

Learning Music Similarity Embeddings for MECOMP



Anthony Rubick, Beyza Ispir, Joseph Berglund

COMP/ELEC 576 - Introduction to Deep Learning - 2025-2026

Abstract

We wanted to build a deep learning model that learns vector embeddings for songs. These embeddings should match how humans think two songs sound similar. The goal is to replace or improve the handcrafted features inside MECOMP, a music player and recommendation tool written in Rust. We will use short audio clips from Creative Commons songs and human similarity triplets for training. The final model should be small, fast, and able to process a local music library quickly on a normal laptop.

Traditional:
Audio → Classifier
Fixed-size
(not for similarity)

Our Approach:
Audio → Embedding
Variable-length
(optimized for similarity)

Background and Motivation

Problem Statement

Music recommendation systems need good embeddings to find similar songs, but current methods have some big problems:

- Most embeddings (like wav2vec) are built for classifying individual songs, not comparing them to each other
- They often need fixed-size inputs, which doesn't work well for songs of different lengths
- Some models are trained on just one genre, so they don't work across different types of music
- Previous work (like the Bliss project) trains separate distance functions instead of making the embeddings themselves better at expressing similarity

Why This Matters

Recommendation is super important for music apps, but right now the methods miss what makes songs feel similar to humans. We're training embeddings specifically to capture how songs compare to each other, so recommendations can be based on how music actually "feels" rather than just metadata tags.

What Makes This New

This is a fresh take on music similarity with deep learning:

1. We train embeddings directly for similarity (not classification)
2. We use human judgments with triplet loss to match how people actually hear music
3. Our model handles variable-length audio, which is what you get in real music libraries
4. It's designed to work with MECOMP, an existing Rust-based music player

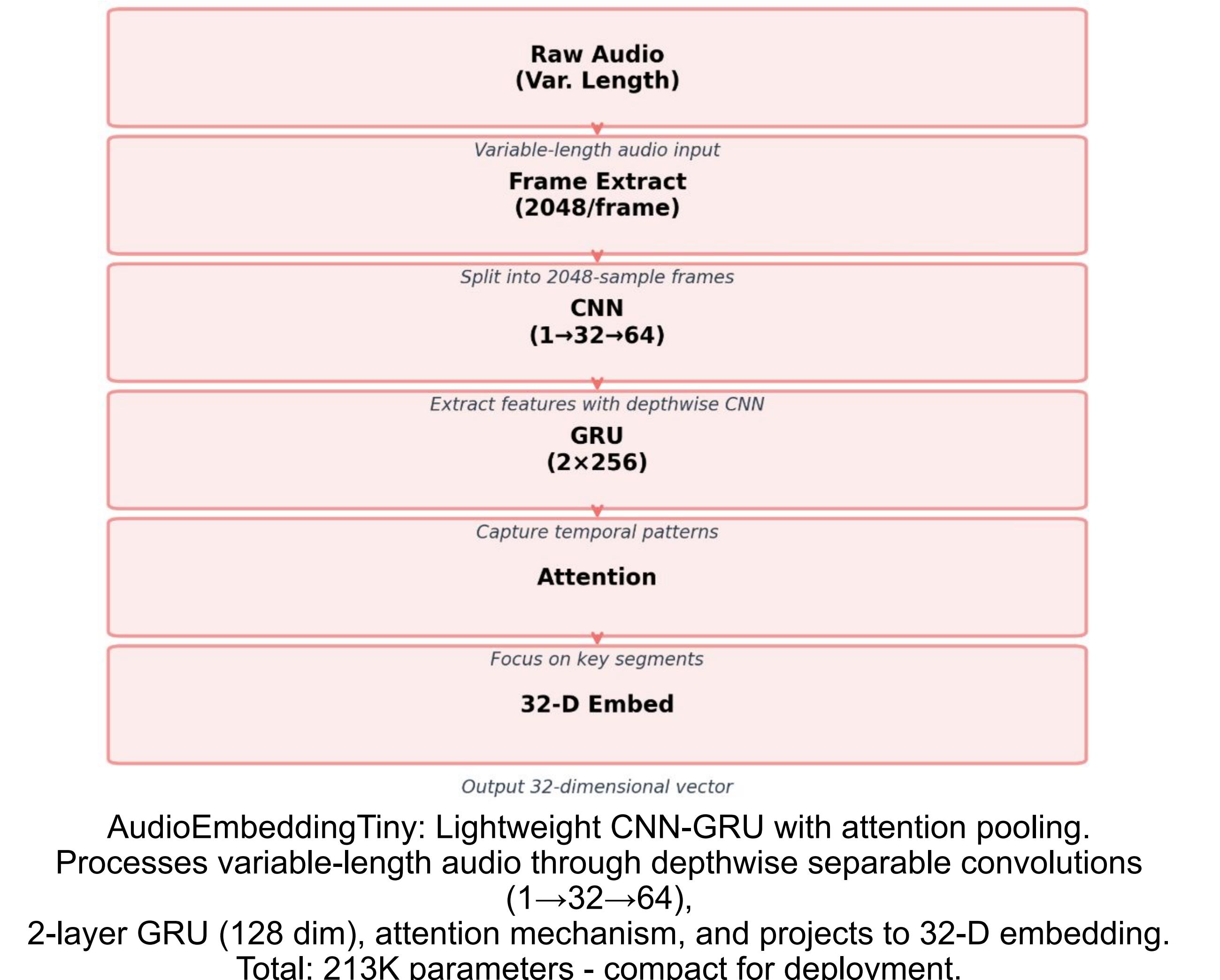
Dataset

We used the Free Music Archive (FMA) Small Dataset:

- 8,000 songs covering evenly distributed across 8 genres
- 8,000 synthetic triplets we generated using genre info
- 2,000 human-annotated triplets (60 fully labeled)
- Audio format: 22.05 kHz mono, clips are 30 seconds long (though to save memory during training, only the first 10-20 seconds are used)

Hyperparameters: Frame=2048 | Embed=32-D | Batch=12→4 | GRU=2×128 | Loss=margin 0.2

Model



Results

