Synthetic Music Detection Through Image Classification

Antonio Cordeiro (1999975), Lorenzo Ugolini (1958654), Emidio Grillo (1996918), Matteo Sorrentini (2023085)

La Sapienza Università di Roma

December, 2024

Abstract

With the advent of more and more sophisticated generative models, it has become exponentially easy to create artificially generated music that highly mimics human compositions. Automated artificial music recognition can play an important role in contrasting the thinning of the differences between what is generated by a human and what is not. This type of work was already investigated in the Afchar et. al. paper [1], where they achieved incredible results in detection accuracy (99.9%), but also highlighted difficulties concerning robustness to manipulation and generalization. In this study, we first utilize spectrograms of AI-generated music tracks to extract color and derivative histograms. We analyze these histograms and employ their features for the identification of music deepfakes. Subsequently, we compare these results with the performance of Convolutional Neural Networks (CNNs) for the same task, demonstrating the necessity and efficacy of the latter models.

1. Introduction

Content creation is being highly revolutionized by advanced generative model, which enable the synthesis of realistic audio, video, text, and music with little more than a click. Along with breaking barriers of possibilities and creativity, they also introduce the problem of distinguishing human-generated content from artificial one. Music field is one of the most involved, with the proliferation of deepfakes generated through advanced neural networks, capable of producing compositions indistinguishable from human works. Detecting these deepfakes is essential for various reasons, as protecting copyright, and supporting fair competition.

This project builds upon these foundations, exploring the use of spectrograms as the primary input to classify music as either artificial or human-generated. Spectrograms, which visually represent the frequency content of an audio signal over time, can be exploited to detect hits and patterns indicative of the presence of generative models. Convolutional Neural Networks (CNNs) are employed to develop a robust classification system.

Contributions. This work makes the following contributions:

- An analysis on the color and derivatives histograms given by the spectograms and using histogram features for deepfake music identification.
- Showing the efficacy of CNNs, using a similar approach to Afchar et al. [1], on a new dataset.
- Exploring the use of a pre-trained CNN model, in particular ResNet18[4], for music deepfake classification.

2. Related Work

The field of AI content detectors is exponentially increasing in last years. While the research is quite advanced for items like texts or images, the works concerning artificially generated music are very rare. Starting from our idea of AI music classification from spectrogram images, the only reference we found embracing this technique was the paper ”Detecting music deepfakes is easy but actually hard” [1]. In this work, they presented a classification method based on Convolutional Neural Networks, also researching the detection of artificially manipulated music pieces.

3. Dataset

The dataset used in this project is created by combining two different datasets. This comes from the necessity of having both labeled artificial and natural music pieces and from the fact that there were no pre-existing datasets suitable for this task.

The two chosen datasets are the *FMA: A Dataset for Music Analysis*[2] for natural music and the *AAM: Artificial Audio Multitracks Dataset*[3] for artificial tracks.

3.1. FMA

The FMA (Free Music Archive) dataset was originally created for Music Information Retrieval, a field concerned with browsing, searching, and organizing large music collections. The complete dataset provides 106,574 tracks, arranged in a hierarchical taxonomy of 161 genres. For our project we used a subset of the total dataset, already provided by the creators, named *fma_small*, composed of 8,000 tracks of 30s labeled with 8 balanced genres. The choice of this dataset was due to the presence of full-length, high-quality audio files. The audio tracks have been processed to generate the spectrogram images.

3.2. AAM

The "AAM: a dataset of Artificial Audio Multitracks for diverse music information retrieval tasks" is composed of 3000 tracks artificially generated by algorithmic composition. The dataset provides rich annotations based on real instruments. One of the key elements to be evaluated in the choice of the artificial music dataset is the generation process. If the generation process is executed starting from given genres or rules, the training process would be highly biased and therefore unreliable when performing classification.

The AAM tracks are generated by an *Algorithmic Composer* focused more on the music structure. It creates tracks based on rules typical of Western popular music, like steady rhythms, 4/4 time signatures, major-minor system chord progressions, singable melodies, and a moderate variety of instruments, including some non-Western ones to introduce diversity. Each music piece is divided into three to five parts (i.e. at least chorus, verse, and bridge) and rooted in principles of repetition, variation, and contrast.

The generation process is guided by mathematical models with random decisions constrained by knowledge-based rules. Key musical attributes such as tempo, key, and length are randomly selected within predefined ranges, ensuring variety. For example, tempos range from 60 to 180 BPM, and keys are randomly selected from all twelve root notes in either major or minor tonality.

To avoid arbitrary randomness, the algorithm uses the memoization concept, where decisions are influenced by previous ones. For instance, the key or tempo of later sections is probabilistically related to initial choices, promoting coherence.

3.3. Dataset Setup

The two datasets could not be directly merged due to some structural differences in the number of tracks and in their length.

The AAM is composed of 3,000 tracks, each available both played by a single instrument and by a mix of them. Of course, for our purposes, we used the mixed tracks. They are also available in MIDI format, which have not been considered since the MIDI format is not directly representable by a spectrogram, but it would need a conversion into mp3 or similar formats.

The FMA tracks have a fixed duration of 30 seconds, while the length of the artificial tracks varies from 120 to 180 seconds. To overcome this difference, for each of the chosen AAM tracks a random 30-second segment is extracted.

To ensure balance between artificial and natural samples, we randomly selected 3,000 tracks from the fma_small dataset, matching the size of the AAM dataset. The two dataset were then merged obtaining a total of 6,000 tracks of 30 seconds each, balanced between AI-generated and genuine samples. Finally, spectrogram images were generated for all tracks in this merged dataset.

4. Models & Results

The first step to start working on our task was to analyze the spectrogram images, their appearance, and the features they show. Then, we proceeded to implement the classification task.

4.1. Histograms Analysis

During preliminary talks about the project, it was shown that it was often possible to distinguish between the two types of tracks just by looking at their spectrograms. For this reason, we performed a preliminary histogram analysis, using both colors and Gaussian partial derivatives histograms.

The color histograms were generated by analyzing the pixel intensities of the spectrogram images. To compute the histograms we only used the red (R) and blue (B) channels; this choice was driven by the characteristic color of the spectrograms, which have the audio frequencies plotted as shades of red on a blue background. The chosen number of bins was 10. In the end, the average color histogram for each class is generated to have an overview of the class behavior.

The derivatives histograms were generated by computing the joint histogram of Gaussian partial derivatives of each image in the x and y directions. The process was repeated for each image of each class and, at the end, the average histogram for both of them was generated.

Both types of histograms present high peaks in specific bins, likely to be attributed to the background of the spectrogram images, with a large number of pixels belonging to it and so falling in the same bins.

For both the techniques the two histograms are very similar to each other. For the color histograms the only noticeable difference is in the average frequency of the background color, a little higher in the Artificial spectrograms than in the Natural ones.

We can go deeper into the analysis by ignoring the noticed peaks and focusing on the rest of the histograms. The idea behind this operation is to consider the background as a part of the image not containing useful information or features. The result of this operation is visible in Fig. 1 and 2.

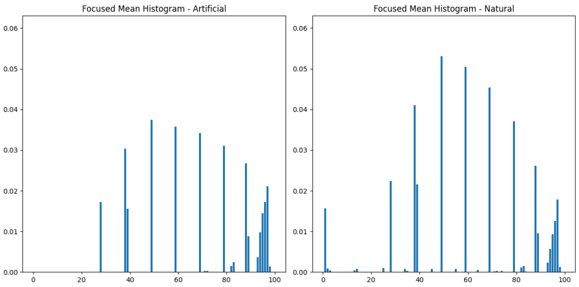

Figure 1: Focused Color Histograms

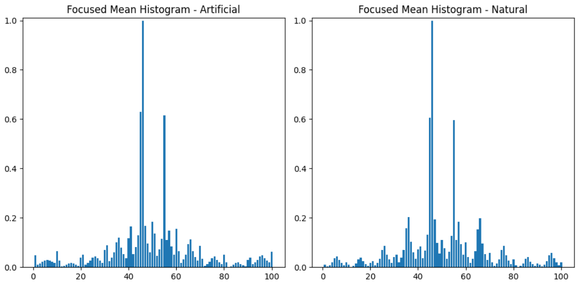

Figure 2: Focused Derivatives Histograms

The differences in the derivative and color histograms between natural and artificial spectrograms are not sufficiently pronounced to serve as definitive distinguishing features. Specifically, the maximum observed frequency differences are 5% for color histograms and 2% for derivative histograms.

4.2. Classification on Histogram Features

The first tentative of performing the classification class has been done by exploiting the features extracted from the previously generated histograms. We created a dataset of histograms, in which each entry got its peak bin normalized to the mean of the values, and then all the values got normalized with the *minmax* normalization technique. The generated dataset was then divided into train and test sets with a $80 - 20$ proportion.

Once the dataset was ready, the first classification was performed with Random Forest. For the Gaussian partial derivatives features the model obtained an accuracy of **91.3%**, with precision and recall both at 0.91.

Replicating the same data normalization techniques and training process with Random Forest on the color histogram features, we obtained a slightly worse result. The overall accuracy was **82.5%**, with precision and recall both at 0.83. From the confusion matrix shown in Figure 3, we can notice how the biggest type of error was labeling human-created tracks as artificially generated.

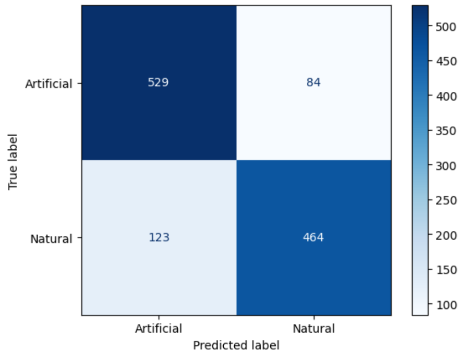

Figure 3: Confusion matrix of Random Forest model on color histogram features

We also explored an alternative approach on the histogram features by implementing a simple Neural Network. The architecture consisted of three fully connected layers, and Binary Cross-Entropy was employed as the loss function. For the training process, we divided the previously created training set into train and validation subsets.

Using Gaussian derivative histogram features, this approach demonstrated a modest improvement, achieving an accuracy of **96%**. However, when applied to color histogram features, the improvement was minimal, with the training process showing high losses for both the training and validation sets. The final accuracy on the test set was **84%**, representing only a slight improvement over the Random Forest baseline.

4.3. Classification on Spectrogram Images

After the first set of analyses on the histograms and their features generated from the spectrogram images, we performed the classification task using directly the images. For this second step, we used Convolutional Neural Networks. First, we used a custom CNN, then a pre-trained one to compare the results. Before the actual learning we set up a Dataset class on which samples got resized to $244 * 244$ images, and then got loaded with the DataLoader to be fed to the networks. They were converted into tensors and normalized.

Our custom model is composed of three sequential convolutional blocks, each combining a convolutional layer, batch normalization, ReLU activation, and max pooling. We included a dropout layer to mitigate overfitting. Finally, two fully connected layers map the high-dimensional features into a reduced feature space before producing a single output for binary classification.

The pre-trained model we used is the ResNet18 [4] architecture adapted to perform binary classification. The ResNet18 model is pre-trained on ImageNet and we leveraged the transfer learning technique. In this phase, the model's parameters were frozen to prevent updates during training, except for the last fully connected layer. This layer was replaced with a configuration modified for binary classification, consisting of a linear layer that reduces the output to a single unit with sigmoid activation function. The used loss function is Binary Cross Entropy with Logits Loss and the chosen optimizer is Adam.

The dataset was split into train, validation and test subsets with a proportion of 0.7-0.1-0.2 respectively. We observed that around 10 epochs were enough to train both models, as it can be observed in Fig. 4 and 5.

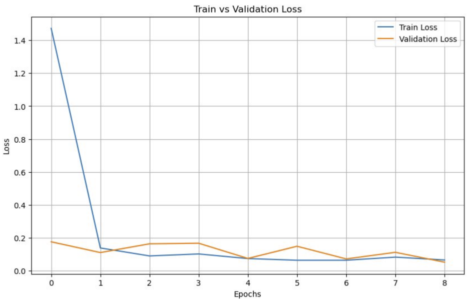

Figure 4: Train vs Validation loss during custom CNN training

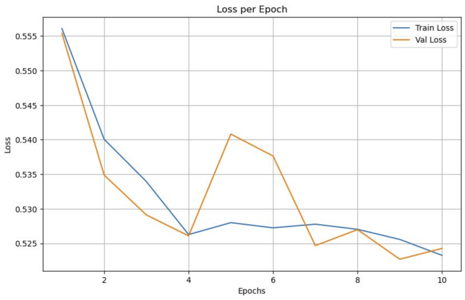

Figure 5: Train vs Validation loss during ResNet18 training

With our custom CNN going for more epochs quickly resulted in overfitting, while for ResNet18 the accuracy did not seem to improve even when training with more epochs. The little difference in the result was supposed to be due to the more complex architecture of the ResNet18 model, which needs a longer training phase to converge to the optimal values; however, even though we tried and increased the training loops, the result didn't improve. Thus, a simpler model performs better for music deepfake classification task. This may happen beacuse ResNet reaches a local minimum and gets stuck.

The accuracy for our custom model in the end was around **99.3%** and around **97%** for ResNet18. The results we obtained with our custom CNN are in line with the work by Afchar et al. [1]. In both cases the f1-score was really high, indicating a well-balanced model. Overall, using this approach with a CNN directly on the spectrogram images showed an improvement in performance with respect to using the hisograms features.

5. Conclusion

In conclusion, our work demonstrates the feasibility and challenges of distinguishing artificial music from human compositions. By leveraging spectrograms, we compared different classification models, achieving very high accuracy with both histogram-based and image-based classification techniques. Using histogram features, Gaussian partial derivatives histograms yielded remarkable results, both with the Random Forest and the Neural Network classifiers. Meanwhile, using image classification on the tracks spectrograms, ResNet18 showed good results outperforming the previous models based on histograms features. A further improvement was achieved with the implementation of a custom CNN model reaching results in line with the work in [1]. An overview of the results is shown in Table 1.

The AAM dataset is composed of algorithmically generated tracks, and its creation is from March 2023. We are well aware that progress in the field generative AI models produces outperforming results day by day. This is why the extremely promising results of our project have to be considered as an initial exploration of the possibilities in deepfake music detection. We can probably say that more recent and sophisticated generative models could represent a tougher challenge, being more able to mimic the structure and characteristic features of human compositions.

In anticipation of harder detection challenges, future work should try this approach with music generated by newer generative models and should aim to build a robust pipeline for deepfake identification.

Metric	Random Forest		Neural Network			
	Deriv. Fts.	**Color Fts.**	**Deriv. Fts.**	**Color Fts.**	**Our CNN**	**ResNet18**
Accuracy	0.92	0.83	0.95	0.83	0.99	0.97
Precision	0.92	0.83	0.94	0.84	0.99	0.97
Recall	0.92	0.83	0.96	0.81	0.99	0.96
F1-Score	0.92	0.83	0.95	0.82	0.99	0.97

Table 1: Results comparison of all classification models used in this work. Random Forest and Neural Network models on Gaussian partial derivatives and Color histogram features, our custom CNN model and ReseNet18 on spectrogram images

6. Contributions

- Dataset: setup, pre-processing; Histogram Anlysis: setup, features extractions, ML and DL models and visualizations; Image classification: custom CNN setup; Report - *Lorenzo*
- Histograms: features extraction; Image classification: ResNet setup and visualizations; Presentations - *Matteo*
- Image classification: ResNet setup and visualizations; Presentations - *Emidio*
- Dataset: setup; Image classification: custom CNN and ResNet training processes and optimization, results evaluation and visualizations; Report - *Antonio*

References

[1] D. Afchar, G. Meseguer-Brocal, and R. Hennequin, *Detecting music deepfakes is easy but actually hard*, 2024.

[2] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, *Fma: a dataset for music analysis*, 2017.

[3] F. Ostermann, I. Vatolkin, and M. Ebeling, "Aam: a dataset of artificial audio multitracks for diverse music information retrieval tasks", EURASIP Journal on Audio, Speech, and Music Processing **2023**, 10.1186/s13636-023-00278-7 (2023) 10.1186/s13636-023-00278-7.

[4] A. V. Sai Abhishek, "Resnet18 model with sequential layer for computing accuracy on image classification dataset", **10**, 2320–2882 (2022).