CLASSIFYING AI GENERATED VS NON-AI GENERATED MUSIC WITH RECOMMENDATION FOR TOP-K RELATED NON-AI SONGS

MATT (JIWOONG) PARK, HASSAN RIZWAN, STANLEY LIU, SPENCER WARE

ABSTRACT. This paper presents models designed to detect AI-generated music and recommend similar human-composed alternatives. We employ a Convolutional Neural Network (CNN) on mel spectrograms to distinguish AI-created music from human-performed music. Upon detecting an AI-generated song, we utilize OpenL3 embeddings and FAISS (Facebook AI Similarity Search) with K-Nearest Neighbor (KNN) algorithm to retrieve the top-10 closest human songs. Our models were trained and evaluated using a dataset from AI-generated SONICS tracks and human GTZAN recordings. Results indicate that the classifier achieves over 99% accuracy, while the recommendation system maintains a high average cosine similarity of 98.6%. Ultimately, our purpose for this project to solve the misuse of synthetic music by creating a query platform to redirect listener attention toward human artists.

1. INTRODUCTION AND BACKGROUND

Recent advances in generative audio models, such as Suno and Udio, have made it possible to synthesize high-fidelity music that is increasingly difficult to distinguish from human-composed tracks. Streaming platforms are already seeing a rapid influx of AI-generated songs, and early user studies suggest that most listeners cannot reliably tell whether a track was created by a human or by a model. This raises practical concerns for both audiences and artists. Listeners who wish to support human musicians lack clear signals about the origin of a track, while human artists face growing competition from synthetic content that can be produced at scale and tailored to popular styles. Additionally, a listener might discover an AI track that they enjoy, but have no obvious way to find human artists producing similar music.

To address these issues, we built a binary audio classifier that distinguishes human-composed tracks from AI-generated music. If an AI-generated song is detected, the system recommends the top-K ($K = 10$) human-composed songs that are closest in timbre, rhythm, and style to the input track. We constructed a dataset by combining AI songs from the SONICS dataset with human-performed music from the GTZAN genre dataset. All tracks were standardized to a fixed sample rate and waveforms were converted into mel spectrograms.

A convolutional neural network (CNN) binary classifier was built to predict the probability that the underlying audio was generated by an AI model. We utilized OpenL3, a pre-trained audio embedding model that inputs an audio spectrogram and extracts a 512-dimensional vector, or "sonic fingerprint," that represents the song's features. Finally, we used FAISS and K-Nearest Neighbor (KNN) algorithm for fast similarity search and clustering of large, high-dimensional vectors. This was to find the 10 closest vectors in our human song database using cosine similarity, defined as $\frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$, to identify the human tracks that most closely match the AI song's vector.

1.1. Contributions.

Our contributions are three-fold:

(1) **End-to-end system for AI detection and human-song recommendation.** We implemented a complete pipeline that detects whether an input track is AI-generated using a CNN-based binary classifier on mel spectrograms, and when an AI song is detected, recommends the top-K ($K = 10$) human-composed songs that are closest in timbre, rhythm, and style.

(2) **Integration of pre-trained audio embeddings with a curated human-song database.** We combine OpenL3, a large-scale pre-trained audio embedding model, with a curated collection of human tracks from GTZAN to create a vector database of 512-dimensional "sonic fingerprints." By running each human song through OpenL3, we enable fast similarity search in a space that captures high-level musical characteristics beyond simple genre or metadata labels.

(3) **Empirical evaluation of AI–human similarity via KNN in embedding space.** We systematically evaluate a K-Nearest Neighbor retrieval scheme that uses cosine similarity in OpenL3 embedding space to map AI-generated songs to semantically similar human tracks. Our experiments show that this approach reliably retrieves musically coherent recommendations and illustrates how embedding-based similarity can be leveraged to turn AI music into a query mechanism for discovering human artists.

2. Related Works

SONICS (Rahman et al., 2025) introduces a large-scale synthetic song dataset and a neural model, SpecTTTra, focusing on long-range temporal patterns for end-to-end synthetic music detection, demonstrating strong performance compared to conventional CNN/Transformer baselines [1, 6]. Our approach uses a lightweight CNN to accomplish this same task on their dataset. The recommendation algorithm acts atop OpenL3, a library developed by Cramer et al. in 2019 that generates deep audio embeddings using a convolutional neural network trained via audio-visual correspondence learning on large-scale unlabeled video data [2]. By combining synthetic song detection with content-based retrieval in a shared embedding space, our approach not only identifies AI-generated music but also recommends acoustically similar human-composed songs when such content is detected, reframing detection as a user-facing redirection mechanism rather than a binary decision.

Related efforts include pretrained audio embedding models such as VGGish and YAMNet for transfer learning, contrastive self-supervised audio representation, and prior work on singing-voice deepfake detection, though these approaches either lack end-to-end song modeling or do not address downstream recommendation [4, 5, 7, 8, 9, 10].

3. Approach

There are three main parts to the project. First is our CNN binary classifier, which, given a song, determines whether the song was generated by an AI model. If the song is determined to be probabilistically AI-generated, then we run the .wav file through genre mapping which converts the input genre into the closest genre in the human-made song dataset. Then the .wav file is passed into the KNN algorithm to find the 10 closest human-made songs with the closest genre.

Dataset Construction (Classifier): Our goal is to learn a detector that focuses on the distinction between AI-generated and human-composed music. We used the SONICs dataset with 1000 AI generated song and the GTZAN dataset with 1000 human songs.

All audio is mapped into a common time–frequency representation based on mel spectrograms to standardize the data (30 second clip length, 16,000 Hz sampling rate, and mel configuration) as detailed in the experimental results section. Conceptually, this design encourages the model to rely on spectral and temporal structure that is stable across datasets (e.g., timbre, texture, and production artifacts) rather than simple dataset-specific cues.

Vector Database Construction (Recommendation System): To construct a vector database of real songs for our recommendation system, we used OpenL3 with an embedding size of 512 and a hop-size of 0.1 seconds for all songs within the `GTZAN` dataset. The resulting encoding (of size 512×300) was then aggregated to be one vector of size 512 to capture some notion of an overall style of a given song, and each embedding vector was then passed through an ℓ_2-norm to ensure that our KNN model's distance metric would be interpretable as a cosine similarity when querying against an input AI song. For our query vectors, we used randomly sampled AI songs from the `SONICS` dataset.

Binary Classifier Design: Given these standardized spectrograms, we frame AI detection as a binary image-classification problem on single-channel 2D inputs using a convolutional neural network.

Local 3×3 convolutions capture short-time spectral motifs (such as characteristic textures, transients, or noise floors), while pooling layers introduce a degree of invariance to small shifts in time and frequency, which is desirable for music signals. After several such blocks, a global aggregation layer condenses the remaining feature maps into a fixed-size embedding, which is passed through a small fully connected head with dropout and a final sigmoid unit that outputs the estimated probability that the input track is AI-generated. The network is trained end-to-end with a standard logistic (binary cross-entropy) objective, so that its internal representation is optimized specifically for separating AI from human music.

KNN: The FAISS library was used to fit a KNN model over the OpenL3-embedded vectors of real songs from the `GTZAN` dataset, with a flat index being used. Since we ℓ_2-normalized our songs, inner product used within this flat index was equal to the cosine similarity between two vectors.

Closest Genre Mapping: There are 10 different genre files that we classified from the human-songs dataset [”metal”, ”disco”, ”classical”, ”blues”, ”hiphop”, ”jazz”, ”country”, ”rock”, ”pop”, ”reggae”]. Because there are more than 30 unique genres that were labeled in the SONICS AI-generated music, we created an algorithm that, given an unseen genre, approximates to the closest of the 10 human-song genres. We do this by comparing the ”average sound” of the AI genre to the ”average sound” of known genres.

First, we precompute the human-music genres by calculating a centroid, or a mathematical average of what the genre sounds like based on its embedded wav format. Then, when we feed an unknown genre from the AI-made music, the algorithm finds all the AI songs in the dataset that are labeled with the same genre. We take the cosine similarity of the average between the AI-made song genres and the 10 human-made song genres and find the highest similarity score. Because each mapping is computationally intensive, we cache each mapped path to avoid duplication.

Top-K Recommended Human-Made Songs: After computing the AI-made song's closest human-made song genre and the cosine similarity of the embedded .wav file between the AI-made song and human-made songs through KNN, we used $k = 100$ for the number of nearest neighbors in KNN we were searching for at any given point, after which we filtered for the top 10 songs (sorted by cosine similarity of the music) within the same genre as the input AI song.

4. Experimental Results

4.1. **Binary Classifier.**

4.1.1. *Data and Pre-processing.* We evaluate our binary CNN classifier on a dataset constructed from two sources: AI-generated music from the SONICS dataset and human-composed music from the GTZAN genre dataset. We subsample 1,000 AI tracks and 1,000 human tracks, and split them into 70/15/15% train/validation/test sets using stratified sampling. For each song, we extract the first 30 seconds, resample to 16 kHz, and convert the waveform into a mel spectrogram with 128 mel bands and a hop size of 512, yielding spectrograms of shape 128×938. Finally, we transform the spectrograms to decibel scale and apply per-example standardization (zero mean, unit variance).

4.1.2. *Evaluation Metrics.* Binary cross-entropy loss and classification accuracy on the training and validation sets were used for to evaluate the binary CNN classifier. Accuracy directly reflects how often the model correctly distinguishes AI versus human tracks and binary cross-entropy loss helps us identify overfitting and optimization issues even when accuracy appears saturated.

4.1.3. *Experimental Setup.* The binary classifier CNN is a four-block 2D convolutional network operating on single-channel mel spectrograms of shape $1 \times 128 \times 938$. The model uses max pooling after each block, global average pooling, and two fully connected layers with dropout before a final sigmoid output. We optimize binary cross-entropy loss with the Adam optimizer (learning rate 10^{-3}), a batch size of 32, and train for 20 epochs on a single GPU.

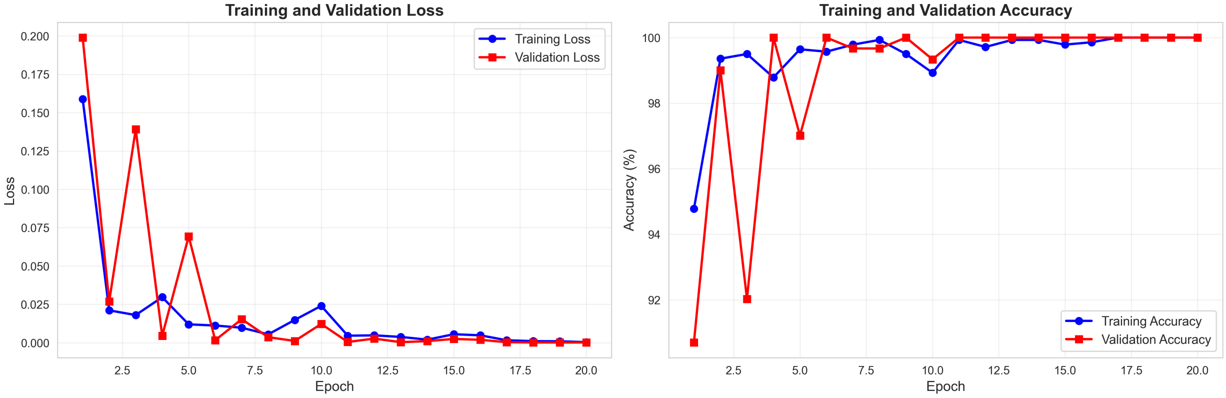

FIGURE 1. Binary Classifier Training and Validation Loss and Accuracy

4.1.4. *Results.* Figure 1 shows that both training and validation loss drop rapidly in the first few epochs (from roughly 0.15–0.20 to ≈ 0.02), then gradually approach near-zero values, while accuracy climbs from about 90–95$\%$ to $\gtrsim 99\%$ by around epoch 8 and then plateaus near 100%. Early spikes in validation loss (e.g., around epochs 3 and 5) disappear after several epochs, and the training/validation curves closely track each other thereafter, indicating that the CNN is able to fit the data well without obvious signs of overfitting. The close alignment of both plots suggests that the model generalizes well within our SONICS and GTZAN splits, and that AI vs. human labels are highly separable in the mel-spectrogram representation under our preprocessing pipeline.

However, the model may be leveraging dataset-specific artifacts (e.g., genre mix, mastering style, encoder settings) that systematically differ between SONICS and GTZAN, rather than purely capturing "AI-ness" versus "human-ness." While the experimental results confirm that our CNN architecture and preprocessing choices are sufficient to achieve excellent detection accuracy on this benchmark, additional experiments on other AI generators and noisier real-world audio would be necessary to fully assess how robust this classifier is beyond the SONICS and GTZAN pair.

4.2. **OpenL3 and Nearest Neighbors Recommendation System.**

4.2.1. *Data.* We used the vector database construction over the `GTZAN` dataset discussed in the Approach section, which contained a mean-pooled vector for each encoding given by OpenL3. The encoding used a hop size of 0.1 seconds and an embedding of size 512. The sampling rate was once again set to 16,000 for consistency with the classifier. Randomly sampled songs from the `SONICS` dataset were used as query points and their cosine similarity was used to evaluate their closeness to a real song.

4.2.2. *Evaluation Metrics.* To measure the effectiveness of our embeddings, we generated a confusion matrix by taking each real song's closest neighbor (excluding itself), and verifying that the genre of the neighbor was equal to the genre of the selected song. The confusion matrix may be seen in Figure 2. We note that while this confusion matrix captures the effectiveness of our embeddings along some notion of a genre dimension, OpenL3 embeddings capture low-level timbral, rhythmic, and textural characteristics rather than explicit genre boundaries. Finally, to evaluate the performance of our KNN model, we evaluated the average cosine similarity of 50 randomly selected AI-generated songs with their top-10 recommendations.

4.2.3. *Experimental Setup.* The recommendation system uses mean-pooled ℓ_2-normalized 512-dimensional OpenL3 embeddings indexed with a FAISS flat inner-product index over the `GTZAN` dataset. AI-generated songs from `SONICS` are embedded identically and used as queries to retrieve the Top-10 human songs via cosine similarity after genre-based filtering.

4.2.4. *Results.* As previously mentioned, our metric for measuring the effectiveness of our OpenL3 embeddings was comparing the genre corresponding to each song's nearest neighbor that wasn't itself to it's own genre. This resultant confusion matrix can be seen here:

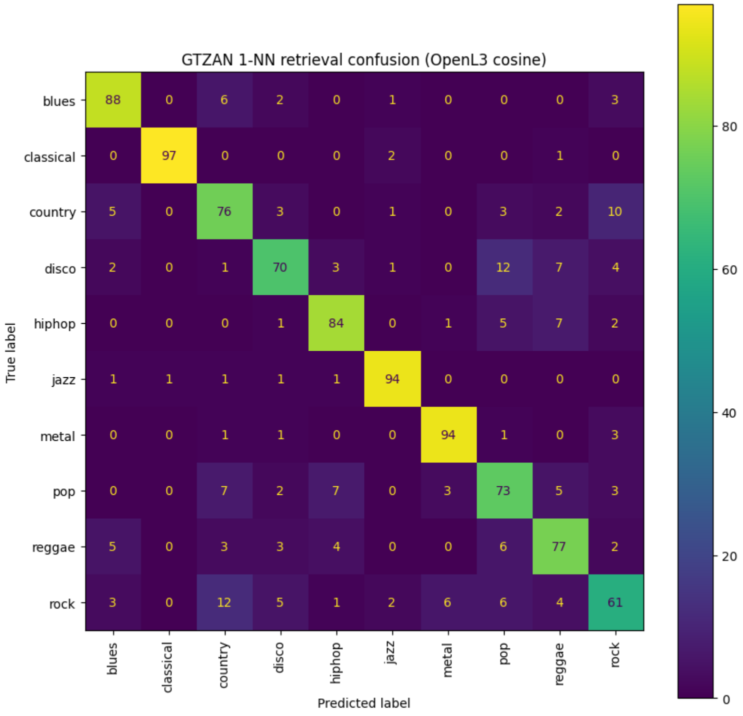

FIGURE 2. OpenL3 Embedding Confusion Matrix

We took 50 randomized datasets for AI-made songs from SONICS and calculated the average of the Top-10 similar Human-made songs. The x-axis represents each of the AI-made song samples, and the y-axis represents the average cosine similarity. All the 50 samples came out to be consistently high, averaging 98.6% Mean Similarity, as shown in Figure 3.

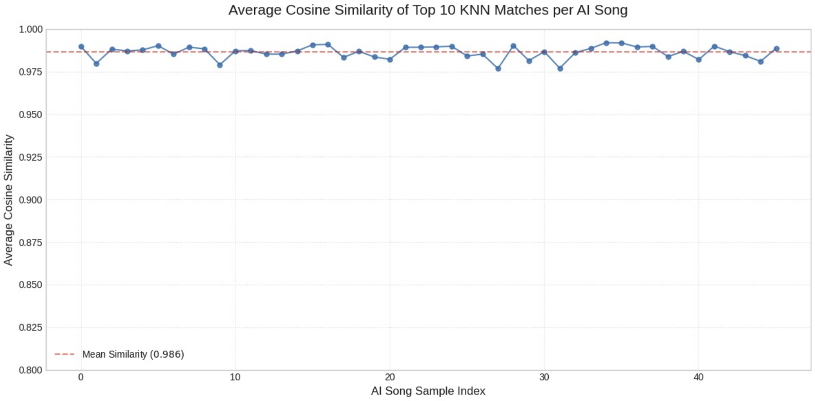

FIGURE 3. Average Cosine Similarity of 50 random AI-made Songs (98.6% Mean Similarity)

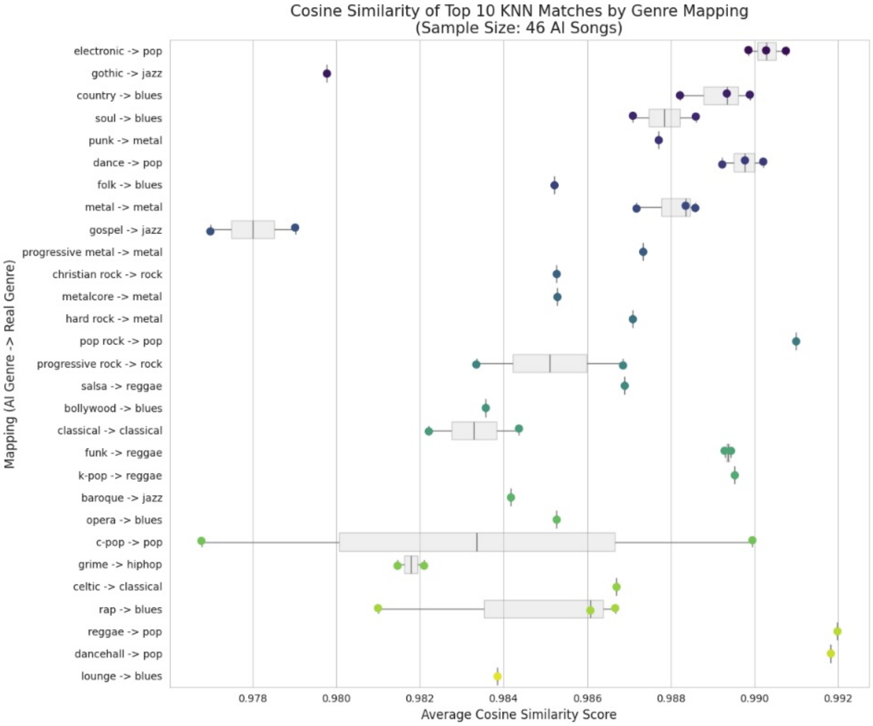

FIGURE 4. Cosine Similarity of Closest Genre Mapping

We also calculated the cosine similarity for each genre mapping. The genre mapping algorithm takes the cluster of songs that are associated with the same genre for both AI-made and Human-made songs. Then, the algorithm computes cosine similarity between each cluster's average embedded value and pick the top similarity for genre mapping. The result shows a high correlation for the selected genre mappings without large variation, with the range 97.7% - 99.2% cosine similarity. Empirically, we can also observe the accuracy of the mappings with examples such as "soul → blues," "c-pop → pop," "progressive metal → metal."

5. DISCUSSION

We believe we had great success in the project with high classification accuracy (90–95% to $\gtrsim 99\%$) for the CNN binary classifier and 97.7% - 99.2% cosine similarity for the result of KNN.

One of the features we tried to include in our KNN model was PCA, mainly due to concerns about overfitting and low accuracy. However, the result turned out that the cosine similarity was already at a high cosine similarity (95%)

without PCA. Furthermore, the results were highly accurate on test sets, reducing our concern about overfitting. In contrast, the inclusion of PCA reduced cosine similarities of songs to 29% - 50%. We suspect that reducing a dense 512-dimensional embedding removed "fine-grained" details that distinguished neighbors.

Our next step in the process is to integrate our model and algorithms into a deployable web application. In the future, we plan on developing a homepage that allows a drag and drop feature of different .wav files. The resulting page will validate whether the song is AI-generated or Human-made, and if the input is AI-generated, suggest the Top-10 human-made songs of similar content and genre in a dashboard display with a preview feature.

REFERENCES

[1] A. Awsaf, ”Sonics,” Hugging Face Datasets, 2024. [Online]. Available: https://huggingface.co/datasets/awsaf49/sonics

[2] A. L. Cramer, H. -H. Wu, J. Salamon and J. P. Bello, ”Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings,” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 3852-3856, doi: 10.1109/ICASSP.2019.8682475.

[3] E. Davies, ”AI-generated music could soon dominate Spotify and Billboard charts, study finds,” The Guardian, Nov. 2025. [Online]. Available: https://www.theguardian.com/technology/2025/nov/13/ai-music-spotify-billboard-charts

[4] Gurjar, K., Moon, Y.-S., & Abuhmed, T. (2023). TruMuzic: A Deep Learning and Data Provenance-Based Approach to Evaluating the Authenticity of Music. Applied Sciences, 13(16), 9425. https://doi.org/10.3390/app13169425

[5] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., … Wilson, K. (2017). CNN architectures for large-scale audio classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 131–135). IEEE. https://doi.org/10.1109/ICASSP.2017.7952132

[6] Rahman, M. A., et al. (2025). SONICS: Synthetic or not—Identifying counterfeit songs. In Proceedings of the International Conference on Learning Representations (ICLR). OpenReview.https://openreview.net/forum?id=PY7KSh29Z8

[7] Plakal, M., Wang, D., & Ellis, D. (2020). AudioSet classification with convolutional neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1125–1129). IEEE. https://doi.org/10.1109/ICASSP40776.2020.9052994

[8] Saadatnejad, S., Wang, Z., Ghasemzadeh, H., & Yadollahpour, P. (2021). Contrastive learning of general-purpose audio representations. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3870–3874). IEEE. https://doi.org/10.1109/ICASSP39728.2021.9413478

[9] Wang, X., Zhang, Y., & Xia, X. (2024). SingFake: Singing voice deepfake detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing. https://doi.org/10.1109/TASLP.2024.3360196

[10] Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., & Li, H. (2023). Spoofing and countermeasures for speaker verification: A survey. Speech Communication, 66, 130–153. https://doi.org/10.1016/j.specom.2014.10.005