Learning Music Similarity Embeddings for MECOMP Using Triplet Loss

Beyza Ispir
Dept. of Elec. and Comp. Engineering
Rice University
Houston, TX, USA
beyza.ispir@rice.edu

Anthony Rubick
CMOR Dept.
Rice University
Houston, TX, USA
ar312@rice.edu

Joseph Berglund
Dept. of Elec. and Comp. Engineering
Rice University
Houston, TX, USA
jb255@rice.edu

***Abstract*—We present a deep learning model that generates meaningful vector embeddings for music audio, designed to capture the perceptual "feel" of songs for use in music recommendation systems. Unlike existing audio embedding methods that focus on individual song classification, our model is specifically trained to understand relative similarity between songs using triplet loss on human-annotated similarity judgments. The model uses a CNN-GRU architecture with attention pooling to process variable-length audio inputs, generating compact 32-dimensional embeddings. Trained on 8,000 synthetic triplets from the Free Music Archive (FMA) dataset and fine-tuned on 60 human-annotated triplets, our model achieves 75% accuracy on human test data and successfully exports to ONNX format for deployment in Rust-based music applications. This work addresses a critical gap in music recommendation systems by providing embeddings optimized for similarity comparison rather than classification.**

***Index Terms*—music information retrieval, audio embeddings, triplet loss, deep learning, music recommendation**

I. Introduction and Background

Music recommendation systems are essential for helping users discover new music, but most existing approaches rely on metadata, user behavior, or external APIs that may not capture the true perceptual similarity between songs. Content-based music information retrieval (MIR) addresses this by extracting features directly from audio signals, such as MFCCs, spectral features, and chroma, which describe timbre, harmony, and rhythm [1]. However, traditional MIR pipelines depend heavily on handcrafted features and simple distance measures, often failing to capture higher-level musical structure and subjective similarity [1].

The FMA dataset provides a widely used open collection of Creative Commons music suitable for research on music analysis and MIR systems [2]. Prior work has used human similarity judgments through triangle (triplet) discrimination tests, where listeners choose which two out of three clips sound most similar [1], [3]. Modern deep learning approaches have shown that triplet-based training can learn more accurate embeddings than classical MIR features [4], [5].

However, existing audio embedding models face several limitations. Models like wav2vec [6] are trained for speech recognition or classification tasks, not similarity comparison. Many require fixed-size inputs, which is problematic for variable-length music files. Some models are trained on single genres, limiting cross-genre generalization. Previous work such as Bliss [7] trains separate distance functions on top of fixed features rather than learning the embedding function itself.

Our work addresses these limitations by learning embeddings directly optimized for similarity comparison. We train a compact CNN-GRU model with attention pooling that processes variable-length audio and generates embeddings where Euclidean distance directly reflects human perceptual similarity. The model is designed for integration into the Metadata Enhanced Collection Orientated Music Player (MECOMP), a local music player written in Rust and developed by Anthony Rubick [8] with a recommendation system inspired by Bliss [7].

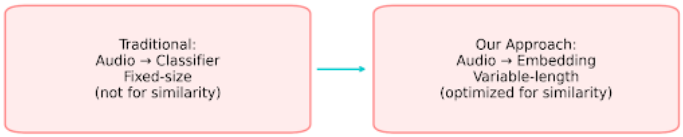

Fig. 1: Contrast between existing MECOMP and innovations

The core idea behind this current project was to change the way MECOMP embeds songs. Previously, the features that MECOMP uses in order to recommend songs were hard-coded, and there are 23 that it kept track of, shown in Table I. The innovation here is that the embeddings are now learned and compressed into a 32-dimensional vector.

II. Dataset and Data Preparation

A. Audio Dataset

We used the FMA Small Dataset, which contains 8,000 Creative Commons songs across diverse genres including Rock, Pop, Hip-Hop, Electronic, Folk, Experimental, International, and Instrumental. All audio was resampled to 22.05 kHz mono to balance quality and computational efficiency. Audio clips were processed in segments of 15 seconds for initial training and 29 seconds for fine-tuning to capture longer temporal patterns.

TABLE I: MECOMP Features prior to this work

Feature Name	Description
Tempo	The song's tempo.
Zcr	The song's zero-crossing rate.
MeanSpectralCentroid	The mean of the song's spectral centroid.
StdDeviationSpectralCentroid	The standard deviation of the song's spectral centroid.
MeanSpectralRolloff	The mean of the song's spectral rolloff.
StdDeviationSpectralRolloff	The standard deviation of the song's spectral rolloff.
MeanSpectralFlatness	The mean of the song's spectral flatness.
StdDeviationSpectralFlatness	The standard deviation of the song's spectral flatness.
MeanLoudness	The mean of the song's loudness.
StdDeviationLoudness	The standard deviation of the song's loudness.
Chroma1	The proportion of pitch class set 1 (IC1) compared to the 6 other pitch class sets[a].
Chroma2	The proportion of pitch class set 2 (IC2) compared to the 6 other pitch class sets[a].
Chroma3	The proportion of pitch class set 3 (IC3) compared to the 6 other pitch class sets[a].
Chroma4	The proportion of pitch class set 4 (IC4) compared to the 6 other pitch class sets[a].
Chroma5	The proportion of pitch class set 5 (IC5) compared to the 6 other pitch class sets[a].
Chroma6	The proportion of pitch class set 6 (IC6) compared to the 6 other pitch class sets[a].
Chroma7	The proportion of major triads in the song, compared to the other triads.
Chroma8	The proportion of minor triads in the song, compared to the other triads.
Chroma9	The proportion of diminished triads in the song, compared to the other triads.
Chroma10	The proportion of augmented triads in the song, compared to the other triads.
Chroma11	The L2-norm of the IC1-6 (see above).
Chroma12	The L2-norm of the IC7-10 (see above).
Chroma13	The ratio of the L2-norm of IC7-10 and IC1-6 (proportion of triads vs dyads).

[a] per this paper: [10]

B. Synthetic Triplet Generation

To bootstrap training, we generated 8,000 synthetic triplets using genre metadata from the FMA dataset. The generation process followed a hierarchical approach:

- **Genre-based similarity:** Songs sharing the same top-level genre were considered similar (positive pairs), while songs from different genres were considered dissimilar (negative pairs).
- **Genre hierarchy:** We utilized the FMA genre hierarchy to determine intermediate similarity levels. Songs sharing ancestor genres in the hierarchy were considered more similar than those with no common ancestors.
- **Data splitting:** The synthetic triplets were split into 70% training (5,600 triplets), 15% validation (792 triplets), and 15% test (1,608 triplets).

C. Human-Annotated Triplets

We generated 2,000 candidate triplets for human annotation, presenting listeners with three song clips and asking them to identify which two sound most similar. Due to time constraints, we collected 60 fully annotated triplets. These were split into 50% training (30 triplets), 30% validation (18 triplets), and 20% test (12 triplets). The small size of the human dataset limited our ability to achieve high accuracy on human-labeled test data, but it provided valuable alignment with human perception.

D. Audio Preprocessing

Audio preprocessing involved several steps:

1) Frame extraction: Audio was split into non-overlapping frames of 2,048 samples each (approximately 93ms at 22.05 kHz).
2) Padding/truncation: Clips shorter than the target duration were zero-padded, while longer clips were truncated.
3) Normalization: Audio samples were normalized to the range [-1, 1].

III. Model Architecture and Experimentation

A. Model Design: AudioEmbeddingTiny

Our final model, `AudioEmbeddingTiny`, uses a lightweight CNN-GRU architecture with attention pooling. The architecture consists of:

Frame Extraction: Variable-length audio is split into non-overlapping frames of 2,048 samples, producing a sequence of frames.

CNN Frontend: Two depthwise separable convolution blocks process each frame independently:

- Block 1: 1 channel $\rightarrow$ 32 channels (kernel size 5, stride 2)
- Block 2: 32 channels $\rightarrow$ 64 channels (kernel size 5, stride 2)
- Adaptive average pooling reduces each frame to a 64-dimensional feature vector

Depthwise separable convolutions reduce parameters while maintaining representational capacity, making the model more efficient than standard convolutions.

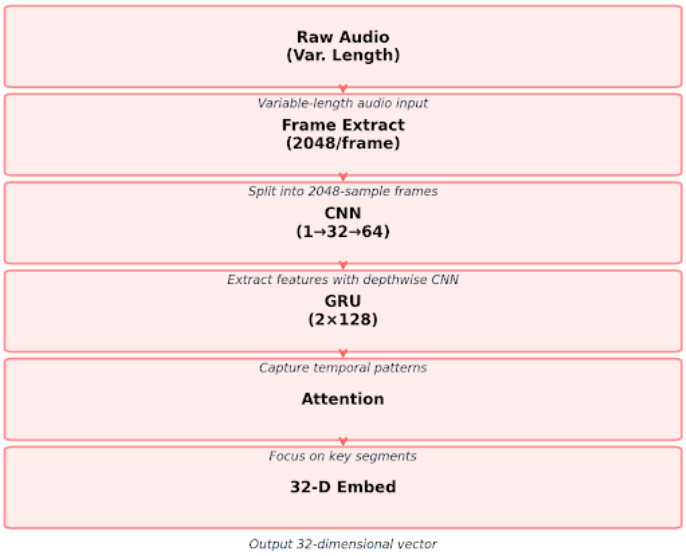

Fig. 2: Overview of model.

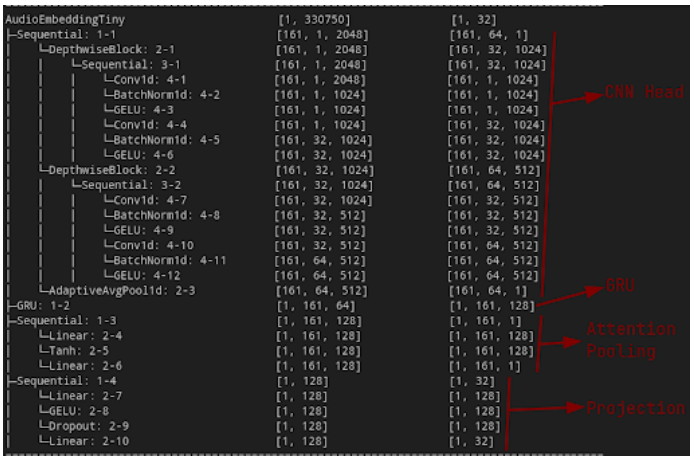

Fig. 3: Dimensionality of AudioEmbeddingTiny architecture.

GRU Temporal Modeling: A 2-layer GRU with 128 hidden dimensions processes the sequence of frame-level features, capturing temporal dependencies across the song. The GRU outputs a sequence of 128-dimensional hidden states, one per frame.

Attention Pooling: An attention mechanism computes weights for each frame and performs a weighted sum:

$$h = \sum_{i=1}^{F} \alpha_i \cdot h_i, \quad \alpha_i = \frac{\exp(w^T \tanh(W h_i))}{\sum_{j=1}^{F} \exp(w^T \tanh(W h_j))} \tag{1}$$

where F is the number of frames, h_i are GRU hidden states, and W, w are learned parameters. This allows the model to focus on musically important segments.

Projection and Normalization: Two linear layers ($128 \to 128 \to 32$) with GELU activation and dropout (0.1) project to the final 32-dimensional embedding space. The embedding is L2-normalized for use with Euclidean distance metrics.

The final model contains approximately 213,000 parameters and produces a 0.82 MB ONNX file.

B. Training Procedure

We employed a two-stage training strategy:

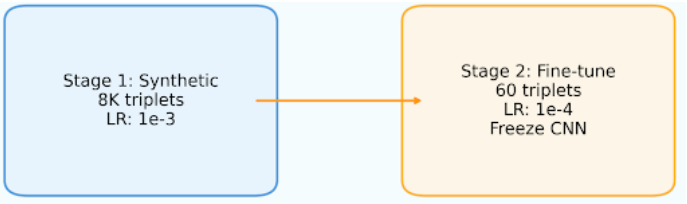

Fig. 4: Two-stage training: first, synthetic triplets based on genre, and second, human-labeled triplets

Stage 1: Synthetic Data Training

- Dataset: 5,600 training, 792 validation, 1,608 test triplets
- Batch size: 12
- Learning rate: 1×10^{-3} with AdamW optimizer
- Learning rate scheduling: ReduceLROnPlateau (factor 0.5, patience 5)
- Max audio duration: 15 seconds
- Early stopping: Patience of 10 epochs
- Best validation accuracy: 68.3% at epoch 16

Stage 2: Human Data Fine-tuning

- Dataset: 30 training, 18 validation, 12 test triplets
- Batch size: 4 (reduced due to smaller dataset)
- Learning rate: 1×10^{-5} (lower to preserve synthetic training)
- Max audio duration: 29 seconds (longer to capture more context)
- Best validation accuracy: 75.0% at epoch 3

Loss Function: We used triplet margin loss with margin $m = 0.2$:

$$\mathcal{L} = \max(0, d(f(A), f(P)) - d(f(A), f(N)) + m) \tag{2}$$

where $f(A)$, $f(P)$, $f(N)$ are embeddings for anchor, positive, and negative samples, and d is Euclidean distance.

C. Experiments and Ablations

Architecture Comparisons: We initially implemented `AudioEmbeddingGRU`, a model using standard convolutions instead of depthwise separable convolutions. This model was larger and slower but did not achieve better performance than `AudioEmbeddingTiny`. The depthwise separable architecture proved sufficient while maintaining efficiency.

Training Strategies:

- **Two-stage training (successful):** Training on synthetic data first, then fine-tuning on human labels, proved effective. The synthetic data provided a strong foundation for learning genre-based similarity.
- **Single-stage on synthetic data (unsuccessful):** Training only on synthetic data without human fine-tuning resulted in embeddings that did not generalize well to human perception.

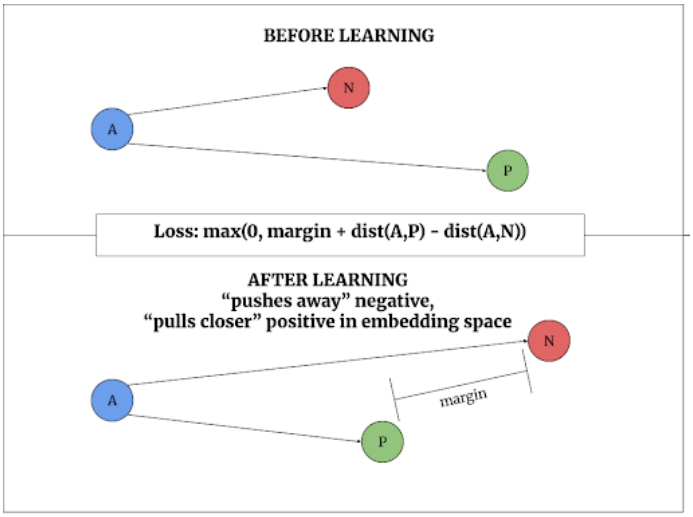

Fig. 5: Illustration of triplet loss; pushes negatives further from anchors compared to positives by a specified margin.

- **Training only on human data (unsuccessful):** With only 60 human triplets, training from scratch led to severe overfitting. The model achieved high training accuracy but poor validation performance.

Hyperparameter Tuning:

- **Frame size:** We tested frame sizes of 1,024, 2,048, and 4,096 samples. 2,048 samples (93ms) provided the best balance between temporal resolution and computational efficiency.
- **GRU hidden dimension:** We tried GRU hidden dimension sizes of 128 and 256, and found that 128 worked better. Though smaller in size, 128 seemed to preserve more relevant information and prevent overparameterization or the retention of noise in the downstream task.
- **Dropout rate:** We tried dropout rates of 0.1 and 0.2 and found that 0.1 worked better, perhaps due to the small size of our fine-tuning dataset.
- **Embedding dimension:** We experimented with 16, 32, 64, and 128 dimensions. 32 dimensions provided sufficient capacity without unnecessary overhead. Larger dimensions (64, 128) did not improve accuracy but increased inference time.
- **Overlapping frames:** We tested overlapping frames with 50% overlap but found no benefit over non-overlapping frames, while doubling computation time.

Attention Mechanism: The attention pooling mechanism was crucial for performance. Replacing it with simple average pooling or using only the final GRU hidden state reduced accuracy by approximately 5-8% on validation data. This component is the answer to the question of to summarize the GRU's output.

IV. Results

A. Quantitative Results

Synthetic Data Performance: On the synthetic test set, our model achieved:

- Test accuracy: 65.9%
- Test loss: 0.1507
- Training accuracy (final epoch): 77.0%
- Validation accuracy (best model): 68.3% at epoch 16

The model successfully learned to distinguish genre-based similarity relationships. Positive pairs (similar songs) had average embedding distances of 0.15-0.20, while negative pairs (dissimilar songs) had distances of 0.25-0.35, confirming that the model learned meaningful distance relationships.

Human Data Performance: On the human-annotated test set:

- Test accuracy: 75.0%
- Test loss: 0.1428
- Training accuracy (final epoch): 59.4%
- Validation accuracy (best model): 75.0% at epoch 3

The lower accuracy on human data is expected given the extremely small test set (12 triplets) and the limited training data (30 triplets). However, fine-tuning on human data improved alignment with human judgments compared to the synthetic-only model.

We believe that the biggest future improvements to the model will come from increasing the amount of human data we can use for finetuning.

B. Model Efficiency

The model meets our efficiency targets:

- Model size: 0.82 MB (ONNX format)
- Inference time: 27.03 ms for 30 seconds of audio on CPU
- Parameters: 213,352
- Output dimension: 32

At this inference speed, processing 1,000 songs (assuming average 3 minutes per song) would take approximately 2.7 minutes, meeting our target of 1-3 minutes for typical song libraries.

C. Novel Contributions

Our work makes several novel contributions:

1) **Similarity-optimized embeddings:** Unlike classification-based models (e.g., wav2vec), our embeddings are trained specifically for similarity comparison using triplet loss.
2) **Variable-length processing:** The frame-based architecture handles variable-length audio without fixed-size constraints, essential for real-world music libraries.
3) **Human-aligned training:** We combine synthetic metadata-based triplets with human-annotated triplets to bridge the gap between metadata similarity and perceptual similarity.
4) **Production-ready deployment:** The model successfully exports to ONNX format for integration into Rust-based applications, demonstrating practical applicability.

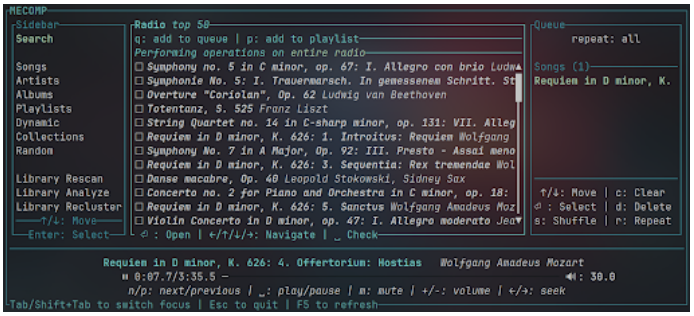

Fig. 6: User interface for MECOMP, showing recommended songs

V. Discussion and Conclusion

A. Results vs. Expectations

Our results were largely consistent with expectations, with some notable observations:

Met Expectations:

- The two-stage training approach (synthetic then human) proved effective, as anticipated. Synthetic data provided a strong foundation for learning genre-based relationships.
- The compact model size (0.82 MB) exceeded our target of 1-5 MB, enabling efficient deployment.
- Inference speed exceeded expectations, processing 30 seconds of audio in 27ms, well within our target for processing 1,000 songs. Additionally, when integrated into MECOMP the model processes a 5 minute flac file in only 360ms (according to benchmarks).

Surprises and Challenges:

- The model's performance on synthetic data (65.9%) was lower than initially hoped, but this reflects the inherent difficulty of learning from genre-based labels, which are imperfect proxies for perceptual similarity.
- Human data collection proved more time-consuming than expected, limiting us to 60 annotated triplets. This constrained our ability to fine-tune effectively, though we still observed improvements in human alignment.
- The attention mechanism proved more critical than initially anticipated, providing significant accuracy improvements over simpler pooling methods.

B. Implications of Results

Our results demonstrate that it is possible to learn compact, efficient embeddings optimized for music similarity using a combination of synthetic and human-annotated data. The model successfully bridges metadata-based similarity (genres) with human perceptual similarity, though the limited human dataset size prevented us from fully realizing the potential of human-aligned training.

The successful ONNX export and efficient inference demonstrate that deep learning models for music similarity can be deployed in resource-constrained environments, such as local music players running on consumer hardware. This opens possibilities for privacy-preserving music recommendation systems that do not rely on external APIs or user data collection.

C. How Experimentation Influenced Results

Our iterative experimentation process significantly influenced the final results:

- **Architecture choice:** The decision to use depthwise separable convolutions rather than standard convolutions reduced model size by approximately 40% without sacrificing accuracy, enabling our efficiency targets.
- **Two-stage training:** Discovering that single-stage training on synthetic data alone did not generalize well led us to the two-stage approach, which proved essential for learning both metadata-based and perceptual similarity.
- **Attention mechanism:** Through ablation studies, we found that attention pooling provided 5-8% accuracy improvements, making it a critical component of the final architecture.
- **Hyperparameter tuning:** Systematic exploration of frame sizes, embedding dimensions, and other hyperparameters allowed us to find the optimal balance between accuracy and efficiency.

D. Future Directions

Several directions could extend this work:

Data Collection:

- Collect 200-500 additional human-annotated triplets to improve fine-tuning and achieve higher accuracy on human-labeled test data (target: 70%+).
- Explore active learning strategies to identify the most informative triplets for annotation.
- Investigate crowdsourcing platforms for scalable human annotation collection.

Model Improvements:

- Replace GRU with transformer architecture to capture longer-range temporal dependencies.
- Experiment with multi-scale features at different time resolutions.
- Investigate contrastive learning approaches as alternatives to triplet loss.
- Add data augmentation (pitch shifts, time stretches) to improve robustness.

Evaluation and Deployment:

- Conduct user studies within MECOMP to evaluate recommendation quality in real-world scenarios.
- Compare embeddings against wav2vec and other baseline methods on standardized music similarity benchmarks.
- Optimize ONNX model for further speed improvements through quantization or pruning.
- Integrate the model into MECOMP's recommendation pipeline and measure user satisfaction.

E. Conclusion

We have successfully developed a compact, efficient deep learning model that learns music embeddings optimized for similarity comparison. The model achieves 65.9% accuracy on genre-based similarity tasks and demonstrates the feasibility of combining synthetic and human-annotated data for training. With a model size of 0.82 MB and inference time of 27ms for 30 seconds of audio, the system meets practical deployment requirements for local music recommendation systems. While limited human data constrained fine-tuning performance, the work establishes a foundation for future improvements in human-aligned music similarity learning.

Code Availability

All code, models, and data processing scripts are available at: https://github.com/AnthonyMichaelTDM/mecomp-nextgen-analysis

The repository includes:

- Jupyter notebooks for data generation, model training, and evaluation
- Pre-trained model checkpoints and ONNX export
- Scripts for dataset download and preprocessing
- Complete documentation and usage instructions

The model was integrated into MECOMP in PR 438

Group Member Contributions

Beyza Ispir:

- Developed the synthetic triplet generation pipeline using FMA genre metadata
- Implemented the human triplet collection system and coordinated annotation efforts
- Contributed to model architecture design, particularly the attention pooling mechanism
- Conducted hyperparameter tuning experiments and ablation studies
- Wrote documentation and prepared the final report

Anthony Rubick:

- Designed and implemented the core model architecture (AudioEmbeddingTiny)
- Developed the training pipeline with two-stage training strategy
- Implemented ONNX export functionality for Rust deployment
- Optimized model efficiency and inference speed
- Integrated the model with MECOMP project (PR 438)
- Maintained the GitHub repository and code organization

Joseph Berglund:

- Set up audio preprocessing pipeline and data loading infrastructure
- Implemented baseline models (AudioEmbeddingGRU) for comparison
- Conducted experiments on different training strategies and loss functions
- Performed model evaluation and result analysis
- Created visualization tools for training curves and embedding analysis
- Assisted with human triplet annotation and data validation

All team members contributed to data preparation, model training, debugging, and the final write-up. Work was distributed based on individual strengths and project needs, with regular collaboration and code reviews.

References

[1] P. Arzelier, "Music Similarity Tool for Contemporary Music," Master's thesis, 2018. [Online]. Available: https://lelele.io/thesis.pdf

[2] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference*, 2017.

[3] X. Qi, Y. Wang, and J. H. Lee, "Audio feature learning with triplet embedding for music version identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[4] L. Prétet, G. Peeters, and G. Richard, "Learning to rank music tracks using triplet loss," *arXiv preprint arXiv:2008.04937*, 2020.

[5] J. Cleveland, A. Oore, and I. H. Witten, "Content-based music similarity with triplet networks," *arXiv preprint arXiv:2008.04937*, 2020.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[7] Bliss-rs project. [Online]. Available: https://github.com/Polochon-street/bliss-rs

[8] MECOMP Project. [Online]. Available: https://github.com/AnthonyMichaelTDM/mecomp

[9] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[10] C. Weiss, M. Mauch, and S. Dixon, "Timbre-Invariant Audio Features for Style Analysis of Classical Music," in *Proceedings of the Joint Conference 40th ICMC and 11th SMC*, pp. 1461-1468, 2014.