CS464
INTRODUCTION TO MACHINE LEARNING
FALL 2021

MUSIC GENRE CLASSIFICATION FROM MEL SPECTROGRAM DIAGRAMS

PROJECT PROGRESS REPORT

Melih Berk Yılmaz - EEE - 21803702
Muhammed Şamil Arınç - EEE - 21803683
Mustafa Çağrı Durgut - CS - 21801983
Oğuzhan Özçelik - CS - 21802194
Yusuf Miraç Uyar - CS - 21802626

28.11.2021

1. Introduction and Background Information

Our aim with this project was to classify music genres by using different types of machine learning algorithms. Models will try to predict genres from Mel Spectrogram diagrams. We planned to convert our dataset which consists of audio files into Mel Spectrogram. A Mel spectrogram is a visualization of the frequency spectrum of a signal, where the frequency spectrum of a signal is the frequency range that is contained by the signal. Then we planned to use this image data as our dataset for various Machine Learning algorithms.

We are using the Free Music Archive Small dataset for our purposes [1]. The dataset consists of 8000 tracks each of them are 30 second long audio files. There are 8 genres consisting of 1000 music data each. For each data, there are 30-sec long wav files. We research the prior works done using the FMA dataset. There are several papers written about training machine learning algorithms using the FMA dataset. One of them is the "Zero-Shot Learning For Audio-Based Music Classification" paper in which people who have written this paper used the FMA dataset to train a model predicting music using a zero-shot learning method which enables the model to predict labels that are not available in train set [2].

Another paper written using the FMA dataset is "An Open Dataset For Multiple Instrument Recognition" which examines the audio files as polyphonic melodies and tries to predict the musical instruments used in the given music file [3]. People writing this paper also generated another dataset using FMA data files to label musical instruments used in each audio file.

The problem we try to solve is not completely brand new. There are several datasets that are used by different papers. The GTZAN dataset is one of them [4]. In the future, we might utilize this dataset too. The dataset consists of 1000(number)x30(sec) audio files. There are 10 genres consisting of 100 music data each. For each data, there are 30-sec long wav files and 432x288 png files showing Mel Spectrogram. These spectrograms in the GTZAN dataset inspired us to train our model on the spectrogram pictures instead of audio files. Since our dataset did not consist of any spectrogram files, we should generate them before training our model. Also, some of the papers take metadata into account. So, there is another CSV file that has 57 different statistical parameters such as variance, mean, etc. This also inspired us to use telemetries in metadata to train our basic models.

2. What's been achieved so far?

Firstly, we gathered our data from the FMA dataset which we found on GitHub [1]. The problem with our dataset was that it was focused on audio files which is not suitable for a machine learning model. So, in order to convert it into usable data, we converted audio files to Mel spectrogram images which we have provided the definition in the Introduction and

Background Information section of this report. We used the Librosa library from Python for this purpose. Another problem with our dataset was that the data we had to work on was on separate files and there was not any documentation or instruction on how to combine them together in order to make them useful. It was somehow time consuming but we managed to combine the data by analyzing it manually. Then, the rest of the work was combining it in Python using pandas and numpy libraries. Like most machine learning projects, we did a data cleaning. There wasn't unnecessary data but there was corrupted data. We removed these data from our dataframes. The data we were given was actually already divided into training, test and validation. However, since there was corrupted data, we had to make our own division of training, test and validation as well. After all that, our data was ready to use. We implemented kNN and tested its accuracy on the dataset. In order to improve the accuracy of our results we tested our algorithms on various k values. We downsampled the images. Initially the images were 128x1291. Due to hardware constraints we firstly used 64x645 downsampled images. Then 32x322 downsampled images. We also are running our kNN on original images. However, we were not really satisfied with the accuracy of the results. So, we decided to test these algorithms on metadata in the feature

2.1 Results Obtained:

2.1.1 kNN Results:

K=	3	5	7	9	11	13	15	17	19	21	23
Euclidean Accuracy	21.37	20.87	24.25	23.37	23.00	23.75	24.00	24.50	23.62	23.37	23.75
Manhattan Accuracy	21.50	22.37	24.12	24.25	24.50	24.12	23.37	23.37	22.87	23.25	24.25

Table 1: 32x322 images fitted

K=	3	5	7	9	11	13	15	17	19	21	23
Euclidean Accuracy	21.12	20.00	21.62	23.25	23.87	23.50	23.62	23.62	23.12	23.75	23.62
Manhattan Accuracy	20.62	22.62	22.37	23.00	23.37	24.00	24.12	23.62	24.00	24.37	23.25

Table 2: 64x645 images fitted

As it can be seen from the tables above, we got best results from k values around 15 to 19. It can be seen in the tables 1-2 that the accuracy of the model using Manhattan Distance metric gives similar results to Euclidean Distance. The time it takes to calculate the accuracy using the Manhattan distance was 226s for 32x322 resolution and 1016s for 64x645 resolution while the Euclidean metrics intervals are 583s for 32x322 and 2805s for 64x645. It seems plausible to say that using Manhattan Distance metric is more reasonable since it decreases the calculation time significantly without changing the overall result remarkably.

Nonetheless, using the kNN algorithm on an image dataset seems useless as we should convert our image matrix into a line vector to use it with kNN algorithm. This converting process causes a loss in features. However, it seems more reasonable to use the kNN algorithm for the telemetries data acquired from the metadata file. We will try to implement it in the second phase.

2.1.2 Logistic Regression Results:

Iterations=	*100*	*175*	*200*	*225*	*500*
Train Accuracy	46.5%	67.3%	72.8%	78.2%	99.9%
Test Accuracy	26.1%	23.6%	24.2%	21.5%	18.4%

Table 3: 64x645 images fitted

Iterations=	*100*	*140*
Train Accuracy	59.9%	70.8%
Test Accuracy	27.2%	28.1%

Table 4: Full scale images fitted

As it can be seen from the tables above, as the iterations increase train accuracy also increases. Yet, this can cause overfitting causing the test accuracy to decrease. So we tried to tune the algorithm by changing the number of iterations. We fit out models for both downsampled and full scale image dataset. Tuning the model for the full scale images was harder since training the data takes much more time. Maksimum test accuracy we obtained was on full scale images with 28.1% accuracy. These results are not satisfactory but expected since we fit our model with image datas. Those models can't relate the relationships between pixel distances to each other. So, in the feature we will fit those models with meta-data instead of mel spectrogram images to obtain better results.

2.1.3 Naive Bayes Results:

We used Categorical Naive Bayes and Multinomial Naive Bayes algorithms to predict the genres of our image data. Their training accuracies are 0.858, 0.294 and their test performance are 0.261, 0.278 accordingly. The alpha (Laplacian Smoothing Coefficient) has no effect on the prediction performance since we do not have any zero values in our data. Also, the prior probabilities have no effect on the accuracy as our data is uniformly distributed among genre classes.

3 What's left?

In addition to the audio files, our data also consists of metadata which has interesting and useful features. To give an example, this metadata consists of statistical features such as kurtosis, max, mean, median, min, skew, etc. Also, there are features extracted from Echonest software such as acousticness, danceability, energy, instrumentalness, liveness, speechiness, tempo, valence. We think that it would be a good idea to use this metadata in machine learning models such as kNN, Naive Bayes, Logistic Regression. The reason why we think like that is that Mel spectrogram image is not really effective in these algorithms and extracting the necessary aspects, which are in metadata in this case, would be a good idea for these models. In addition to that, we also want to improve our result on Mel spectrogram data. To achieve that goal, we are planning to build a CNN (Convolutional Neural Network) model and train our model using Mel spectrogram data. The reason why we have this plan is that we know image classification is a hard task in machine learning and one of the most useful and most effective ways of doing that is by using a CNN model. Once we have finished all of that, the rest will be tuning the parameters of our models for the best result possible.

4 Division of Work

Melih Berk Yılmaz: Worked on data preparation. Helped logistic regression and looked up solutions to overcome some problems related with data. Worked on the conversion of audio data to spectrograms. Took part in implementation of kNN algorithm.

Muhammed Şamil Arınç: Worked on the conversion of the data from audio file to mel spectrogram png file. Helped the implementation of the kNN algorithm. Implemented the Naive Bayes algorithms.

Mustafa Çağrı Durgut: Helped conversion of the data from audio files to spectrogram pngs. Did literature survey, reviewed and debugged the kNN algorithm and Naive Bayes algorithm.

Oğuzhan Özçelik: Worked on data reading and preparation. Did proof-reading on the kNN algorithm part. Wrote some part of the report. Found data structure related solutions to project's problems.

Yusuf Miraç Uyar: Worked on data preparation and reading, kNN algorithm, and Logistic Regression algorithm. Tuned those algorithms. Also helped on project progress report preparation.

5. References

[1] X. Bresson, M. Defferrard, K. Benzi, and P. Vandergheynst, “MDEFF/FMA: FMA: A dataset for Music Analysis,” GitHub. [Online]. Available: https://github.com/mdeff/fma. [Accessed: 28-Nov-2021].

[2] J. Choi, J. Lee, J. Park, and J. Nam, “Zero-shot learning for audio-based music classification and tagging,” arXiv.org, 19-Mar-2020. [Online]. Available: https://arxiv.org/abs/1907.02670. [Accessed: 28-Nov-2021].

[3] E. J. Humphrey, S. Durand, and B. McFee, “OpenMIC-2018: An open dataset for multiple instrument recognition,” 23-Sep-2018. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/248_Paper.pdf. [Accessed: 28-Nov-2021].

[4] Tzanetakis, George and Cook, Perry *Musical Genre Classification of Audio Signals* in IEEE Transactions on Speech and Audio Processing, vol. 10:293 - 302,2002.

Appendix

Link of our code (subject to change):
https://colab.research.google.com/drive/1mir2PSvSYCVv_cXQCY3vsXFzm__5Hf0L#scrollTo=3pHmZYkWsIVW.