Automatic Music Genre Classification

Ilay Yavlovich, Amit Karp
June 2023

1. Introduction

The music industry has gone digital in the past decade, now in a click of a button every user holds a huge music library which they may choose from. To keep us entertained and on their platform, companies develop algorithms that recommend new songs for their users that they predict will match their taste based on previous music preferences.

Algorithms like this are based on releases from the same artists and genre recognition.

In our project we aimed to develop a deep learning network that will classify songs by their respective genre. There is some work that has been done on this subject. In the "Genre Classification Using Pytorch – Tutorial" by Minz Won, Janne Spijkervet and Keunwoo Choi they used the GTZAN dataset to classify the music genre. This dataset is relatively small (10 genres, 100 songs with a 30-second-long sample each). We wanted to use a bigger dataset with full songs and more genres, that is why we have chosen the MTG-Jamendo dataset which contains 55,525 tracks annotated by 87 genre tags.

2. Methodology

We will use a CNN network based on previous works that have been done, then we will try to improve our results while examining two methods that we hope will improve our baseline results: using Majority Vote and using Weights. Because we are using full songs there are two ways to classify each song, the first is to look at the spectrogram of the song and look at where there is the most energy concentration, then taking a 30 second window surrounding this energy concentration and deciding based on this window the song genre. The second is to divide the song into an agreed upon segments and send throw the CNN all these segments. Then we will use Majority vote – creating a histogram of the classification of each segment and picking the one with the most votes.

2.1. Dataset

The MTG-Jamendo dataset, while being sizably lager then the GTZAN, contains a lot of multi label songs, meaning that each song can have up to 9 genres that will make our results hard to dissect. So, the first thing that we did is filter our data set to be all single label songs, our work can be seen in these images:

We have ended up with about a third of the original dataset, our work will be

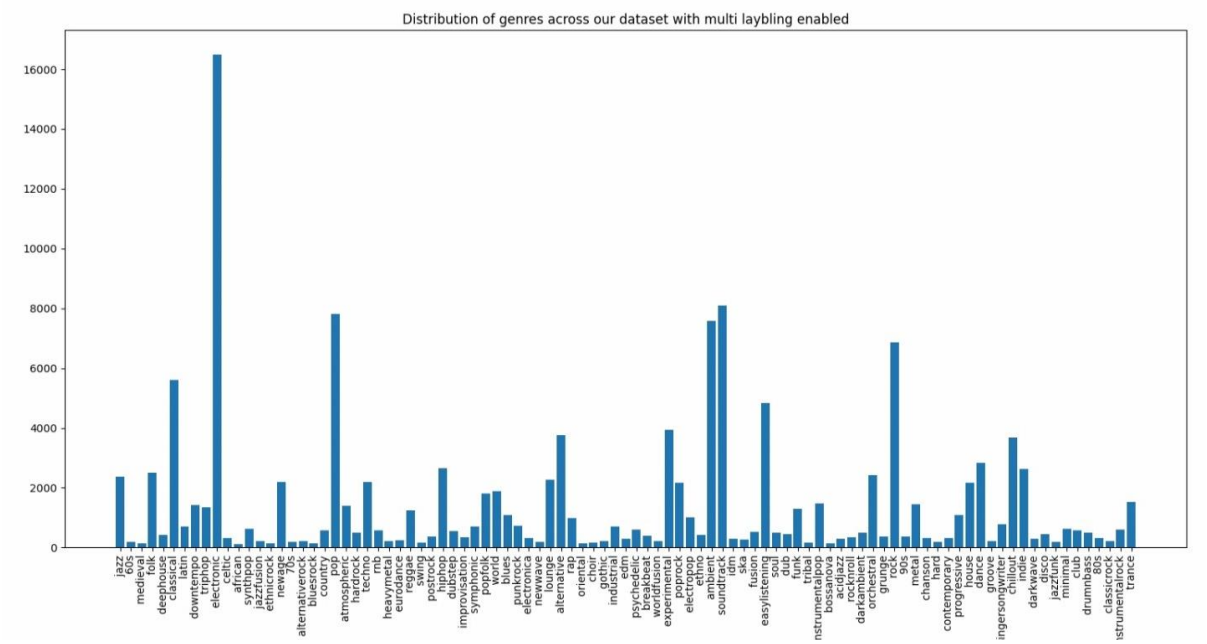


FIGURE 1 MULTI LABELED SONGS

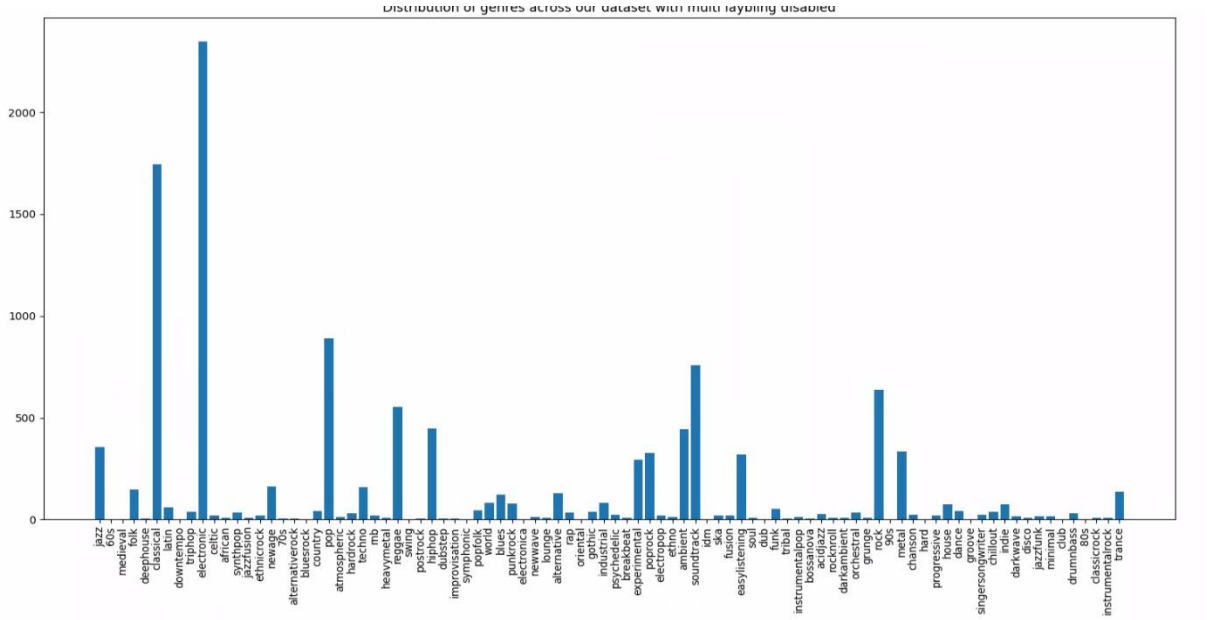


FIGURE 2 SINGLE LABELED SONGS

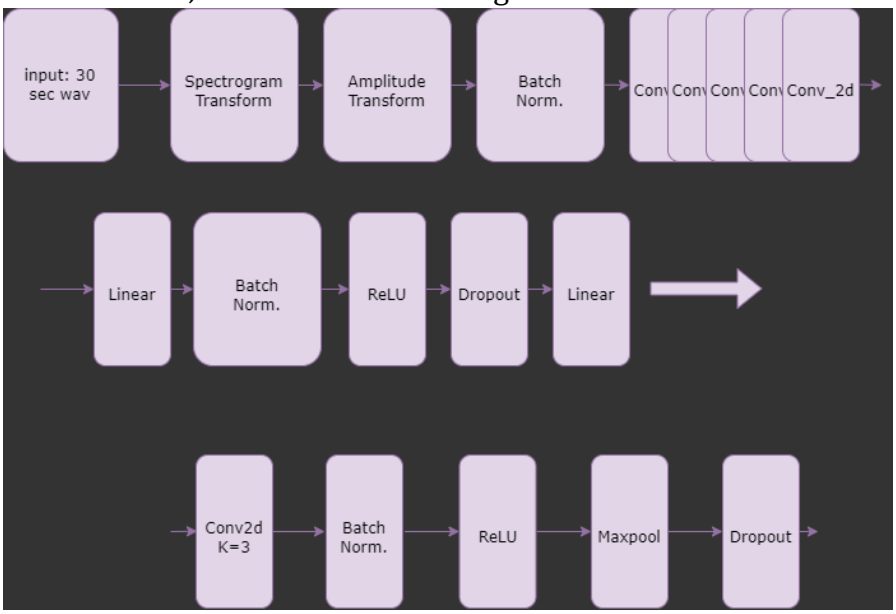
based on the top 10 genres so that we will have a more balanced dataset.

2.2.Augmentation

Our dataset now is much larger than the GTZAN dataset, but it is not balanced. We can see that the Electronic genre has around 2500 songs while the Jazz has around 400 songs. This unbalance in the dataset which can lead to a bias in our results that we want to avoid, to avoid this we tried to add weights to our CNN but we will show later that the result was not promising. We then took each song and split it into 30-second-long segments, each segment was then used to enlarge our dataset even more. All of the songs were in a .mp3 format that we changed to a .wav and all songs were turned into mono type music file for us to use on our architecture.

2.3. Architecture

We have used the architecture based on the work specified in the article mentioned above, we can see it in this figure:



We are using a 5-layer network each containing a Conv2d segment with a kernel size of 3 as stated in the figure.

FIGURE 3 A GENERAL LOOK OF OUR ARCHITECTURE

2.4. Algorithm

We will first augment our data, which includes all the steps mentioned above (change songs format, change audio to mono, splits the songs to segments). Then we will use the CNN above on our 10 largest genres on three instances (without any addition, with the Majority vote, with adding weights and with both weights) all with 30 epochs. We have chosen a 60-20-20% split between training, validation, and testing. We will make sure that segments that are from the same songs will be at the same split of our dataset to eliminate a situation where there are repeating parts along the song, and we would not want to train on one part and then test on the

other. We have chosen to use the cross-entropy loss function to test the accuracy of our model.

3. Results

3.1. Result without modification

This will be setup as our baseline for any further calculations. The first run was on our augmented dataset but without using majority vote or weights we got an accuracy score of 64% across 10 genres, on a same scenario that we have tested but on the top 3 genres we have gotten an accuracy score of 78%.

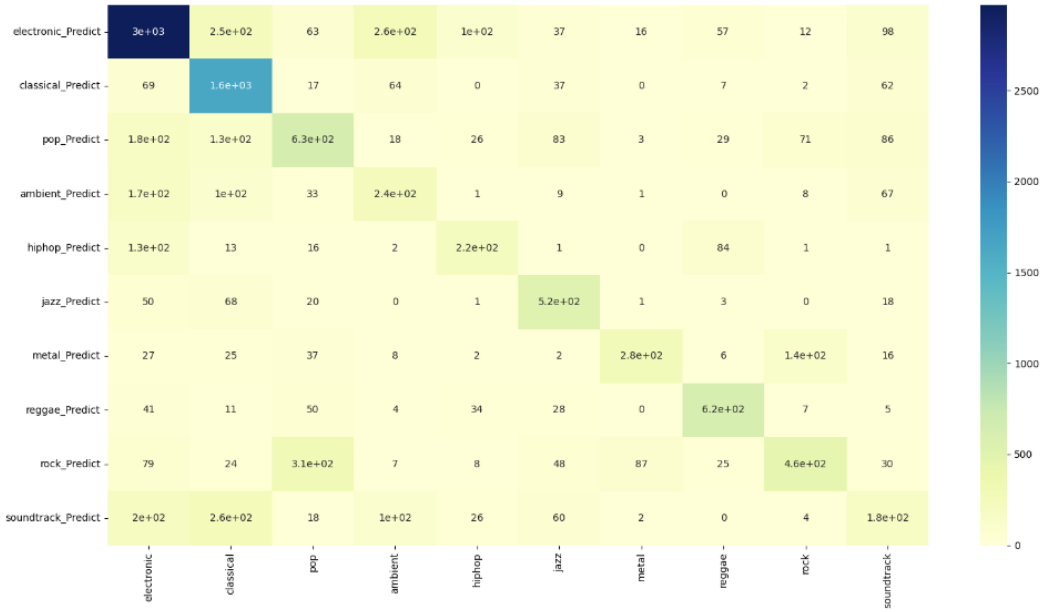


FIGURE 4 10 GENRES WITHOUT MODIFICATIONS

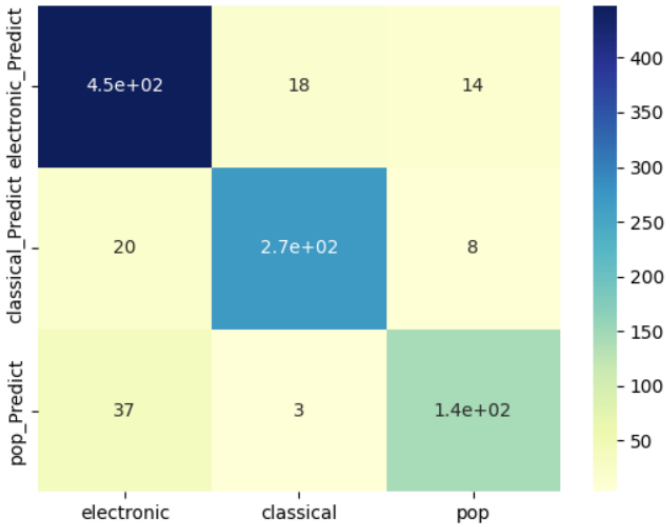


FIGURE 5 3 GENRES WITHOUT MODIFICATIONS

3.2. Result with majority vote

We got an accuracy score of 68% across 10 genres, on a same scenario that we have tested but on the top 3 genres we have gotten an accuracy score of 89%.

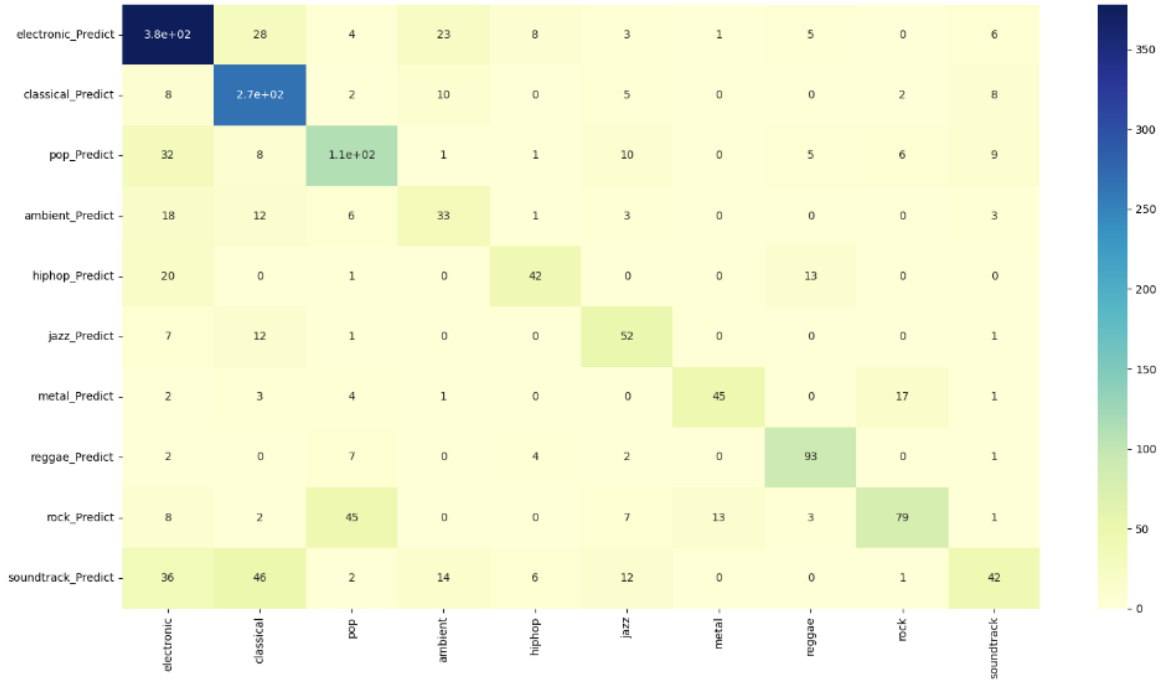


FIGURE 6 10 GENRES WITH MAJORITY VOTE

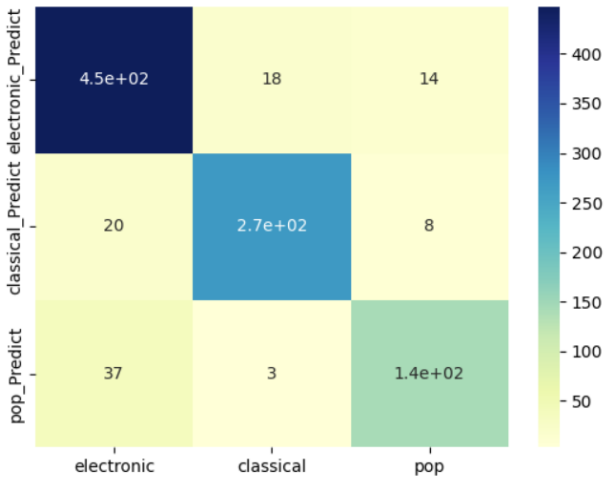


FIGURE 7 3 GENRE WITH MAJORITY VOTE

3.3. Result with weights

because our dataset is not balanced, we tried to balance it with weights. The way we chose to implement was to add a bias to a genre that is smaller on each chunk of data the data loader is choosing. We calculated the size of each genre, and then the weight of each genre is $\frac{1}{genre\ size}$, thus the bigger the genre the smaller the weight is, and we will choose less of them in the process of training. our result with this kind of weights were 30% and with the majority vote was 37%.

4. Conclusion and further work

4.1. Conclusion

Basing on our baseline of 64% accuracy we can say that the use of majority vote improves our results (to 68%) and the use of weights to balance our dataset is not. We have reasons to believe that the use of the weights damaged our results and a different way to balance the dataset may be more beneficial. We think that some of the genres that we choose are too ambiguous, the "soundtrack" genre for example is one of the top 10 genres that we have chosen but it is not a different genre from the rest, some of the soundtracks can be identified with other genres and because we give him more weight we can enlarge this error and this is viewed in the decrease in the accuracy score while using weights.

4.2. Further work

As we saw in the "Evaluation of CNN-based Automatic Music Tagging Models" article written by Minz Won, Andres Ferraro, Dmitry Bogdanov, Xavier Serra, the shorter the segments of music the better the results are. In the article they viewed a few segments divide and we want to do so as well, we have the framework to divide our database to 15-seconds segments and based on the results that we will get try to conclude if it will be better to divide our segments even more and see the improvement of accuracy score over time to compute. Try to choose different genres that are more obscure from each other and use them to classify with the same weights. We can even think of different weighting options like we have learned in class to see if we get different results. We would also like to classify songs with multi labeled genres using an assortment of softmax and a different way to use the majority vote and maybe look on the top 3 genres classified and for them if it's over some percentage threshold that we have chosen then we will classify it as the music genre.

5. References

5.1. Minz Won, Janna Spijkervet, Keunwoo Choi. *"Music Classification: Beyond Supervised Learning, Towards Real-world Applications"*. 2021.

5.2. Ndiatenda Ndou, Ritesh Ajoodha, Ashwini Jadhav. *"Music Genre Classification: A Review of Deep-Learning and Traditional Machine-learning Approaches"*. 2021.

5.3. Nikki Pelchat, Craig M. Gelowitz. *"Neural Network Music Genre Classification"*. 2020.

5.4. Minz Won, Andres Ferraro, Dmitry Boganov, Xavier Serra. *"Evaluation of CNN-based Automatic Music Tagging Models".* 2020.