Music Genres Classification via Deep Learning Models

Hassan O. Kandil, Andrew E. Ehab, *The American University in Cairo*

Supervised By: Mohamed Moustafa, PhD.

Abstract In this research paper, we propose our findings regarding the topic of music genres classification. We have applied machine learning techniques in order to be able to correctly classify the genre of a song. We used spectrograms in order to better represent the musical tracks. We also experimented with dividing the track to windows of different sizes. We experimented with multiple architecture as a Convolution Neural Network, Recurrent Neural Network and finally, a mix of a convolution network followed by a recurrent network (C-RNN). We found that a convolution network is able to outscore that of a recurrent network however, the winning architecture was the C-RNN that was able to even outscore the convolutional network but significantly increased the number of parameters.

I. INTRODUCTION

With the growth of online music databases and easy access to music content, people find it increasingly hard to manage the songs that they listen to. One way to categorize and organize songs is based on the genre. Music genres are categories that have arisen through a complex interplay of cultures, artists, and market forces to characterize similarities between compositions and organize music collections. Yet the boundaries between genres remain fuzzy, making the problem of music genre recognition (MGR) a nontrivial task. In addition, The dramatic increase in the amount of published music all over the web has created a two challenges:

- The need to automatically organize a collection
- The need to automatically recommend new songs to a user knowing their listening habits.

Thus, we need to is to be able to group songs in semantic categories using artificial intelligence techniques since being able to automatically classify and provide tags to the music present in a user's library, based on genre, would be beneficial for audio streaming services such as Spotify and iTunes. Accordingly, we proposed a solution to this problem as follows: given a music track, our aim is to classify its genre as one of 10 different genres through different deep learning models.

II. Related Works

Music genre classification has been a widely studied area of research and a lot of machine learning techniques have been used to solve this classify genres into a variant number of categories. A group of researchers at Stanford proposed a solution to the genres classification problem in which they applied a variety of machine learning techniques to classify 16 music genres. The different models they used were the SoftMax, Logistic Regression, SVM, the SVM RBF and the KNN. The model with the best accuracy used SVM RBF and achieved an accuracy of 68.07%. This models were trained using a very large dataset which is composed of 25,000 30-second tracks built from the FMA dataset [4]. A different approach to the problem was by another researcher from Stanford University using a VGG16 Convolutional Network Model, which is a very deep model with a large number of training parameters. Training was done using the popular GTZAN dataset. The best validation accuracy reached was around 86% [3].

III. Contribution

At the beginning of our research we find that the most widely used dataset for this problem was the GTZAN due to its small and compact size which made it easier to train. However, as explained by some researchers; this popularity was misleading to many people; since it has been used without properly examining its contents and the assessing its quality and the results it produced [1]. Accordingly, our research was, at first, targeting the use of a larger dataset. We choose the less explored FMA-Medium dataset because of the many different genres compared to the GTZAN as well as having 25K tracks in addition to its better reliability. However, the dataset was hard to deal with. FMA datasets are composed of mp3 (compared to .au format of the GTZAN) files that consume a lot of RAM. Also, the dataset is hugely unbalanced. Some genres had 27 tracks while others had 4K tracks (out of 25K total). We then shifted our

.

focus to the small variant of the FMA dataset which is balanced and had 8K total number of tracks. Unfortunately, due to limited computation resources, it was not possible to load the data and performing data augmentation properly.
However, not giving up to the idea of using the GTZAN, we decided to build up our own dataset. We refer to this dataset as GTMA which is of feasible size yet bigger than the GTZAN. We also wanted to experiment with different deep learning models on this dataset to compare between their results, asses them and reach the best accuracy possible using this new dataset.

IV. Methods and Procedures

A. *Mathematical Background and Pre-processing*

In order to use a 2D convolutional network, we needed to be able to represent the audio waveform in the form of an image. Existing literature pointed us to move the data to the Mel Spectrogram domain. To do so, after loading the second 10 seconds of the track, we divide it to a set of windows. Each window is the Fourier Transform to get the frequency components. We then map the frequencies to the mel-scale, modeling the human perception of the changes. Then we take the log of the values. Finally, take the discrete cosine transform, correlating the values of the frequency components. For the sake of noticing the difference between each genre, we present 2 images of 2 different genres. (see Figure 1, 2)

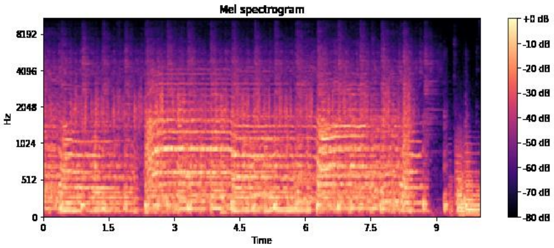

Fig. 1. Rock Genre Spectrogram

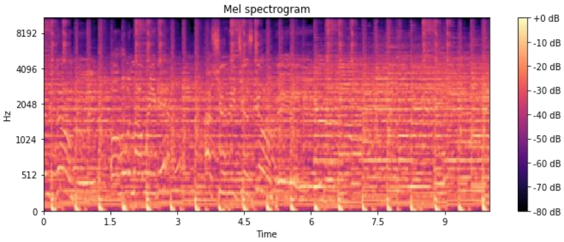

Fig. 2. Pop Genre Spectrogram

B. *Dataset*

Due to the lack of enough resources to load a large number of tracks from the heavy FMA dataset and in order to have a more reliable, balanced dataset, we created a new dataset and used it for training; the newly created dataset was a combination of GTZAN and FMA dataset [2]. Since the GTZAN and the FMA datasets have different genres labels, we needed to take different portions from different genres from the 2 datasets. It is composed in total of 2,500 tracks split into 10 different balanced genres, each track is 30 seconds. The dataset is divided into training, validation and testing sets with a ratio of 8:1:1.

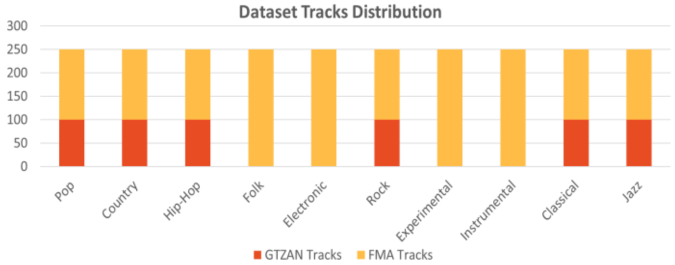

Fig. 3. Dataset Distribution

C. *Architecture*

We had three different models, in order to experiment with different architectures and find the best one (the one with the highest testing accuracy) suitable for classifying genres; where the input to each model was the preprocessed training data (floating points of the tracks) and each track was divided into 1.5 second windows:

Model A: CNN

As a first approach to the problem, we designed a CNN composed of 5 convolutional layers. As with images, the convolutional layers act as feature extractors which where each layer extracts more distinctive features from each window of the tracks through convolving with different filter sizes. Each layer is followed by a max-pooling layer in order to down sample the tracks. Then, in order to prevent overfitting problem, we added a dropout layer of 25% probability after each max-pooling. Finally, the features are fed into a fully connected neural network of 3 hidden layers is used and 1 output layer which outputs the final scores for the 10 classification genres (see Figure 3)

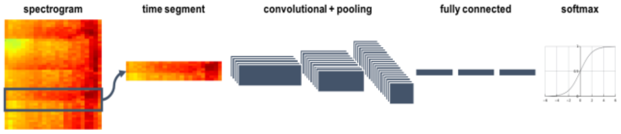

Fig. 4. CNN Model Diagram

Model B: RNN

As a different approach to the problem, instead of extracting features using the convolutional neural network, we tried adding the memory element to the classification problem in order to make the scores of the genres for a certain song depend on the previous windows of that track. So, a recurrent neural network was used in the form of a Long Short-Term Memory (LSTM) model. The model was composed of a LSTM layer of 128 neurons and a dropout of 50% and then the output of the LSTM layer was flattened and fed into a fully-connected neural network of 3-hidden layers with 150, 100, 50 neurons respectively and a final output layer for the scores of the classes.

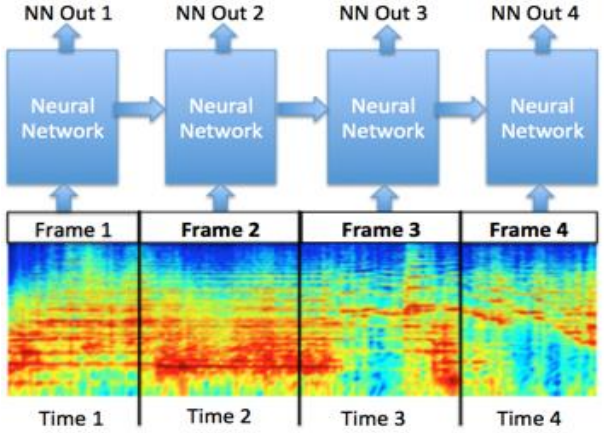

Fig. 5. RNN Model Diagram

Model C: C-RNN

After achieving good results with the RNN model approach (close to that of the CNN model), we tried merging between the two models to get the best results out of them both through a Convolutional Recurrent Neural Network; where the Convolutional Neural Network extracts the features of each track window; then these features are fed into an Recurrent Neural Network to make the classification decision depend on all the previous windows and then finally the output of the LSTM is fed into the Fully-Connected Neural Network to get the final classification scores of the classes. The CNN has 5 layers and the RNN is composed of one LSTM layer of 128 neurons; and finally, the FCNN is similar to the previous two approaches with 3 hidden layers (150,100,50 neurons respectively).

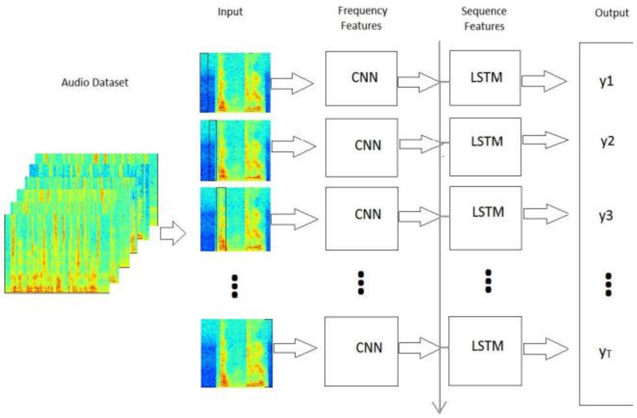

Fig. 6. C-RNN Model Diagram

V. Experimentation

In order to fine-tune our hyper-parameters to achieve the best possible results, we experimented by the following:

1. Removing and adding Convolutional Layers: removing convolutional layers decreased the accuracy significantly against the reported accuracy while adding more layer showed no real benefit. The accuracy didn't increase but the number of learnable parameters increased.
2. Removing and adding Fully Connected Layers: removing fully connected decreased accuracy as well. However, adding more layers showed increased accuracy.
3. Adding more filters in the convolution layers as well as the more neurons in the fully connected showed no benefit but increased the number of parameters.
4. We also experimented with changing the window size of the data. We started with 3 seconds window (following the literature) but experimentation showed that 1.5 second window yielded better results. We related this to the filters being able to capture finer details in the smaller window.
5. As for the RNN model, we tried adding an additional LSTM layer with less number of neurons, but it yielded less accuracies since the network become deeper which needed more data points and more epochs to train properly; we also tried different numbers for the size of the LSTM layer and settled on a number of neurons equal to the size of track frames (128 neurons) since it yielded the best results.

VI. Results And Discussion

Model A (CNN):

Using the CNN architecture mentioned above, the results achieved are promising. Whereas, a CNN network of almost 370K weights can predict the correct genre out of 10 possible genres with an accuracy 73%.

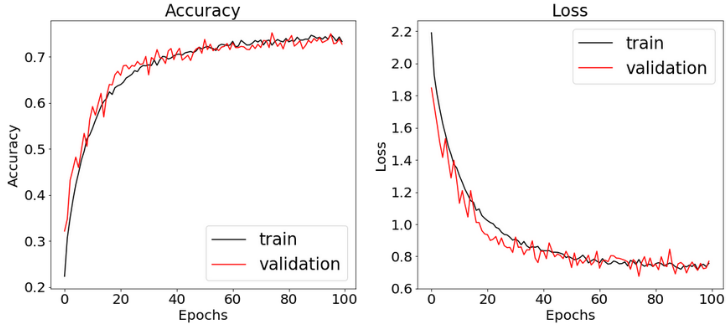

Fig. 7: Results of CNN network

Model B (RNN):

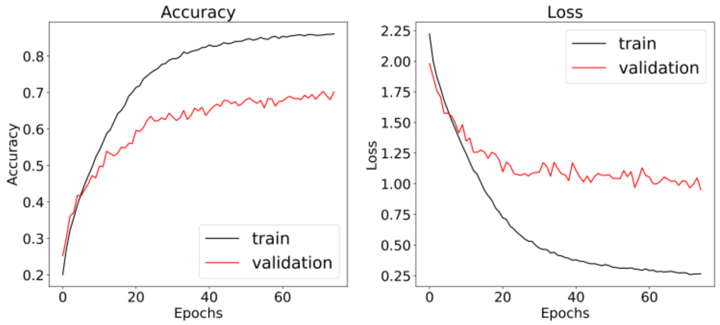

Fig 8: results of RNN network

Using an RNN architecture, the network reached an accuracy of 68.66% which is considered good but cannot outscore the CNN architecture. The loss was high.

Model C: (CRNN)

Finally, we combined the previous 2 networks to form a convolution network followed by a RNN network as an experiment. This has proved to provide superior results. It outscored both the RNN and the CNN architectures. It reached a testing accuracy of 77.2%

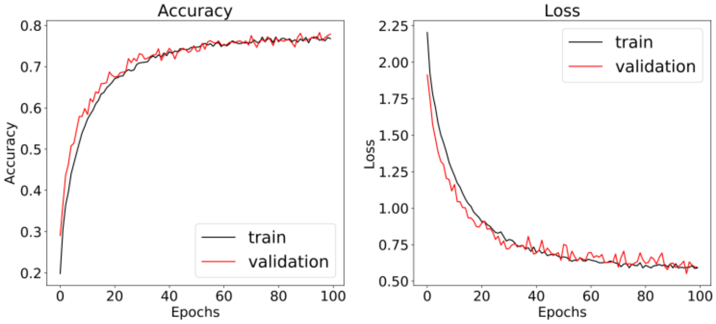

Fig 9: Results of CRNN network

Table 1. Different models results

Model Name	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy	Testing Accuracy
CNN	0.701	0.761	73.23%	75.16%	73.0%
RNN	0.2587	0.967	85.82%	70.32%	68.66%
C-RNN	0.5931	0.551	76.37%	78.27%	77.2%

VII. Conclusion

Researchers should build up a proper, balanced and updated dataset. Also, Convolutional networks have proven to be powerful in extracting features from musical tracks by simply looking at how they are represented in the Mel spectrogram domain. Also, dropouts have proven to be extremely important to fight the overfitting problem. Moreover, the use of the "ELU" activation function has also greatly impacted the accuracy to reach high values faster

VIII. Future Work

Future researchers should try to experiment with a deeper network architecture. However, this should come from after building up a bigger dataset. Researchers should also try experimenting with an ensemble network with 1D convolution. The network should also be tested against more reliable data, preferably coming from FMA Datasets.

References

1. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use Bob L. Sturm, June 10, 2013
2. FMA: A DATASET FOR MUSIC ANALYSIS Michaël Defferrard Kirell Benzi Pierre Vandergheynst Xavier Bresson
3. De Eguino, M. (2017). Deep Music Genre. Stanford University.
4. Guo, I., Gu,Z., and Liu,T., "Music Genre Classification via Machine Learning," CS 299 Final Project, 2017.