate the notes/strings, so that the guitar could be tuned for an entire chord voicing. Additional functionality could include averaging multiple chords in a key, or different tuning systems.

- **Reverse Auto-Mixing**: This project will take as its input a finished stereo mix and raw multi-track audio, and reverse engineer the EQ, panning and level adjustment of the multi-track audio to recreate the stereo mix. This will be done either in reaper or we will make a simple daw environment.

- **Sound Source Separation**: This program would take songs with multiple instruments (vocals, drums, guitars, etc) and seperate the audio into its component parts. I am open to where this project could go, building upon previous work. Potential functionality could include: a chord voicing transcriber, instrument identifier, etc.

- **Chiptune Track Extender/Generator using Markov chains.** Takes as input a chiptune track (for simplicity), learning musical patterns, and creates similar but original sounds based on what it is trained on. Can be used to artificially extend songs or generate variations on a given track.

- **BPM Calculator:**
  This project would use song audio files and determine the beats per minute of the song. It could identify any changes in the beats per minute during the song. This project could be used to learn how to perform songs at the original speed or with the creation of a playlist to match up songs with similar beats per minute.

- **Phase Vocoder:**
  The simplest way to change the pitch of an audio recording is to change the playback speed. For a digital audio recording, this can be accomplished by sample rate conversion. Unfortunately, the frequencies in the recording are scaled at the same ratio as the speed, transposing its perceived pitch up or down in the process. Slowing down the recording to increase duration also lowers the pitch, speeding it up for a shorter duration also raises the pitch. Thus the two effects cannot be separated when using this method. A drum track containing no pitched instruments can be moderately sample-rate converted to adjust tempo without adverse effects, but a pitched track cannot.
  One way of stretching the length of a signal without affecting the pitch is to build a phase vocoder.
  Basic steps:

1. compute the instantaneous frequency/amplitude relationship of the signal using the STFT, which is the discrete Fourier transform of a short, overlapping and smoothly windowed block of samples;
2. apply some processing to the Fourier transform magnitudes and phases (like resampling the FFT blocks); and
3. perform an inverse STFT by taking the inverse Fourier transform on each chunk and adding the resulting waveform chunks, also called overlap and add (OLA).

- **Beginning Band Virtual Ensemble:**
  A program to try and emulate the "ensemble" experience for a remote beginning band (ex. middle school band class online due to covid). The first stage of the task would be to take several recordings, separate out background noise, combine them without "fixing" problems like pitch and tuning, and play it back to the students. The second stage would be to have the students record again while playing along to the previously assembled recording, subtract the previous recording from each student's new recording, and assemble the exercise again for playback. The idea is that the students will be able to play along with the previous iteration of the band's ensemble practice, with the assembled recording improving noticeably with each iteration as the students improve.

- **Chord Shift**
  Program takes in a song and modifies the chord progression by first identifying all unique chords in the song, then allows the user to specify a new chord for each identified chord. Would require isolating each instrument and modifying them. For songs with voice, it could possibly leave the voice unmodified to have interesting sounding mixes.
  - Extract individual instruments
  - Identify chords
  - Replace chords with given set by modifying existing chords

- **Chord Progression Generator**
  Program takes in a chord played by an instrument. Will recognize the chord and then identify what possible chords can be played next to continue a chord progression. Each new chord played will change the different chord possibilities possible in order to complete the progression.
  - Identify chords
  - Display possible chord progressions

- **Automated Transcription Program**
  A program to take audio input and inscribe it in standard notation or allow for export into music inscription software. In its most constrained form, would work for a single instrument and probably produce a MIDI, could be expanded to produce more complete and detailed transcriptions.

- **Melody maker assistant:**

The scenario starts with humming recognition. The user sings or hums a short melody, and the system should generate the rest. The First idea is to make a GAN system with an RNN (LSTM) as the generator. The Model can process the data in wave or MIDI form; right now I don't know which one is better and most available for this task. An advanced feature could be selecting the genre of music that the system should make.

- **Interval recognition teacher:**
  One of the primary jobs for a musician is to recognize and sing classic intervals in a major or minor. My teacher tried to help me with popular songs I learned when I was a child. He picked straightforward and the beginning of that music and coded them in my mind. The Model will ask the user to give it music pieces that the user knows very well (for example, jingle bells). Then the Model should recognize and separate the main melody. After that, it should pick the most significant intervals at the beginning of a verse. In the end, the system should tell the interval's name and give that part of music to the user to listen to it many times and be master in that interval.

- **Creative auto tuner:**
  This Idea is like the melody maker assistant from some aspects. But the main difference is that the user will say a sentence of a poem, and the system should generate a melody on that and autotune the first signing signal to the generated song.

- **Make a piano version of your favorite song:**
  The Idea is to find a different line of music and make a midi file from that, even the vocal. Then synthetic it with piano samples.

- **vocal deep-fake:**
  get a tiny sample of a person's voice and make a system to generate every speech with that person's voice. This video describes how we can build it. Also, this video mixed the previous one with a video rendering. Here are the papers:
  https://arxiv.org/abs/1806.04558
  https://arxiv.org/abs/1912.05566

- **Classification of Classical Persian Music in Different Dastgahs**
  The aim of this project is to identify the most popular different Dastgah (melody types) in Classical Persian Music. Classical Persian Music can be classified as one of the following famous Dastgahs: "Mahur", "Homayun", "Segah", "Chahargah", "Rast-Panjgah", "Shur". This classification can be made by extracting common acoustic features such as pitch, frequency, and identifying whether the music is using a major or minor scale.

  Here is a relevant paper about this. However, we are not using the same approach: https://www.researchgate.net/publication/220723522_Iranian_Traditional_Music_Dastgah_Classification

Here are some examples of different Dastgahs:
http://parhamnassehpoor.com/persian-music.html

- **Guitar to Midi with Effects**
  We want to be able to take audio signals from a guitar, preferably in real-time but potentially recorded, and convert it into other software instruments via MIDI. This system will need to identify monophonic melodies and chords to break down into separate pitches. It will also need to identify velocity (loudness) of the live instruments. Beyond this, we'd use different sound effects to change the characteristic of the sound (e.g. the direct sound is from the guitar, but the source of a reverb is from an organ).

- **Cover Songs Identification**
  Sometimes the cover of the song is different from the original one in terms of timbre, tempo, structure, or language of the song. From the user's perspective, finding a favorable version of a song can be enjoyable. This project aims to measure the similarity between cover songs and an original song by exploring shared music features in different versions.

- **Breaking each Persian Fusion Music into two Pieces of Music, one "Persian Classic", and the other one "Non-Persian Music"**
  Persian fusion music is a mixture of Classical Persian with Western music. The music contains Persian traditional instruments such as "santur", "tar", "setar" and "tanbur", etc. , and western instruments such as "piano", "guitar", "violin" ,etc. The idea behind this project is to break this genre of music into two genres by identifying instruments and genres. The two other generated pieces of music will include a piece of Persian Classical music, and the other genre of music will potentially be "Electric Music".

- **TempoRun**
  Simple Application: The application will extract the runners tempo using their phones accelerometer and attempt to match the tempo of the next song to the runners current tempo.
  Stretch Goal: Given a seed song the application will create a live playlist that adapts to the runners tempo. The application will extract the runners tempo using their phones accelerometer and attempt to match the tempo of the next song to the tempo of the runner, and use song recommendations to match the seed song.

- **Bird Caller ID**
  Monitoring of animal sounds is essential to accounting for climate change and environmental issues. Along with sight, sound can be used to observe certain bird species since there is a wide variety of distinct bird calls and songs. Simple to complex implementation of this idea can be done through iterative development. Firstly, raw audio of bird sounds can be processed and compared to pre-existing libraries. Further implementation may include estimation of distance and location, more complex audio

(environment), and approximate genre classification of bird sounds (for interest). Conversely, sinusoids may be synthesized to mimic bird sounds and pass through the recognizer.

- **Sheet Music to Chiptunes:**
  This tool will take as input, sheet music and play the song in a chiptunes engine. A OCR scan will be used to identify the music contents and stored in a format such XML, then passed into a chiptunes engine.

- **Style transfer:** I am thinking of using a technique called style transfer convert a song to another genre using convolutional neural networks. This is normally done with images but there has been previous work done on convolutional networks applied to audio, using time as one of the dimensions. Here is an explanation if you're interested: https://youtu.be/6wcs6szJWMY?t=4058

- **Automated harmonizer**: Basic idea: you sing to the computer, and it plays your recording back with harmony that it created by itself. Thus, it would need to identify the tempo and key of your song (at minimum) and make up appropriate chord progressions based on that. Not entirely sure how feasible this is, but it's something I'd be willing to invest a lot of time in to get some kinda version of it off the ground.

- **Smart Sampler:** Given waveform data, say from a jazz saxophone, classify into midi data to create a midi instrument plugin. The goal is not to accurately recreate an instrument, but to create an "artistically interesting" midi instrument that one might use in their production. For example, given 10 isolated Jimi Hendrix guitar tracks, an instrument would be automatically created through waveform analysis attempting to identify notes and chords.

- **Automatic Music Generation with 1st and 2nd order Markov Chain Model:**
  Input: IBM Watson Beats MIDI (https://github.com/cognitive-catalyst/watson-beat , style transferred MIDI data based on reinforcement learning and neural networks) or we could extend this open source project if it is possible.
  Output(Expectations):
  1. MIDI data via 1st and 2nd order Markov Chain Model (They have different characteristics with music applications) using Music 21 python library.
  2. Put the MIDI data on Music Software or Language and verify it.
  3. Verify it with Electronic drum kits or Synthesizer

  If it is related to automatic generation in music, I am very flexible and open-minded to change the specific topic or interest. Computer Music students might be more interested in this project since it could be potentially related to MUS407 project; however, this project is opened for everyone. I did my musical project with Max/Msp which was Beethoven moonlight sonata 1st movement melody style transfer by 2nd order Markov Chain Model. I sincerely hope to meet students who have similar interests

- **Interactive Visualizer/Synthesizer using Unity Engine:**
  Apologies if my thoughts are vague, but I figured I should at least put down this concept and see who's interested. This project breaks down into two distinct ideas:
  1. A Unity application which reacts dynamically to an audio signal, visualizing the audio is an aesthetically pleasing way (in 3D if possible).
  2. An interactive unity application in which the user's actions can synthesize audio, or alter a given audio signal.

  More interesting than either, though, would be to combine both concepts into a single application, such as unity "game" in which audio input is visualized as part of a 3D environment, and by interacting with said environment the user can alter the output audio. In essence, a playable, immersive audio plugin.

  This project would suit those with an interest in computer graphics and DSP, and some familiarity with the Unity engine. Please contact me if interested- this idea is malleable and could use some additional ideas.

- **Creating a musical "trailer" for a set of songs:**
  The general idea is to take a set of songs, whether it be an album or the Billboard Top 40, and create a single song that combines parts of each song in the set into a cohesive musical structure. By analyzing the tempo, harmonics and melodies, I hope to be able to time stretch and pitch shift the songs in the set and have an intelligent method of stitching them together.

- **Music Visualiser with Photographs:**
  The main idea is to create a visualiser that will animate any given photograph with the style of the music. Now the animation should be something subtle and elegant, instead of making objects jump or something dramatic, i.e. in a photograph of a landscape, detecting the edges and animate only those, leaving the rest static.
  How to achieve this is still unsure, but I can use image convolutions with different filters to achieve the effects. Now the main goal will be to match the filters to the music, knowing how to move/animate those filters depending on the music and knowing which parts of the photograph to animate.

- **Language identification in Songs:**
  This project will involve reading in songs from various languages and attempting to train and algorithm to be able to identify the language(s) used in the song. This will involve using various AI techniques in order to solve the problem

- **Information Hiding:**
  Watermarking, fingerprinting, steganography and other information hiding techniques. There are many algorithms like LSB so that the challenge is to understand and achieve them, with what we learned in the class. I prefer using Python or C#. Python is more

easy based on what we learned in the class while C# is easy to generate an interface. I have some experience in information hiding but never tried it on audio.

- **Automatic Beat Generation:**
  The end product, I imagine, is a tool that will take as input a musical sample of some sort (without drums) and then output that same sample with drums correctly placed overtop of it. This will involve beat tracking and/or tempo induction to find where the drums need to be placed. The drums could be sets of midi drum samples. There could be numerous sets of drum samples corresponding to different genres of music so that one input sample could produce multiple unique beats in different styles (hip-hop, lofi, trap, rock, etc..). Additional work could include modifying the original sample, such as changing the pitch and tempo, to help it match more with the selected genre.

- **Choose-Your-Own Karaoke:**
  The goal is to remove (or lessen) specific components of a song (e.g. Vocals, Guitar, Percussion) by identifying and isolating the wanted frequencies. Specific instruments should be singled out and removed. Left/Right Audio cancelling or other removal techniques could be used to edit the tracks. This could be used to practice a chosen instrument along with the song to help with learning and playing music.

- **K.K. Slider Sings:**
  K.K. Slider is a singing and guitar-playing dog from the video game series Animal Crossing. He possesses a unique voice that is recognizable to many. Currently, there exists a niche genre of videos that includes swapping popular song vocals with vocals from K.K. Slider, often times amassing hundreds of thousands of views. More on K.K. Slider can be found in this article, including his significance in popular internet culture https://kotaku.com/k-k-slider-is-the-most-influential-musician-of-our-gen-1837746755
  The idea of this project would be to automate the process of swapping vocals in a song with another voice. Since K.K. Slider would only sing melody and not lyrics, I think it would be easier to execute. Likewise, there exist samples on the internet for this character's voice, so getting all the correct pitches would not be difficult. The idea seems silly on the surface, but I think it has a lot of potential.

- **De-Sampler:**
  The idea here is to create a tool that would operate similarly to Shazam but for identifying samples in a song. An example of this would be a hip-hop track that incorporates vocal samples from a soul song. The tool would analyze the hip-hop song and output the name of the soul song that the hip-hop song sampled. The scope of this project would have to be limited to a specific set of songs and the samples would need to be fairly recognizable (i.e. not too distorted, reversed, chopped up, etc).

- **Isolating instruments from music using deep learning:**

The intention of this project is to build an instrument learning/covering software that can mute a specific track (instruments or vocals), so players can play along with the rest of the music. For identifying and isolating the components of a piece of music, some AI techniques are required (e.g. Convolutional Neural Networks), so the implementation will be based on Python. We will start with one distinctive component (drum or vocal), and refer to some existing projects.

- **Genre/tag prediction based on an existing set of tracks and user-inputted tags:**
  Using a large set of existing tracks with user-inputted tags we hope to develop a tool that will suggest an applicable tag for new songs using the existing track/tag pairs as a training model. As this concept has been pursued before we also are considering using different methods and implementations from other studies and comparing the results. It might be interesting to compare tags suggested based on a model of user-inputted tags vs tags suggested based on different tag-predicting algorithm as well.

- **Emotion Classification:**
  Previously in one of my classesI worked on a project on emotion classification using the ravdees dataset. My team and I used an SVM to correctly identify the speaker emotion 77% of the time. In this project I would like to continue where my team and I left of previously and use different machine learning algorithms to see if we can improve the classification accuracy.

- **SP1 - Browsing and classification of Orca vocalizations**
  Data:
  I can provide access to a subset of the Orchive

  Code:
  https://github.com/ChristianBergler/ORCA-SLANG
  https://www.youtube.com/watch?v=mOWFjOkhqtw

  Papers:
  Ness, Steven, et al. "The Orchive: Data mining a massive bioacoustic archive." *arXiv preprint arXiv:1307.0589* (2013).
  Bergler, Christian, et al. "ORCA-SPOT: An automatic killer whale sound detection toolkit using deep learning." *Scientific reports* 9.1 (2019): 10997.
  Bergler, Christian, et al. "ORCA-SLANG: An Automatic Multi-Stage Semi-Supervised Deep Learning Framework for Large-Scale Killer Whale Call Type Identification." *Interspeech*. 2021.

- **SP2 - Autotune presence detection**
  Datasets:
  Smule multi-lingual vocal performances:

https://zenodo.org/records/2747436
Smule amazing grace:
https://zenodo.org/records/3596940

Code:
https://github.com/gtzan/mir_program_kadenze/tree/master/course1/session3
https://github.com/ederwander/PyAutoTune
https://thewolfsound.com/how-to-auto-tune-your-voice-with-python/

Papers:
Wager, Sanna, et al. "Deep autotuner: A pitch correcting network for singing performances." *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020.
Provenzano, Catherine. "Making voices: The gendering of pitch correction and the auto-tune effect in contemporary pop music." *Journal of Popular Music Studies* 31.2 (2019): 63-84.
Schwär, Simon J., Sebastian Rosenzweig, and Meinard Müller. "A Differentiable Cost Measure for Intonation Processing in Polyphonic Music." *ISMIR*. 2021.

- **SP3 - Real-time guitar tuner:**
  Datasets:
  https://www.idmt.fraunhofer.de/en/publications/datasets/guitar.html
  https://research.google.com/audioset/ontology/guitar_1.html

  Code:
  https://librosa.org/doc/main/generated/librosa.pyin.html
  https://github.com/logsol/tuner
  https://juce.com/

  Papers:
  Murphy, J. W., Mathews, P., Kapur, A., & Carnegie, D. A. (2014). Robot: Tune Yourself! Automatic Tuning for Musical Robotics. In *NIME* (pp. 565-568).
  Mauch, Matthias, and Simon Dixon. "pYIN: A fundamental frequency estimator using probabilistic threshold distributions." *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014.

- **SP4 - Text to setting guitar effects**:
  Datasets:
  https://www.idmt.fraunhofer.de/en/publications/datasets/audio_effects.html
  https://egfxset.github.io/

  Code:
  https://github.com/spotify/pedalboard

Papers:
Cartwright M, Pardo B. Social-EQ: Crowdsourcing an Equalization Descriptor Map. In ISMIR 2013 Nov 4 (pp. 395-400).
https://zenodo.org/records/7817839
Comunità M, Stowell D, Reiss JD. Guitar effects recognition and parameter estimation with convolutional neural networks. arXiv preprint arXiv:2012.03216. 2020 Dec 6.

- **SP5 - Automated DJing**

Datasets:
https://aistdancedb.ongaaccel.jp/
https://google.github.io/aistplusplus_dataset/download.html
https://github.com/GiantSteps/giantsteps-tempo-dataset
https://zenodo.org/records/1101082

Code:
https://github.com/Gariscat/loopy
https://github.com/jiaaro/pydub

Papers:
Kell, Thor, and George Tzanetakis. "Empirical Analysis of Track Selection and Ordering in Electronic Dance Music using Audio Feature Extraction." *ISMIR*. 2013.
Tsuchida S, Fukayama S, Hamasaki M, Goto M. AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing. InISMIR 2019 Nov (Vol. 1, No. 5, p. 6).
Knees, P., Faraldo Pérez, Á., Boyer, H., Vogl, R., Böck, S., Hörschläger, F. and Le Goff, M., 2015. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR); 2015 Oct 26-30; Málaga, Spain.[Málaga]: International Society for Music Information Retrieval, 2015. p. 364-70.*. International Society for Music Information Retrieval (ISMIR).
Bittner RM, Gu M, Hernandez G, Humphrey EJ, Jehan T, McCurry H, Montecchio N. Automatic Playlist Sequencing and Transitions. InISMIR 2017 Oct 23 (pp. 442-448).
Yadati K, Larson MA, Liem CC, Hanjalic A. Detecting Drops in Electronic Dance Music: Content based approaches to a socially significant music event. InISMIR 2014 Oct 27 (pp. 143-148).

- **SP6 - Guitar Assistant**

Datasets:
https://www.idmt.fraunhofer.de/en/publications/datasets/guitar.html
https://research.google.com/audioset/ontology/guitar_1.html

Code:
https://github.com/jackmcarthur/musical-key-finder

https://www.justinsalamon.com/codedata.html

Papers:

Chuan CH, Chew E. Polyphonic audio key finding using the spiral array CEG algorithm. In2005 IEEE International Conference on Multimedia and Expo 2005 Jul 6 (pp. 21-24). IEEE.
Papadopoulos, Hélene, and Geoffroy Peeters. "Local key estimation from an audio signal relying on harmonic and metrical structures." *IEEE Transactions on Audio, Speech, and Language Processing* 20.4 (2011): 1297-1312.
Weiß C, Schreiber H, Müller M. Local key estimation in music recordings: A case study across songs, versions, and annotators. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2020 Oct 13;28:2919-32.
Kaliakatsos-Papakostas, Maximos A., Andreas Floros, and Michael N. Vrahatis. "A clustering strategy for the key segmentation of musical audio." *Computer Music Journal* 37.1 (2013): 52-69.
Abeßer J, Schuller G. Instrument-centered music transcription of solo bass guitar recordings. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017 May 8;25(9):1741-50.
Kumar R, Biswas A, Roy P. Melody extraction from music: A comprehensive study. Applications of Machine Learning. 2020:141-55.
Salamon, Justin, and Emilia Gómez. "Melody extraction from polyphonic music signals using pitch contour characteristics." *IEEE transactions on audio, speech, and language processing* 20.6 (2012): 1759-1770.

- **SP7 - Beat/melody audio auto-tagging:**

  Datasets:
  FMA - https://github.com/mdeff/fma
  MTG Jamendo: https://mtg.github.io/mtg-jamendo-dataset/
  Music4All https://sites.google.com/view/contact4music4all

  Code:
  https://www.justinsalamon.com/codedata.html
  https://github.com/CPJKU/madmom
  https://scikit-learn.org/stable/modules/multiclass.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.
  Böck S, Korzeniowski F, Schlüter J, Krebs F, Widmer G. Madmom: A new python audio and music signal processing library. InProceedings of the 24th ACM international conference on Multimedia 2016 Oct 1 (pp. 1174-1178).
  Salamon, Justin, and Emilia Gómez. "Melody extraction from polyphonic music signals using pitch contour characteristics." *IEEE transactions on audio, speech, and language processing* 20.6 (2012): 1759-1770.
  Kim, Jong Wook, et al. "Crepe: A convolutional representation for pitch estimation." *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018

Choi, Keunwoo, et al. "Convolutional recurrent neural networks for music classification." *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017.

- **SP8 - Music Generation with deep learning**
  Datasets:
  https://magenta.tensorflow.org/datasets

  Code:
  https://magenta.tensorflow.org/
  https://colab.research.google.com/notebooks/magenta/gansynth/gansynth_demo.ipynb

  Resources:
  https://philippepasquier.com/systems

  Papers:
  Huang CZ, Vaswani A, Uszkoreit J, Shazeer N, Simon I, Hawthorne C, Dai AM, Hoffman MD, Dinculescu M, Eck D. Music transformer. arXiv preprint arXiv:1809.04281. 2018 Sep 12.
  Hawthorne C, Stasyuk A, Roberts A, Simon I, Huang CZ, Dieleman S, Elsen E, Engel J, Eck D. Enabling factorized piano music modeling and generation with the MAESTRO dataset. arXiv preprint arXiv:1810.12247. 2018 Oct 29.

- **SP9 - Track Splitting Spectrograph Visualizer in Godot**
  Datasets:
  https://freesound.org/

  Code:
  https://godotengine.org/
  https://docs.godotengine.org/en/stable/tutorials/audio/index.html

  Papers:
  Stevens, Richard, and Dave Raybould. *Game audio implementation: a practical guide using the unreal engine*. CRC Press, 2015.
  Haas, John K. "A history of the unity game engine." *Diss. Worcester Polytechnic Institute* 483.2014 (2014): 484.
  Resch, Thomas. "Rwa-a game engine for real world audio games." *NIME*. 2015.

- **SP-10 Guitar audio to tab generator**
  Datasets:
  https://www.idmt.fraunhofer.de/en/publications/datasets/guitar.html
  https://research.google.com/audioset/ontology/guitar_1.html

Code:
https://github.com/jackmcarthur/musical-key-finder
https://www.justinsalamon.com/codedata.html

Papers:
Kehling, Christian, et al. "Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score-and Instrument-Related Parameters." *DAFx*. 2014.
Chen, Yu-Hua, et al. "Automatic composition of guitar tabs by transformers and groove modeling." *arXiv preprint arXiv:2008.01431* (2020).
Bittner, Rachel M., et al. "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation." *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

- **SP-11 Deep learning for perceptual audio decompression**

  Datasets:
  Any collection of audio/music files

  Code/Documentation:
  https://xiph.org/flac/
  https://github.com/facebookresearch/encodec

  Papers:
  Rim, Daniela N., Inseon Jang, and Heeyoul Choi. "Deep neural networks and end-to-end learning for audio compression." *arXiv preprint arXiv:2105.11681* (2021).
  Défossez, Alexandre, et al. "High fidelity neural audio compression." *arXiv preprint arXiv:2210.13438* (2022).
  Wu, Yi-Chiao, et al. "Audiodec: An Open-Source Streaming High-Fidelity Neural Audio Codec." *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.
  Yang, Runxuan, Yuyang Peng, and Xiaolin Hu. "A fast high-fidelity source-filter vocoder with lightweight neural modules." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).
  Zeghidour, Neil, et al. "Soundstream: An end-to-end neural audio codec." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021): 495-507.

- **SP-12 Deep fake: voice cloning**

  Datasets:
  https://librivox.org/
  https://paperswithcode.com/dataset/deep-voice-deepfake-voice-recognition

  Code:
  https://benaandrew.github.io/Voice-Cloning-App/

Papers:
Arik, Sercan, et al. "Neural voice cloning with a few samples." *Advances in neural information processing systems* 31 (2018).
Jemine, Corentin. "Real-time-voice-cloning." *University of Liége, Liége, Belgium* (2019): 3.
Tan, Xu, et al. "A survey on neural speech synthesis." *arXiv preprint arXiv:2106.15561* (2021).
Li, Naihan, et al. "Neural speech synthesis with transformer network." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

- **SP-13 Detecting AI generated audio**
  Datasets:
  https://ai.meta.com/datasets/dfdc/
  https://www.kaggle.com/c/deepfake-detection-challenge
  https://keithito.com/LJ-Speech-Dataset/
  https://zenodo.org/records/5642694

  Code/Resources:
  https://github.com/RUB-SysSec/WaveFake
  https://github.com/MarkHershey/AudioDeepFakeDetection?tab=readme-ov-file

  Papers:
  Khalid, Hasam, et al. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset." *arXiv preprint arXiv:2108.05080* (2021).Khalid, Hasam, et al. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset." *arXiv preprint arXiv:2108.05080* (2021).
  Wijethunga, R. L. M. A. P. C., et al. "Deepfake audio detection: a deep learning based solution for group conversations." *2020 2nd International Conference on Advancements in Computing (ICAC)*. Vol. 1. IEEE, 2020.
  Khanjani, Zahra, Gabrielle Watson, and Vandana P. Janeja. "How deep are the fakes? focusing on audio deepfake: A survey." *arXiv preprint arXiv:2111.14203* (2021).
  Hamza, Ameer, et al. "Deepfake audio detection via MFCC features using machine learning." *IEEE Access* 10 (2022): 134018-134028.

- **SP-14 - Creating/manipulating audio with controller**
  Datasets:
  https://freesound.org/

  Code/Resources:
  https://chuck.stanford.edu/
  https://supercollider.github.io/
  https://cycling74.com/products/max
  https://puredata.info/

  Papers:

Wessel, David, and Matthew Wright. "Problems and prospects for intimate musical control of computers." *Computer music journal* 26.3 (2002): 11-22.
Serafin, Stefania, et al. "Virtual reality musical instruments: State of the art, design principles, and future directions." *Computer Music Journal* 40.3 (2016): 22-40.
Miranda, Eduardo Reck, and Marcelo M. Wanderley. *New digital musical instruments: control and interaction beyond the keyboard*. Vol. 21. AR Editions, Inc., 2006.
Cook, Perry. "2001: Principles for designing computer music controllers." *A NIME Reader: Fifteen years of new interfaces for musical expression* (2017): 1-13.

- **SP-15 - MIR library in Go**
  Datasets: Any dataset with audio recordings
  https://freesound.org/

  Code/Resources:
  https://librosa.org/doc/latest/index.html
  https://pytorch.org/audio/stable/index.html
  https://pypi.org/project/madmom/
  https://go.dev/

  Papers:
  https://conference.scipy.org/proceedings/scipy2015/pdfs/brian_mcfee.pdf
  Yang, Yao-Yuan, et al. "Torchaudio: Building blocks for audio and speech processing." *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.
  Böck, Sebastian, et al. "Madmom: A new python audio and music signal processing library." *Proceedings of the 24th ACM international conference on Multimedia*. 2016.

- **SP-16 - Visualization of Music Based using the Unreal Engine**
  Datasets:
  Any dataset with audio recordings
  https://freesound.org/
  https://mtg.github.io/mtg-jamendo-dataset/

  Code/Resources:
  https://www.youtube.com/watch?v=3JYyNucyMBk
  https://docs.unrealengine.com/5.0/en-US/working-with-audio-in-unreal-engine/

  Papers:
  Masullo, Massimiliano, Hasan Baran Fırat, and Luigi Maffei. "Virtual acoustics with game engines." *Proceedings of ICSV25*. International Institute of Acoustics and Vibration IIAV, 2018.
  Atherton, Jack, and Ge Wang. "Chunity: Integrated Audiovisual Programming in Unity." *NIME*. 2018.

Holloway, Alexandra, et al. "Visualizing audio in a first-person shooter with directional sound display." *1st Workshop on Game Accessibility: Xtreme Interaction Design (GAXID'11) at Foundations of Digital Games, Bordeaux, France*. 2011

- **SP-17 - Real-time Interactive Music Accompaniment System**

- **SP-18 - Extending Songs**
  Datasets:
  https://sites.google.com/view/contact4music4all
  https://github.com/mdeff/fma
  use audio tracks by breaking them down into multiple time segments and predict what follows

  Papers:
  Polyphonic music generation generative adversarial network with Markov decision process:
  https://link.springer.com/article/10.1007/s11042-022-12925-w
  Controlling Perceived Emotion in Symbolic Music Generation with Monte Carlo Tree Search:
  https://arxiv.org/abs/2208.05162