Music Genre Classification

Atakan Ayyıldız[1] **Göktuğ Candemir**[1] **Ahmet Kasım Toptas**[1]

Abstract

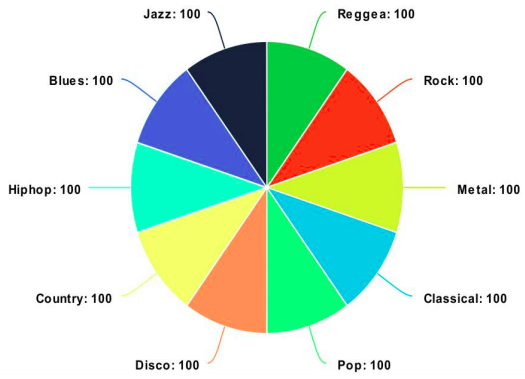
Music is a hot topic for years and many researches have been done for it. Because music has always occupied a great place in human life and when it comes to music, the first thing to do is classification. Therefore in this project, we mainly try to classification the music according to their genre using different machine learning algorithms and artificial neural network.

In this article, we talk about music classification and the results we have achieved. We want to determine the musical genres of the songs using machine learning and deep learning methods. There are a thousand songs in our data. And we have 10 classes. Our data is a balanced dataset. In the feature extraction phase, we processed 30-second songs and obtained certain features and we made classification by using these features in machine learning algorithms. Previously on progress report we achieved a predictive value of 60 percent as an average accuracy with our machine learning models. But now we achieved very high test accuracy(96%).

1 INTRODUCTION

Music has thousands of years of history, has become an indispensable part of human life, and has been called the 'food of the soul' among people. As it has been a popular subject throughout history, music is a popular subject today and many researches and studies have been done on it. In this project, we tried to classify different music genres. The first study on this subject was the 'Musical Genre Classification of Audio Signals' study by George Tzanetakis and Perry Cook in 2002. This study has been a reference for many studies from now on. GTZAN data was used in this study and many other studies. We also used the GTZAN dataset in our project. Our goal in selecting this dataset is to make it more meaningful to compare with previously obtained results. GTZAN dataset is a balance dataset. Dataset includes 10 classes and there are 100 tracks for each class. It is not reasonable to take all the music for the feature extract phase and it does not give good results as a result of machine learning algorithms. Therefore, 30 seconds of the music was taken from the middle of its duration.

In this project, we mainly try to classification the music according to their genre using different machine learning algorithms such as KNN (K-nearest neighbor), Logistic Regression , SVM (Support Vector Machine), Random Forest, LSVM (Linear Support Vector Machine) and artificial neural network. We achieved a predictive value of 60 percent as an average accuracy with our machine learning models. This was our first approach and we failed on this. Hence we need a completely different approach using deep learning and its extracted features. The detailed explanation is in the Methodology section. In order to explain this approach with simply we divided the songs into ten equal pieces then we obtained ten classification results for each song, then we combined them to get one classification result. This gives very higher accuracy in both deep learning and traditional machine learning algorithms than our previous work.



2 Related Works

In this music genre classification area there has many publications and every new publication continued adding something to the previous one. It started with "Musical Genre Classification of Audio Signals" written by George Tzanetakis.(1) In this paper the main approach is to characterize common typical rhythmic structure, instrument signals. The signal processing was handled on GTZAN dataset. Different features such as MFCC and Spectral Flux were extracted. As a result it gives 61% accuracy. Even though it did not give accurate results it is seen as a touchstone by many. Because the newcomers progressed by adding on this publication.

Second related work is "Music Genre Classification using Machine Learning Techniques" written by Hareesh

Bahuleyan(2), published in 2018. Here the dataset was extracted from ten second YouTube videos (a huge dataset), each wav file has 880 KB in total 34GB. In the first phase VGG-16 was used in the deep learning part. Fine tuning and transfer learning was used while implementing the pre-trained model without hand-crafted features. On the second phase features extracted manually as in the previous article, but this time features extracted with librosa libray which we will have used in our work. After that these extracted features were trained on traditional classifiers. In this paper MFCC features were considered as most important features and we note that for our work.
Third one is "Music Genre Classification With Machine Learning Techniques" written by Ali KARATANA and Oktay YILDIZ in 2017(3). Using again GTZAN dataset signals are processed to extract the features, these features are the same as the second related work, then traditional machine learning algorithms are implemented. Even though it was a simple study, this paper indicates that choosing the right features are important. Choosing only rhythmic features gives good accuracy only on Reggae and HipHop.

In 2014, Sander et al. (4) took a different approach than what had been done so far. Music information retrieval tasks were often solved using traditional feature extraction. Therefore, they carried out a study on the mid-level representation of the musical sound as a new approach for that time. They used spectrograms as an example. In doing so, they also used the lower level of the musical sound—raw audio signals—and found that spectrograms performed better than raw audio signals. But when using spectrograms, they used the entire song content. Thus, they investigated whether it is possible to learn end-to-end for music sound using convolutional neural networks for music classification. They used the Magnatagatune dataset (5).

In this study by Zhang et al. in 2016(6), deep neural networks were used and they used GTZAN dataset. Short-time Fourier transforms of each song are used with CNN in this article. While doing this, they took the songs as 3-second pieces and increased the data size. They also summed up the probabilities of the parts of each song and obtained a result. They tried to extract more statistical data from the features using avg-pool and max-pool when using CNN. In addition, residual connections were also used in the study and the results were compared. As a result, they found that they could improve the results of music classification by using pooling and using residual connections.

In this study, Ceylan et al.(7) made a music prediction study using the gtzan dataset and used CNN while using it,in 2021. This study is similar to the study of Zhang et al. Differently, they used mel frequency cepstral coefficients (MFCC). Finally, they had better classification results than Zhang et al.(6)

3 Methodology

3.1 Classification Methods

3.1.1. NEAREST NEIGHBORS CLASSIFIER:

The KNN algorithm is a very simple and powerful algorithm that is also used for classification and regression. KNN algorithm does not give very good results in case of too many features. Therefore, the number of features should be reduced before being used in datasets containing too many features. In order for the algorithm to work better, we determined the best hyperparameters using gridsearchcv from python sklearn and trained the model with these parameters.

3.1.2. LOGISTIC REGRESSION:

Logistic regression is an algorithm that is frequently used in classification problems and does this with probability calculations.Logistic regression is generally preferred for binary classification problems.That's why we performed the classification process using the multinomial logistic regression algorithm.

3.1.3. RANDOM FOREST:

Random forests is a supervised learning algorithm. The random forest algorithm works well when you have both categorical and numerical features. Random forest works well without scaling the input features. But random forest requires too much computation power and finding optimal values with grid search takes a lot of time.

3.1.4. LinearSVC:

Supports both dense and sparse input and the multi-class support is handled according to a one-vs-the-rest scheme. SVM works relatively well when there is a clear margin of separation between classes. SVM is relatively memory efficient.

3.1.5. SVC:

SVM is one of the supervised learning methods .Frequently used in classification and regression problems. SVM is a space-based machine learning algorithm that aims to separate data by drawing decision boundaries. It has a complex structure but can be used in small and medium sized datasets. It can also be used in datasets containing a large number of features.We standardization on the dataset before putting it into the algorithm. Difference between SVC and LSVC, SVC use kernel function so the model

learned a more nonlinear decision boundary.

3.1.6. ARTIFICIAL NEURAL NETWORK

In addition to machine learning algorithms, we also performed the classification process using the artificial neural network. Artificial neural networks consist of connections between nodes and It is designed inspired by the human brain. It is expected to give better results than standard machine learning algorithms.

3.2 Before CNN

We have seen some machine learning algorithms which are used for music genre classification in previous works so we are trying to get better results using these algorithms. We split data into 3 sets as train,validation and test. We used data as 80% for train and %20 for testing sets. We converted musical data to numerical data using Librosa(8) library which is very commonly used for music and sound analysis in Python language. We have features for our machine learning models and we hoped to hyper-tune our models to get the best possible parameters with grid search cross-validation. We used Scikit-learn for machine learning task.

Our feature size is 57. We have Chroma Features(9), Spectral Features, root-mean-square energy (RMSE), Zero Crossing Rate, tempo and MFCC features. Since we have continuous data, we have to use those features statically. So we take the mean and variance of those features. We used those features in our machine learning models with default parameters. After, we used those features in our models without standardization. We got terrible results with that approach. Our average accuracy was 20%. Then, we used standardization. We fit our training set to standardization. And we got transformations of all sets with respect to the training set. Hence, we get a noticeable increase. Our average accuracy becomes 65%. Then, we used grid search to get out optimum hyper-parameters. Our accuracy slightly increased.

Besides machine learning, we wondered what results we could achieve with deep learning. We used Keras for our neural network implementation. Since our data is small,as the complexity of our model grew, we expected that our neural network would be exposed to ”over-fitting”.

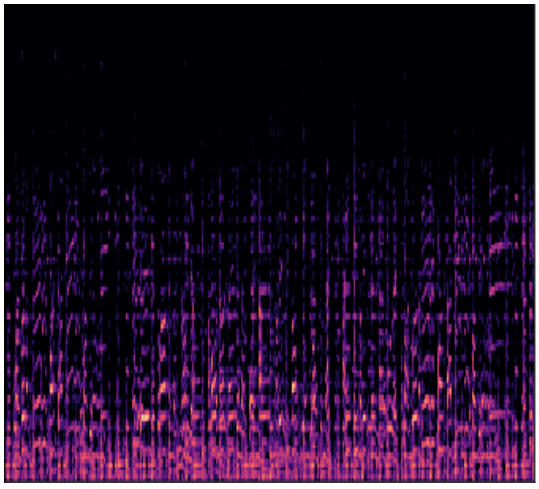
3.3 With CNN

3.3.1. Mel-Spectrogram:

Firstly we wanted to get mel-spectrograms and used them in CNN model. Mel Spectrogram is a visual that represents the signal strength (or amplitude) of a signal at various frequencies found in a given waveform. Librosa library used to extract the features in the CNN model.

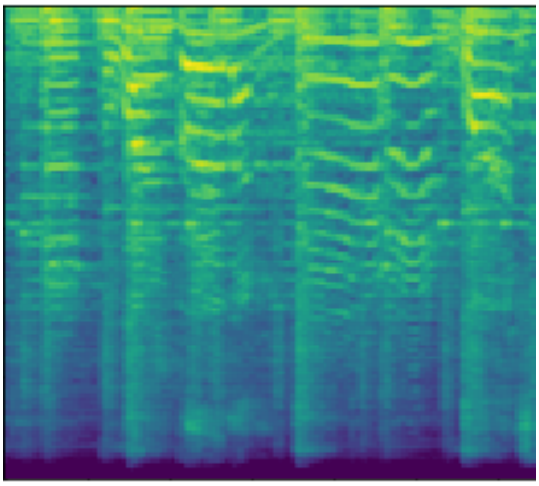
3.3.2. Our Method:

Since our dataset is relatively small, there are not many examples in our data. We only have 1000 samples, and when we pull out the spectrograms of all songs, that is, when we extract the 30-second spectrogram picture of each song, when we try to classify it in the CNN structure, we got very good results for training set but got bad results for our validation set. Our model was overfitted easily. So we want to get mel-spectrograms of 3 second pieces of each song. We thought we would get more information for our CNN model.

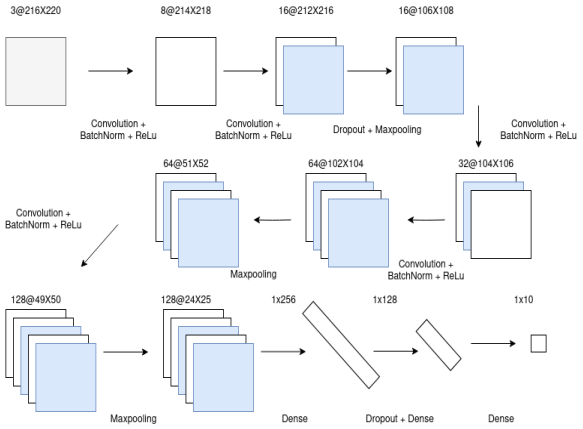


So, we came up with spectrograms for every 3-second track. This is a bit of a long process and requires patience. At the end of this process, since we have 1000 tracks for each class and 10 classes in total, we have a dataset consisting of 10,000 tracks. Then we split our tracks into 6000,2000,2000 pieces for our **training,validation and test sets**. But we keep each song’s parts in just one set. Thus, we do not receive information from the test and validation sets.

When we look at the pictures that come out as a result of the spectrogram extraction process, since these white spaces are common in all these pictures, we cropped the picture to make it easier to distinguish and clear the picture from the white paddings. As an example, the process is seen in the pictures below.



We have previously obtained accuracy values using traditional machine learning algorithms and the features we have. We had argued that we could increase these values even higher if we did feature extraction ourselves. Also we want to use our 3 second spectrograms in CNN with dense layer. So we want to get accuracy results from our CNN with dense layer.

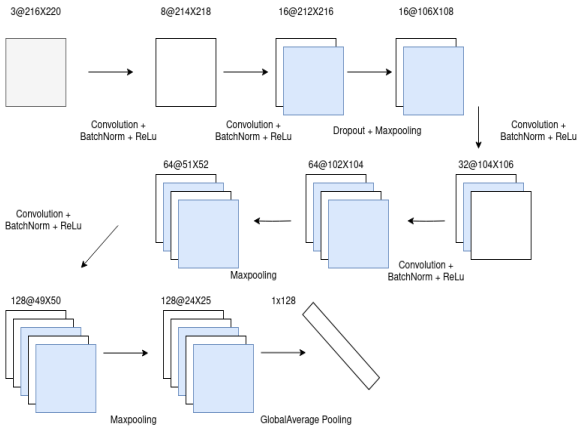


It is our CNN model with dense layers. We use that model for feature extraction and classifying 3 second spectrograms.

But we have some problems with our 3 second spectrograms. If we put input as 3 second spectrograms, we would get prediction for 3 second spectrograms. So we got our accuracies for all of 3 second spectrograms. We thought if we merge features of 3 second spectrograms with ordered, we would get higher accuracy for 30 second song in feature extraction. Also we thought that we can sum our output probabilities of each 3 second spectrograms for each song for our CNN classification task.

When making predictions in our CNN model, each music consists of 30 seconds, that is,3-second spectrograms of 10 piece. Therefore, we collected the probability values of every 10 spectrograms from the result of CNN with Dense Layers which use output activation function as Softmax activation function when predicting. Then we take the maximum of these total probability values and make a prediction out of every 10 spectrograms. Because each song has 10 spectrograms.

In feature extraction we have to merge our features from 3 second spectrograms. Hence, we get 3 second spectrograms features with order(each song has ordered 3 second spectrograms total of 10 piece) for our train,validation,test sets. We added global average layer after our last Max Pooling layer so, we got 128 dimension feature for each 3 second spectrograms. We did it because of we want to get less features for each song. Thus, we have 128*10=1280 features for each song. In this way, we extracted features using the CNN model from the spectrograms, and we retrained the models using these new features.But we have to tell that we used our training set as 800 songs and test set as 200 songs in our firstly basic machine learning models. On the other hand, we used 600 songs for training set and 200 song for test set. Because we dropped 200 songs for validation set in our CNN classification.



4 Experimental Results

4.1 Before CNN

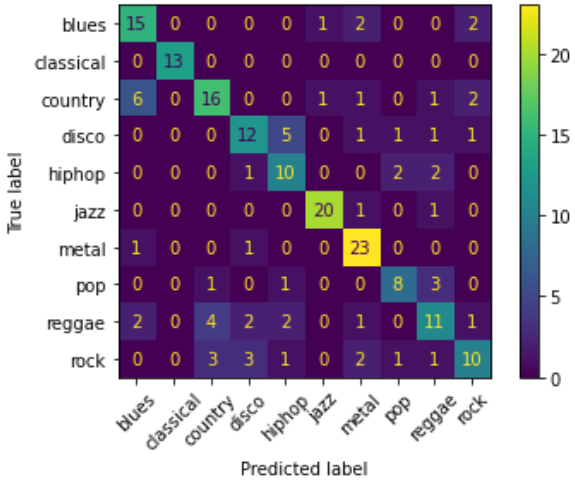
We used 5 machine learning classification algorithms namely KNN, SVM, Logistic Regression, Linear SVC, and Random Forest. We trained all algorithms using the same dataset, and we trained them with the most optimum parameters by performing hyper parameters tuning. The results obtained with the models created with machine learning algorithms turned out to be very close to each other.

The result we achieved with the artificial neural network gave better results than traditional machine learning algorithms. But when we increased the complexity of neural networks, our result began to be lower. So we created a simple model with 2 hidden layers.

Accuracy performance is as in the table below.

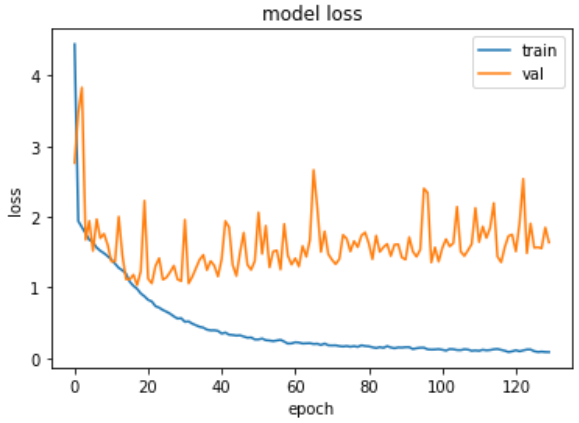
Model	Test Accuracy
KNN	0.69
Logistic Regression	0.7
Random Forest	0.69
SVM	0.7
Linear SVM	0.7
2 Hidden NN	0.77.5

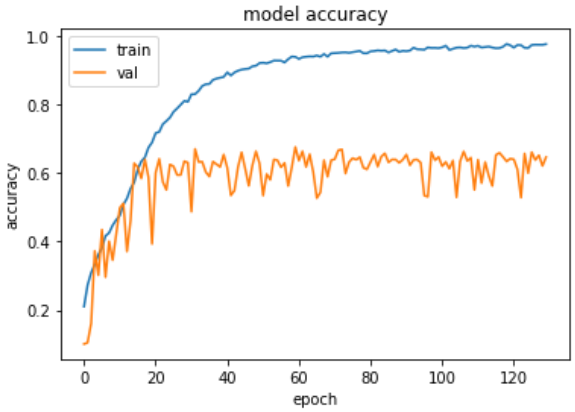
Random Forest and KNN models train accuracy was approximately 99%, but we found that it got lower accuracy values on test data. This is an overfitting problem. We did not observe such a problem with other models.



4.2 With CNN

This time, we classified our 3 second spectrograms and obtained results with 10 piece of track for each song which method is explained in our methodology. We used Adam optimizer,32 for batch size and learning rate is 0.001. We have loss and accuracy graphics below.





According to our loss graph our model does not converge at all. So we can early stop (at epoch 15) in order to time efficiency and avoiding the over-fitting. But this loss values are not essential for our model implementation. Our approach was explained in the Methodology part. We initialize a checkpoint to save the best model with respect to the validation dataset.

As you can see, we obtained around 65-70% accuracy for validation and best loss for validation was around 1. But we have to say that that values for each 3 second tracks, so we used model outputs which is probabilities of which class the 3-second pieces belong to with and we sum those values for 10 pieces for 30 second songs. So when we add the softmax values of the 3-second length spectrogram results for each piece which is 3 second, the output of the class which it belongs to has the largest Softmax value in all of them. Hence, we took the class with the maximum value as the prediction. So we obtained 96.5% accuracy for our test set.

For the feature extraction, we used our feature extractor model which is explained in our methodology,too. We got features got 128 features for each song. So for the 30 second we merged 10 pieces and we obtained 1280 features. We trained the machine learning models by using our new features and we achieved results suitable for our purpose.

Model	Test Accuracy
KNN	78.5
Logistic Regression	94.0
Random Forest	94.0
SVM	93.5
Linear SVM	91.15
CNN with Dense L.	96.0

Accuracy value is not always sufficient for us to evaluate model results. Therefore, interpreting the result using multiple metrics instead of a single metric gives more reliable results. That's why we also calculated different metric values of precision,recall and f1-score.

This figure below belongs to the Random Forest algorithm with our extracted Mel spectrograms.

	precision	recall	f1-score	support
blues	0.95	0.90	0.93	21
classical	0.95	0.95	0.95	20
country	1.00	0.95	0.98	21
disco	1.00	0.87	0.93	23
hiphop	0.95	1.00	0.97	19
jazz	0.90	1.00	0.95	18
metal	0.95	1.00	0.97	19
pop	1.00	0.80	0.89	25
reggae	0.80	1.00	0.89	16
rock	0.90	1.00	0.95	18
accuracy			0.94	200
macro avg	0.94	0.95	0.94	200
weighted avg	0.95	0.94	0.94	200

5 Conclusion

In this project, we mainly focus on getting better accuracy results than the other related works. First of all, we use Librosa library and get 57 important features from the 30-second track. Some of those are Spectral Features, RMSE, MFCC, Zero Crossing Rate, tempo, etc. We created some traditional machine learning models and trained them using features that we have got but we saw that models are not good enough to get predict unseen test datasets. Those models give approximately 70% accuracy.We thought that could multilayer neural networks might give better results than machine learning models. We trained 2 hidden layer neural networks using the same features and get a %77.5 accuracy which is a relatively better result than machine learning models but it still not enough for our demands, our purpose. Therefore, we have seen that the features we have do not train our model well enough. So, we obtained new features by performing feature extraction using Mel spectrograms. For this, we thought that we could get much better results by dividing the 30-second tracks into smaller parts and using the spectrograms of these small parts, and we divided each 30-second tracks into 10 tracks of 3 seconds and produced the mel spectrogram of each piece. Using the 10,000 mel spectrograms we obtained in the CNN model, we trained the model sufficiently and completed the feature extraction process by taking the learned features from the model.

This time, we trained the machine learning models we used at the beginning of the project by using our new features and we achieved results suitable for our purpose. We obtained 100% accuracy for our training set, 96.5% accuracy for the validation set, and 96% accuracy for the test set. Those accuracy values are higher than other works which we looked at. Thus, we can say that the project offers a successful solution to the music genre classification problem in accordance with its purpose.

Our approach has some strengths and weaknesses.To give an example of strengths, our feature extraction model uses global averaging and it returns less features. It is good for our ML models.Also feature extraction with CNN works well.To give an example of its weaknesses, our KNN model can't predict as other ML models because of Curse Of Dimensionality. Also our CNN model overfits very fast so we have to be careful. Because when our model is overfitted, we predict songs "country" as "rock". All of our other predictions are correct.

Our final comparison for models

Our final comparison for models

Model	Test Accuracy with old Features	Test Accuracy with extracted Features
KNN	69.0	78.5
Logistic Regression	70.0	94.0
Random Forest	69.0	94.0
SVM	70.0	93.5
Linear SVM	70.0	91.15
2 Hidden NN	77.5	-
CNN with Dense L.	-	96.5

References

[1] G. Tzanetakis and P. Cook *"Musical Genre Classification of Audio Signals", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, vol. 10, no. 5, pp. 293-302, 2002.*

[2] H. Bahuleyan *"Music genre classification using machine learning techniques," arXiv preprint arXiv:1804.01149, 2018.*

[3] Karatana, A., Yıldız, O. *Music genre classification with machine learning techniques. 25th IEEE Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 1-4. 2017*

[4] S. Dieleman and B. Schrauwen *"End-to-end learning for music audio," in Acoustics, Speech and Signal Processing (ICASSP),2014 IEEE International Conference on. IEEE, 2014, pp. 6964–6968.*

[5] Edith Law and Luis von Ahn *"Input-agreement: a new mechanism for collecting data using human computation games," in Proceedings of the 27th international conference on Human factors in computing systems, 2009.*

[6] Zhang,W., Lei,W., Xu,X., Xing,X. *Improved music genre classification with convolutional neural networks. Proc. Interspeech 2016,3304-3308., http://doi.org/10.21437/Interspeech.2016-1236,2016*

[7] Ceylan,H., Hardalac, N., Kara,A., Hardalac, F *"Automatic Music Genre Classification and Its Relation with Music Education,",World*

Journal of Education Vol. 11, No. 2; 2021, https://doi.org/10.5430/wje.v11n2p36,2021

[8] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. *librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference, pp. 18–25. 2015.*

[9] M. Kattel, A. Nepal, A. Shah, and D. Shrestha *"Chroma feature extraction," 01 2019.*