Introduction

Music plagiarism has been a legal gray area for many decades. The complexity of music making contributes to the difficulty of defining plagiarism among songs. In the past, music plagiarism occurred in two contexts: with a musical idea (melody, motif) or sampling (taking a portion of one sound and reusing it in a different song). According to U.S. copyright law, in the absence of a confession, musicians who accuse others of stealing their work must prove "access"—the alleged plagiarizer must have heard the song—and "similarity"—the songs must share unique musical components. The focus of this project will be on the similarity between two songs, i.e. we will train a network as a similarity metric between two segments of audios. Due to the limitation of time and resources, this project will purely focus on audio similarity. Similarity in lyrics, for example, will be out of the scope of this project. Our aim is a model that reliably reports a low distance between similar audio files and a high distance between dissimilar ones.

In addition producing a similarity score, our key performance indicators are determining if there is an existing trend with court decisions in music plagiarism, and (if time permits) building a web for giving a similarity score between any two audio files. Having a model that can effectively accomplish these tasks has potential utility to stakeholders such as artists, record labels and other copyright holders, tech companies in the music industry (e.g., Spotify, Deezer, Tidal), and the judiciary system.

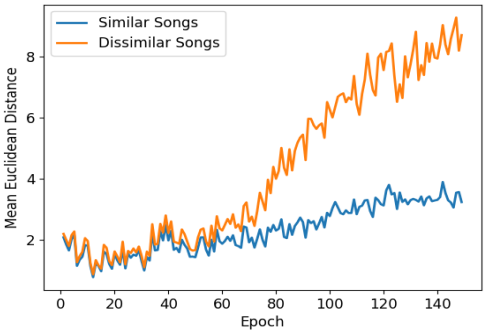
Methodology

We followed the approach of [Kasif and Thondilege 2023]. The dataset used consists of songs appearing in WhoSampled.com. WhoSampled.com provides lists of songs that have sampled another song, along with timestamps of where the relevant sample begins. We start by compiling pairs of songs from whosampled.com, along with the relevant timestamps. The pairs are then downloaded from YouTube as .mp3 files (for a total of 1128 pairs of songs). With all the audio files downloaded, we utilized two different methods of extracting the sampled portion. The first method is to slice ten seconds of the sampled segments. The second method is to slice the duration equivalent to 8 bars of the sampled segments, in an attempt to generate diagrams invariant under different tempo. To make these extracted sample portions usable as input to our model, we translated them into images using two different approaches of music representation: mel-spectrogram and chroma feature. For mel-spectrograms, fast Fourier transform is first applied on audio signals and then frequencies are converted to mel scale, which is a unit of pitch such that equal distances in pitch sound equally distant to the listener. Visualizing these mel-scaled frequencies in terms of their amplitudes yields the mel-spectrogram. For chroma features, frequencies are binned into twelve different pitch classes that can capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation. In sum, we have four types of images – mel-spectrogram 10 seconds, mel-spectrogram 8-bar, chroma feature 10 seconds, chroma feature 8-bar. All of these different image types serve as separate inputs to the Siamese convolutional neural network (CNN) that we use to train to produce a feature map that encodes information about the input images. The

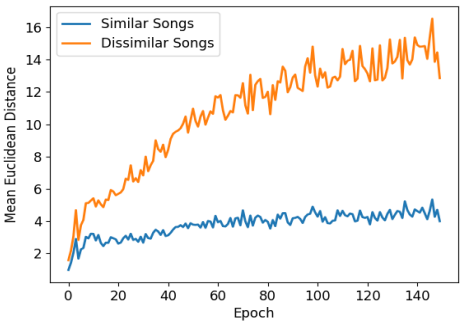
Siamese CNN serves as an encoder to extract information from the input images. During the training, three images are randomly drawn, one is the anchor image, one is an image from the same class, and the last one is an image from a different class. Next, the loss function minimizes the Euclidean distance between the feature maps of the anchor image and the image from the same class, while maximizing the distance between the feature maps of the anchor image and the image from a different class. By doing so, the trained model is able to extract information from the image that groups songs with the same sample together while separating different songs. Specifically, we used Xception from [Chollet 2017] as our CNN. We trained the networks on the four datasets using Adam optimizer for 150 epochs.

Results

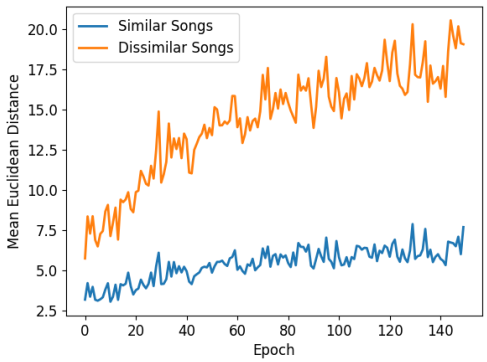
As a result of the training process, we were able to obtain four models that could distinguish similar songs and dissimilar songs within the training datasets of chroma-feature 8-bar samples, chroma-feature 10 seconds samples, mel-spectrogram 8-bar samples, and mel-spectrogram 10 seconds samples, respectively. The mean Euclidean distance between the dissimilar songs is much greater than the mean Euclidean distance between the similar songs, after a training of around 150 epochs. Note that—since minimizing the triplet loss function is only equivalent to separating the mean Euclidean distances by a margin— we would not be able to achieve a near-zero mean Euclidean distance between the positive sample and the anchor sample. We ended up with a value around two for our mean Euclidean distance between our similar song pairs, which suggests that the training via Triplet loss would not be able to provide us with a metric such that we could measure the similar song pairs by a distance close to zero. That being said, by minimizing the triplet loss function, ideally, we should be able to generate models that could separate the positive and negative samples by giving a much greater distance to dissimilar song pairs in comparison with similar song pairs. The result of the training with all four datasets is shown below in Figure 1.



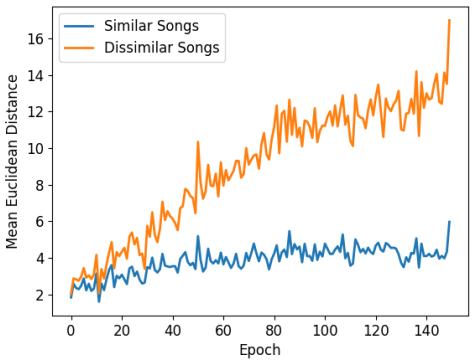
a



b



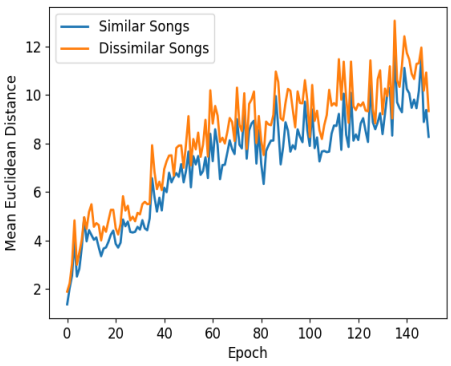
c



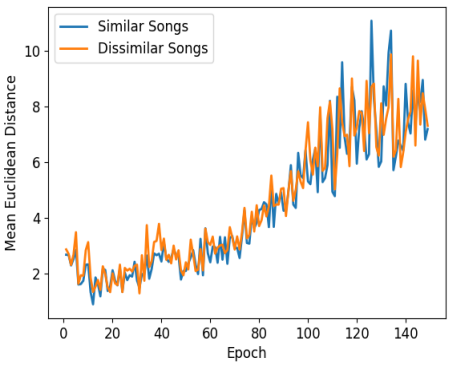
d

Figure 1: Mean Euclidean distance from training results, with datasets: a. Chroma Feature 8-bar; b. Chroma Feature 10-seconds; c. Mel-Spectrogram 8-bar; d. Mel-Spectrogram 10-seconds.

We then validate our trained models with our testing dataset and compute the Euclidean distance for the testing datasets of chroma-feature 8-bar samples, chroma-feature 10 seconds samples, mel-spectrogram 8-bar samples, and mel-spectrogram 10 seconds samples, respectively. The validation result, as shown below in Figure 2, shows that the mel-spectrogram 10 seconds samples is the dataset that has trained the best model, being able to separate the similar songs and dissimilar songs with a value around ten in the mean Euclidean distance. The mel-spectrogram 8-bar samples performed the second best, giving a separation in the distance of around five. The two datasets generated from chroma-feature, namely the chroma-feature 10 seconds samples and chroma-feature 8-bar samples, do not show a recognizable difference on the mean Euclidean distance between similar songs and dissimilar songs, meaning that they cannot serve as plausible models for separating the similar and dissimilar songs.



a



b

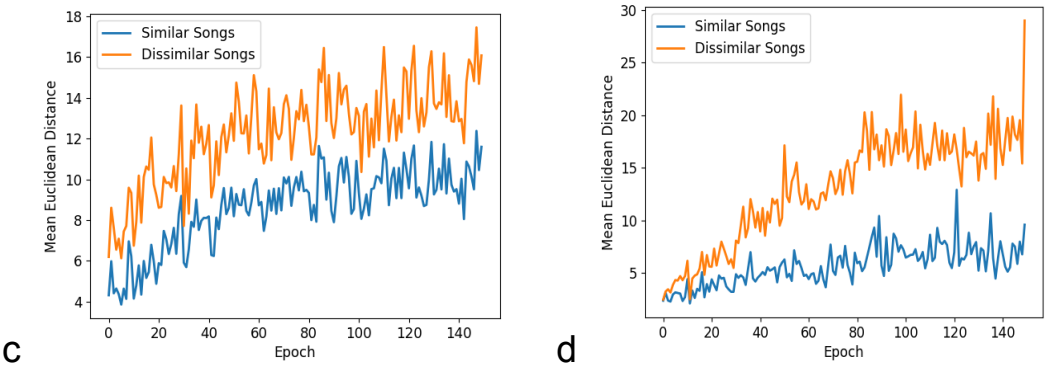


Figure 2: Mean Euclidean distance from validation results, with datasets: a. Chroma Feature 8-bar; b. Chroma Feature 10-seconds; c. Mel-Spectrogram 8-bar; d. Mel-Spectrogram 10-seconds.

Finally, we used the best-performing 10-second mel-spectrogram model to calculate a similarity score for legal cases that we collected, 9 guilty, 6 settled, 6 acquitted. The similarity score is labeled as Euclidean Distance in the y-axis of Figure 3. As we discussed before, the smaller the distance is, the more similar the two songs are. Our model produced very small distances for guilty cases, while larger distances for settled or acquitted cases. Even though the sample size for the court cases is small, and there is overlap between the boxplots of guilty and not-guilty cases, our model had a notable differentiation in distances between guilty and settled cases.

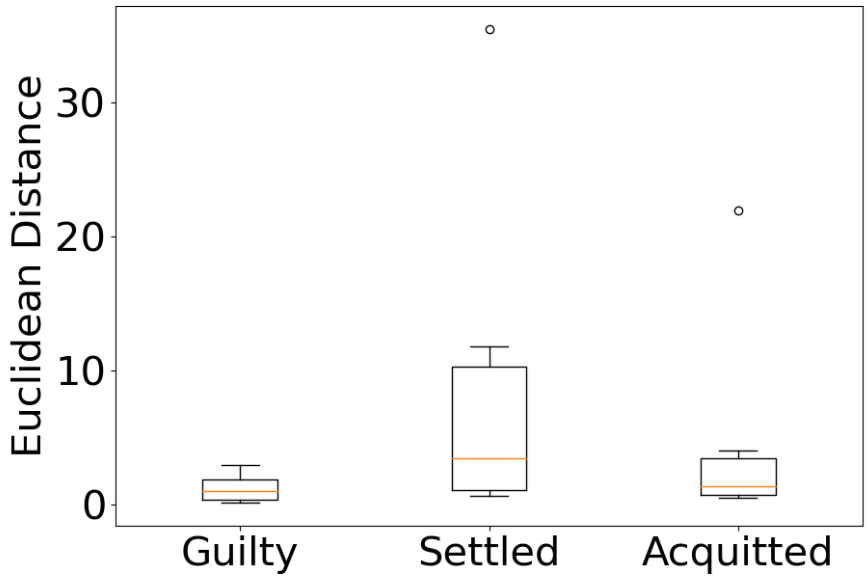


Figure 3: Euclidean distance for Court Cases

Conclusion

In conclusion, our model was able to distinguish similar songs from dissimilar songs. Due to this, we believe that we can reasonably expect our model to give useful results when used by music industry professionals to assess whether a new song is similar enough to be a potential case of plagiarism. The use of Chroma features may have had poor results due to dominance of percussion used in our dataset. Chroma feature may prove more effective when used in cases where a melody or a tune is more saliently plagiarized. In the future, it might be beneficial to consider what type of plagiarism is being examined, such as beat vs melody, when selecting the dataset. Additionally, numeric representation of music might be preferable to image representation when processing the data. For example, using the values in the mel spectrogram rather than the image of it. Other avenues of possible improvement would in clyde investigate other types of networks. Possible architectures include Autoencoder, Long-Short Term Memory and Recurrent CNN.

References

[Chollet 2017] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p./pp. 1251--1258), [Kasif and Thondilege 2023] Kasif, G. and Thondilege, G. (2023). Exploring Music Similarity through Siamese CNNs using Triplet Loss on Music Samples. *2023 International Research Conference on Smart Computing and Systems Engineering (SCSE).*