

Team Members

Reggie Bain

Emelie Curl

Larsen Linov

Tong Shan

Glenn Young

**“Good Composers
Borrow, Great
Ones Steal”**

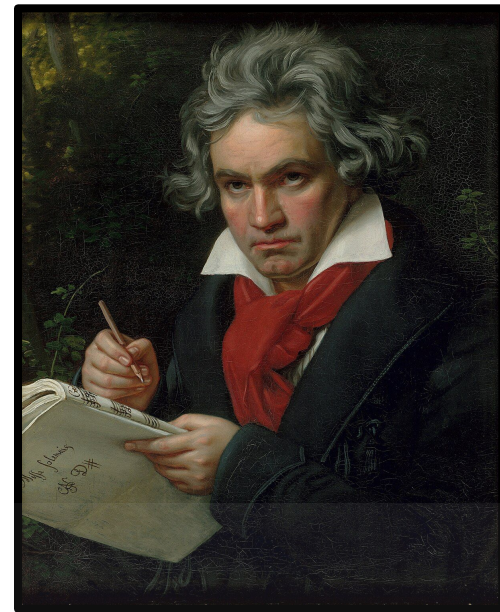


**Erdos Institute Deep
Learning Boot Camp
Summer 2024**



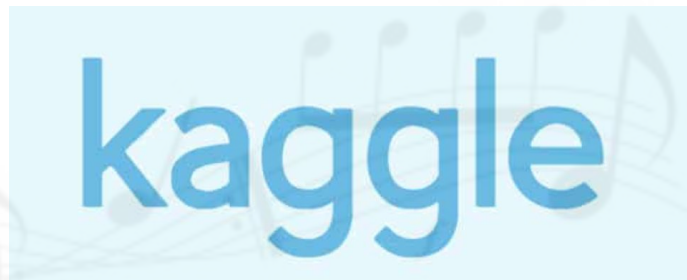
Motivation: Musical borrowing (i.e. covers, sampling, unconscious plagiarism, intentional copying, etc.) has a long history, spanning classical music of Mozart and Bach all the way up to the music we would consider “modern” involving artists such as Ed Sheeran, Katy Perry, and Olivia Rodrigo.

Goal: Use deep learning to detect when two music clips are very similar



Our Data Sources

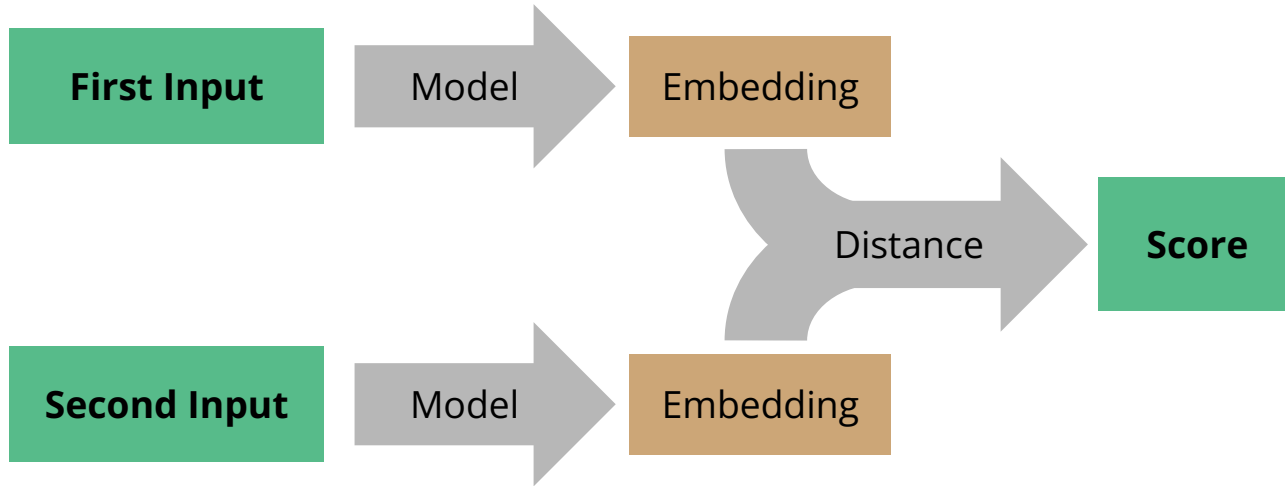
- We used data from a variety of sources based around the Million Song Dataset (MSD).
- Additionally, various online APIs were used to gather available audio samples.
- The audio samples we were able to obtain were also converted to images which became our primary source for training models.



Siamese Neural Networks

We used a *Siamese Neural Network* structure to measure similarity.

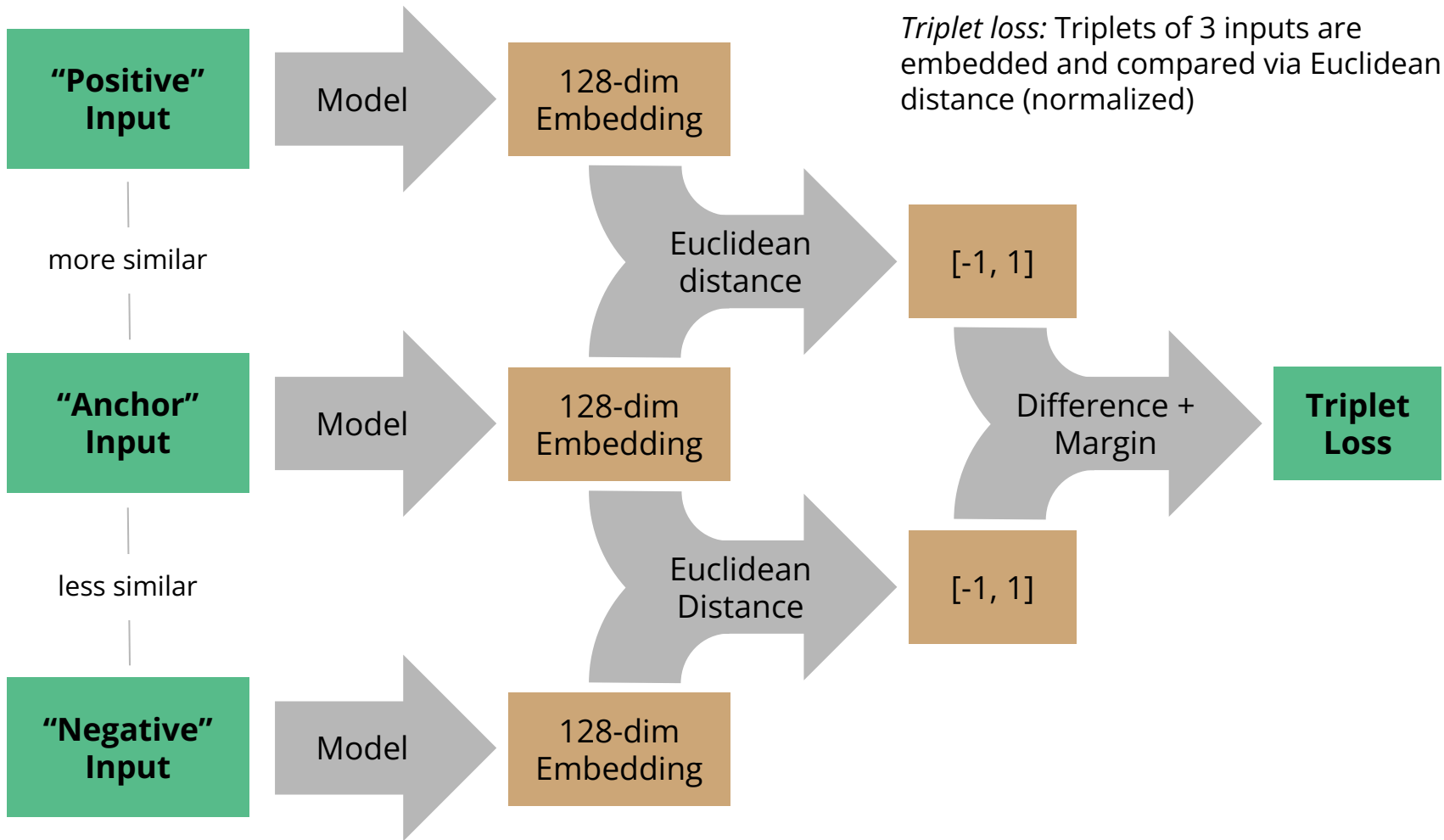
A SNN embeds its inputs into a high-dimensional Euclidean space, where they can then be compared by distance or cosine similarity.



Training with Triplet Loss

- We used **three** inputs at a time (“anchor”, “positive”, “negative”), with the *triplet loss function*, to create meaningful embeddings for comparison.
- Triplet loss aims to *minimize the distance between similar inputs* and *maximize distance between different inputs*

$$\mathcal{L}(A, P, N) = \max(\|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 + \alpha, 0)$$

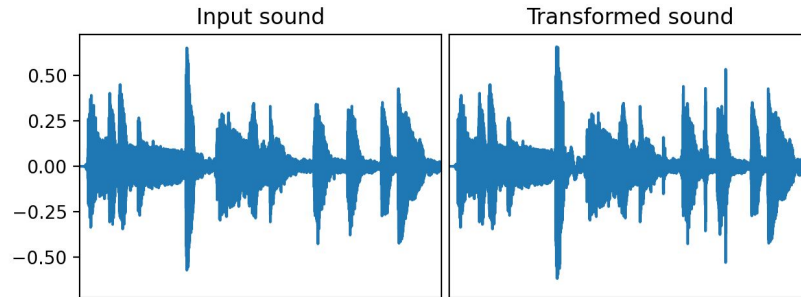


Data Augmentation

We needed pairs of audio clips that are the “same song” even if they don’t sound exactly the same.

Starting with original music files, we generated similar ones via audio data augmentation (e.g. pitch shift, adding background noise, time distortion).

Python library: `audiomentations`



Preprocessing

Most* model architectures considered take images as input.

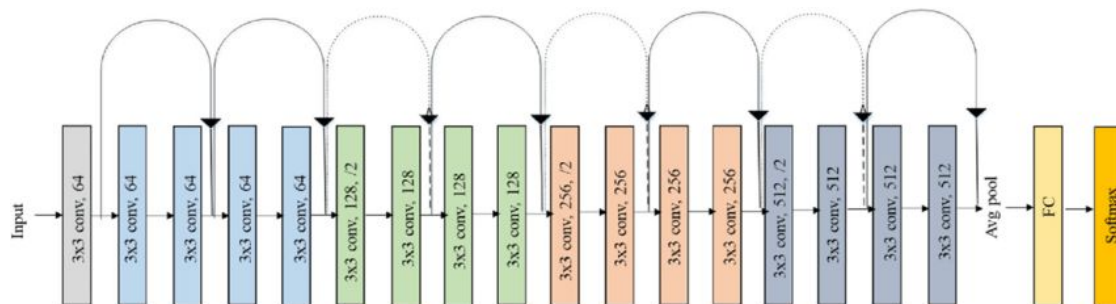
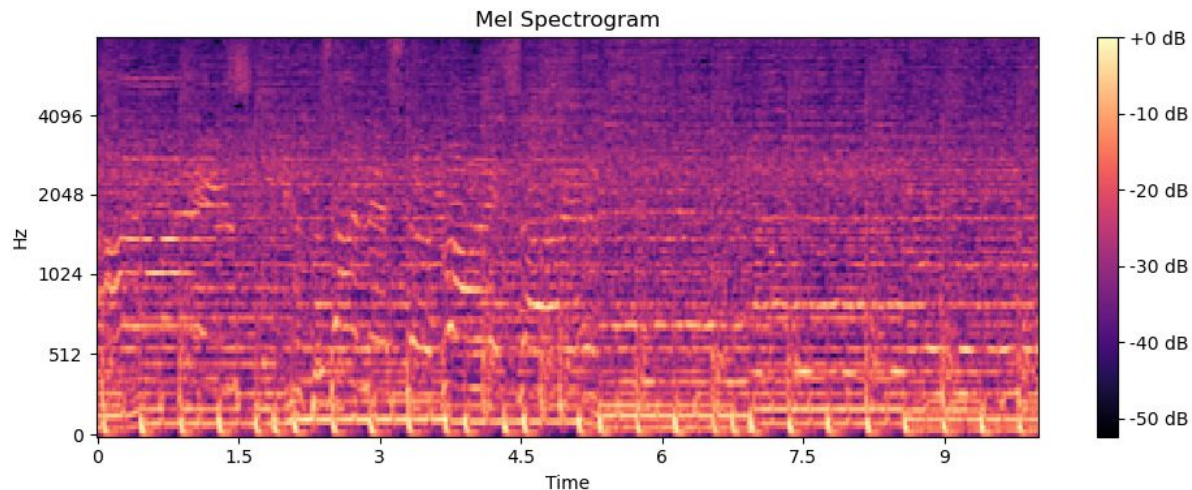
For these models, each audio clip was first converted to an image.

The *Mel Spectrogram* shows the volume of each pitch at each moment in time. (See example at upper right.)

Python library: `librosa`

Each spectrogram was fed into several different model architectures, including one built on pre-trained weights from ResNet-18. (Architecture at lower right.)

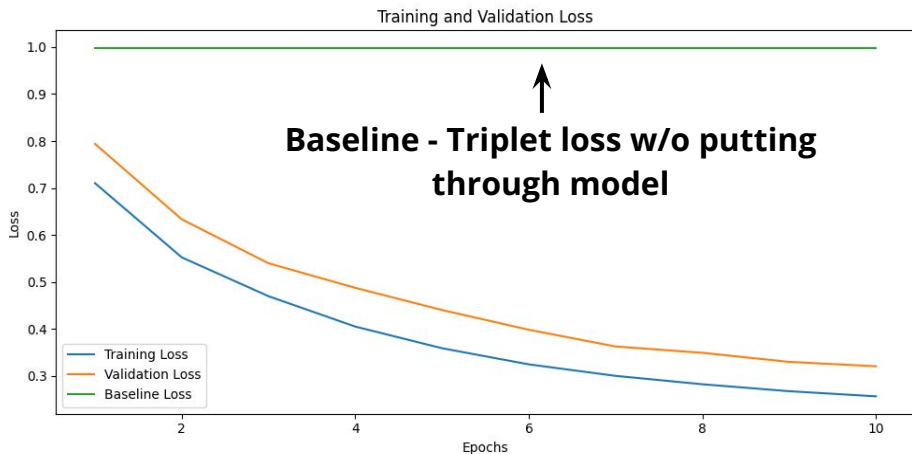
**Hubert takes raw audio as input*



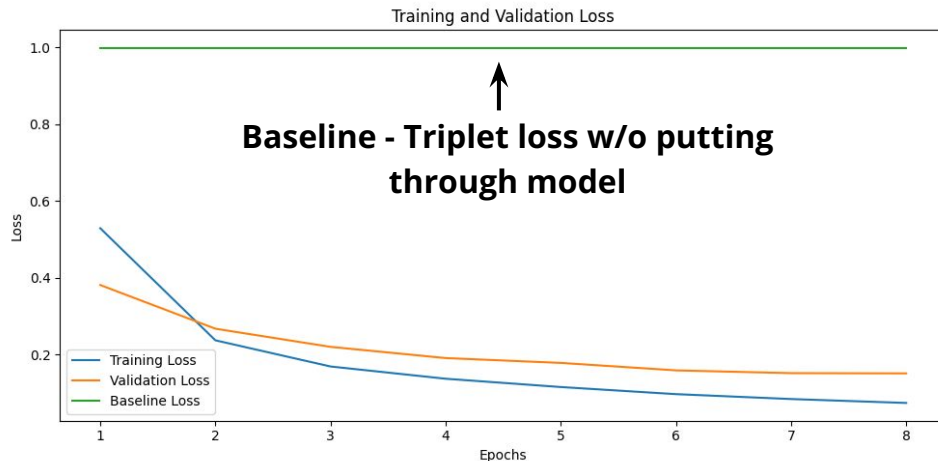
Results

- Studied various model architectures to create embeddings that minimize triplet loss

12 Layer CNN (2.2M params)



Fine Tuned ResNet-18 (11M+ params)



Results

Our ResNet-18 fine tuning yielded the best results, beating the triplet loss of our baseline (triplet loss *without* creating embeddings)

- Batch size: 32
- All layers unfrozen
- 6 epochs (limited by GPUs)

We achieved our goal of beating the baseline but there is much more progress to be made.

Metric	Result
Best Validation Triplet Loss	0.1514
Baseline Triplet Loss	0.9983
Pct Improvement over Baseline	84.83%

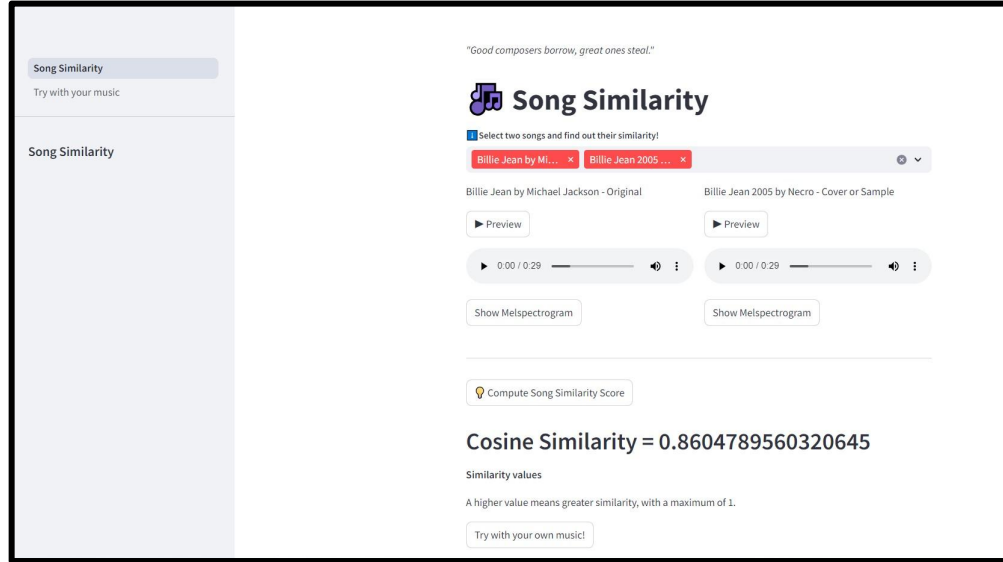
Web Demo

We made a Streamlit demo showing similarity scores.

It allows you to explore popular songs and upload your own audio to test the model.

URL:

<https://song-similarity-webapp.streamlit.app/>



The screenshot displays the 'Song Similarity' web application interface. At the top, a quote reads: "Good composers borrow, great ones steal." The main heading is "Song Similarity" with a music note icon. Below the heading, a prompt says "Select two songs and find out their similarity!". A search bar contains two entries: "Billie Jean by Mi..." and "Billie Jean 2005 ...". Below the search bar, two song preview cards are shown: "Billie Jean by Michael Jackson - Original" and "Billie Jean 2005 by Necro - Cover or Sample". Each card has a "Preview" button and a progress bar showing "0:00 / 0:29". Below the preview cards, there are "Show Melspectrogram" buttons for each song. A "Compute Song Similarity Score" button is located below the preview cards. The result is displayed as "Cosine Similarity = 0.8604789560320645". Below the result, there is a section titled "Similarity values" with the text "A higher value means greater similarity, with a maximum of 1." and a "Try with your own music!" button.

Future Directions

Going forward, ***improving data quality*** is a top priority:

- Storage for longer song samples, multiple recordings of songs
- Create more robust triplets with more difficult positives/negatives
- Study embeddings using tempograms/chromagrams, etc.

Additionally, we can work to ***improve our models*** via

- More training epochs, explore hyperparameter space
- Try different loss functions,
- Train plagiarism classifier on top of our embeddings
- Further study/incorporate features used in plagiarism cases



THE ERDŐS INSTITUTE

Helping PhDs get jobs they love.

Helping you hire the PhDs you need.

Thank you!

The Erdős Institute

Lindsay Warrenburg

Marcos Ortiz

