Project Proposal: Deep Music Source Separation

Winter 2026 Final Project

Team Proposal

1 Executive Summary

Instead of selecting one of the three pre-defined options (Pose Estimation, Hallucinations, Inpainting), we propose an **"Alternative Project"** focused on **Music Source Separation (MSS)**.

The Goal: To take a mixed music track (a "mixture") and use Deep Learning to extract a single instrument stem (e.g., piano) from it.

The Research Angle: We will implement and compare two distinct deep learning paradigms:

1. **Frequency Domain (The "Image" Approach):** Treating audio as a 2D Spectrogram and using computer vision techniques (U-Net).
2. **Time Domain (The "Waveform" Approach):** Treating audio as a 1D signal and using raw audio convolutions (Demucs architecture).

This project fulfills the course requirement to "train a model... working with a non-trivial amount of data" and extends learned material in a "research context".

2 Methodology & Architectures

We will implement two different models to solve the same problem, allowing us to perform the comparative analysis required for a high-grade report.

2.1 Approach A: Frequency Domain (Baseline)

- **Concept:** The audio waveform is transformed into a time–frequency representation using the **Short-Time Fourier Transform (STFT)**. The complex STFT coefficients are converted into a **magnitude spectrogram**, which is treated as a 2D image. Source separation is then formulated as an image segmentation problem over the time–frequency plane.
- **Signal Representation:** Although the STFT is complex-valued and contains both magnitude and phase information, only the **magnitude** is provided as input to the neural network. The phase is discarded during learning, simplifying the representation but making audio reconstruction ill-posed.
- **Architecture: 2D U-Net.**
  - **Input:** Magnitude spectrogram of the mixture signal ($Time \times Frequency$).
  - **Encoder:** A sequence of 2D convolutional layers with downsampling extracts hierarchical spectral features, such as harmonic structures and instrument-specific frequency patterns.

  - **Decoder:** Symmetric upsampling layers reconstruct the time–frequency resolution. Skip connections between encoder and decoder layers preserve fine-grained spectral details.
  - **Output:** A real-valued **soft time–frequency mask** with values in $[0, 1]$, indicating the contribution of the target source in each time–frequency bin.
- **Source Estimation and Reconstruction:** The predicted mask is applied element-wise to the mixture magnitude spectrogram to estimate the target magnitude. Audio reconstruction is performed using the **mixture phase**, followed by an inverse STFT.
- **Training Objective:** The network is trained in a supervised manner by minimizing an L_1 loss between the predicted target magnitude spectrogram and the ground-truth magnitude spectrogram of the isolated source.
- **Pros:** Interpretable representation; stable training; leverages well-established CNN architectures.
- **Cons:** Phase information is discarded during learning. Reusing or approximating the phase during reconstruction introduces audible artifacts, especially in regions where multiple sources overlap in frequency.

2.2 Approach B: Time Domain (Main Contribution)

- **Concept:** Instead of relying on a fixed time–frequency transform, the model operates directly on the raw audio waveform. This allows the network to learn an internal representation optimized for the source separation task while preserving all temporal and phase information.
- **Architecture: Simplified Demucs.**
  - **Input:** Raw waveform tensor with shape `[Batch, Channels, Time]`.
  - **Encoder:** A stack of strided 1D convolutional layers progressively downsamples the signal in time. These layers act as learnable band-pass filters, effectively learning a task-specific time–frequency decomposition.
  - **Bottleneck:** A recurrent module (LSTM) operates at the lowest temporal resolution, capturing long-range temporal dependencies such as rhythm, instrument continuity, and musical structure.
  - **Decoder:** Transposed convolutions and upsampling layers reconstruct the waveform. Skip connections from the encoder ensure accurate recovery of fine temporal details.
- **Training Objective:** The model is trained end-to-end by minimizing an L_1 loss directly on the waveform between the predicted signal and the ground-truth target source.
- **Pros:** State-of-the-Art (SOTA) audio quality; phase information is handled implicitly; avoids explicit signal reconstruction steps and associated artifacts.

3 Dataset & Training Strategy

We will use **Supervised Learning**, which creates perfect "Ground Truth" data efficiently.

3.1 Dataset: MUSDB18

This is the standard academic dataset for Music Source Separation. It contains ∼150 professionally recorded tracks. Unlike standard audio files, these tracks are provided as **multi-track**

stems, meaning the Vocals, Drums, Bass, and Other (Piano/Synth) are stored as separate, perfectly aligned audio channels. This allows us to construct perfect ground-truth targets (Y) and inputs (X).

3.2 Data Augmentation & Curriculum Strategy

To improve generalization and robustness, we will implement a dynamic data augmentation pipeline:

- **Random Mixing (Remixing):** Instead of training only on the original songs, we will generate synthetic mixtures on-the-fly by summing stems from *different* tracks (e.g., combining "Drums from Song A" with "Piano from Song B"). This prevents the model from memorizing specific song structures and forces it to learn instrument timbres.
- **Curriculum Learning:** We will experiment with a progressive training difficulty schedule:
  - **Stage 1 (Simple):** The model trains on mixtures of only 2 sources (e.g., Target Instrument + 1 Distractor).
  - **Stage 2 (Intermediate):** We increase complexity to 3 sources (Target + 2 Distractors).
  - **Stage 3 (Full):** The model trains on the full mixture (Target + All Distractors).

  This approach allows the model to learn basic separation filters before tackling complex, dense audio mixtures.

4 Work Plan & Report Structure

The course guidelines state that alternative projects are "not recommended" unless the students are "highly driven" and provide a detailed "Related Work" section.

Proposed Report Sections:

1. **Introduction:** Define the "Cocktail Party Problem" and the shift from Spectral to Waveform modeling.
2. **Related Work:** Review key papers (Open-Unmix for spectral, Wave-U-Net for 1D convs, Demucs for SOTA).
3. **Methodology:** Detail our implementation of the 1D Convolution blocks and the U-Net, including the Curriculum Learning pipeline.
4. **Experiments:**
   - **Quantitative:** Measure Signal-to-Distortion Ratio (SDR) on a held-out test set.
   - **Qualitative:** Visual comparison of Spectrograms and listening tests.
5. **Work Report:** Documentation of code written and challenges faced.

5 Feasibility & Logistics

- **Submission Deadline:** 5.3.2026.
- **Libraries:** PyTorch (core), `musdb` (data loading), `torchaudio` (transforms).

- **Compute Strategy:** Audio models can be memory-intensive. We will likely downsample audio to 16kHz or 22kHz to speed up training on standard GPUs.
- **Why this works:** It combines **Fourier Analysis** (STFT), **Linear Algebra** (Matrix operations), and **Deep Learning** (RNNs/CNNs), leveraging our existing academic strengths.