# Music Artist Classification Using CRNN

Sarthak Tandon, *Syracuse University*
Syracuse, NY, USA
stando01@syr.edu

Dhrumil Shah, *Syracuse University*
Syracuse, NY, USA
dshah13@syr.edu

***Dataset link:*** ***https://www.kaggle.com/dhrumil140396/mp3s32k***

## Abstract

There have been many attempts at developing machine learning and deep learning algorithms for the task of identifying artists based on the audio file. In this paper we have developed an algorithm which uses mel-spectrogram from audio files and applies different CRNN architectures for artist detection tasks. We are not able to improve the performance of the state-of-the art CRNN architectures for the same task but came very close. We have used frame level features and evaluated models using both song level split and album level split for different time windows.Our model achieves an average F1 score of around 67.92%.

## 1.Introduction

There are many tasks which are related to audio signal processing which have been explored by researchers . Some of those tasks include genre classification, song identification, chord recognition, sound event detection, mood detection and feature extraction. These kinds of problems fall under the umbrella of music information retrieval. In the past decade an "App called Shazam developed music detection algorithms but those algorithms are based on database search"[1], "so those algorithms require a large amount of database lookups and in - turn lots of computation efforts"[3]. Contrary to what Shazam does, humans are significantly good at identifying songs and artists given enough data . "This ability to process and identify with music has roots in neuroscience" [4] and thus an algorithmic representation would work best if it included an analogous learning model .

There are many algorithms based on machine learning techniques which are developed for artist detection tasks. "But those algorithms use low dimension feature vectors to summarize the audio signals because those models suffer from the curse of dimensionality"[1]. We will compare the performance of all the machine learning models and deep learning based models in the next section. One limitation those models encountered was that there are very few sample songs per artist, but human beings are very good at these tasks despite limited sample size .There for successfully solving the problem of music -artist detection would automate many tasks in the area of Music Information retrieval.

## 2. Previous Works

Before we analyse the performance of the models let's understand some basic terms.

- **Frame level features** : Frame level features consider the patterns contained in the very short period time frame of the audio file.
- **Song level features** : Song level feature considers the entire song in order to detect the artists. For example let's say 2 songs which have the same MFCC graph for a given time frame but those MFCC are out of order . Then according to song level features both songs are the same.
- **Song based split**: While splitting the data into training , testing and validation sets

we split all songs randomly irrespective of their album.

- **Album based split**:When spitting the data into training,testing and validation set considering all the songs in the album x percent of songs should go to training , y percent should go to testing and z percent of songs should go to validation set, where x+y+z=100.

In the past there have been numerous attempts to solve the problem of music artist detection using machine learning and deep learning . In this section we will talk about those attempts:

1) "Whitman et al. have used SVM on the MFCC images of the songs. Dataset used by them is not public but it consists of 21 artists . They got around 50 percent accuracy with SVM. Music consists of mainly 2 kinds of patterns.
    - Artist's music style
    - Pattern Associated with Album

   As per the analysis by authors model with SVM was not able to address the patterns related to albums. "[10]
2) "Fixing the problems of previous models , Daniel P. W. Ellis at Columbia university used full-covariance gaussian classifier to achieve the accuracy of around 59 percent . He also released the artist20 dataset which is used by almost all the future researchers who are working in the music information retrieval domain . " [11]
3) "All the models we have discussed previously used frame level features. Madel et al. have studied the impact of using song level features and frame level features using SVM and GMM . According to their analysis models with song level features outperform the models with frame level features." [12]
4) "Just one year before Zain et al. have used convolutional recurrent neural networks for the task of artist detection. They have used artists 20 dataset and used frame level features and song level features. In this project we are using frame level features so we will use the results of their models with frame level features as a baseline."[1]

Here is the summary of all the model performances:

| **Authors** | **Feature Level** | **Split Type** | **Model** | **Accuracy** |
|---|---|---|---|---|
| Whitman | Frame | Song | SVM | 0.5 |
| Ellis | Frame | Album | GMM | 0.59 |
| Mandel | Frame | Album | GMM | 0.541 |
| Mandel | Frame | Album | SVM | 0.687 |
| Mandel | Frame | Song | SVM | 0.839 |

(Source :[1])

Baseline Model Results with frame level features :

| **Split** | **F1 score 1s** | **F1 score 3s** | **F1 score 5s** |
|---|---|---|---|
| Song | 0.729 | 0.765 | 0.770 |
| Album | 0.482 | 0.513 | 0.536 |

(Source :[1])

---

## 3. Dataset & Preprocessing

The dataset which we are using is the artist20 which was created by Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University. It includes songs for twenty artists, each having 6 albums with songs encompassing a range of musical styles. Some of the Dataset specifications are as follows:

| Features | Values |
|---|---|
| Size | 1.29 GB |
| No. of Tracks | 1413 |
| No. of Artists | 20 |
| Bitrate | 32kbps |
| Albums per Artists | 6 |

In the past, due to the curse of dimensionality and lack of extensive compute resources, classification was done with low dimensional features with past work leaning towards vector summaries of frequency content in an audio window. Due to this researchers in the past worked with Mel-frequency cepstral coefficient (MFCC) representation of the songs. The issue with this approach is that it loses the temporal structure of the audio. Given this and the fact that we wanted to experiment with CNN architecture, we decided to use the Mel-Spectrogram transformation for our data set.

"A spectrogram is a representation of frequency content over time found by taking the squared magnitude of the Fourier Transform of a signal."[6] The following steps take place when transforming an audio file to get a mel spectrogram:

1. We take the audio sample over time to represent an audio signal which is then mapped from time to the frequency domain using the Fast Fourier Transform (FFT). This is performed on overlapping windowed segments of the signal.

2. We then adjust the frequency to a log scale and the amplitude to create a spectrogram.

3. Finally, we adjust the frequency to the mel scale to form a mel spectrogram.

The formula used Mel Scale and the decibel scale for these transforms are as follows:

$$m = 2595 \log_{10}(1 + f/700)$$

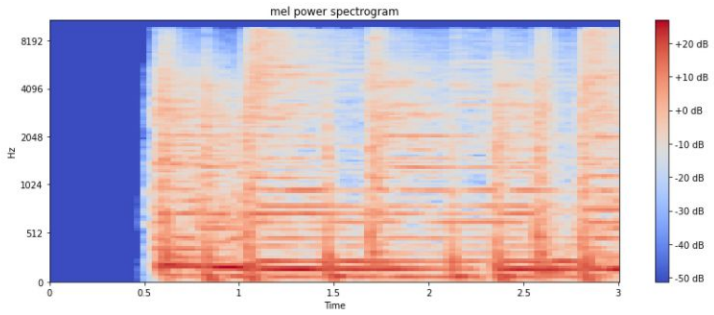
$$d = 10 \log_{10}(m/r)$$

Here *"f"* stands for frequency and *"r"* stands for reference power for log scaling.

The parameters used during these transformations are as follows:

| Parameters | Values |
|---|---|
| Sampling Rate | 16kHz |
| No. Mel Bins | 128 |
| FFT Window size | 2048 |
| Hop Length | 512 |
| Power for Log-Scaling | 1.0 |

Here is a Mel Spectrogram visual of a 3 sec sample of a random song from the dataset.




When looking at the ways of splitting the dataset into training, validation and testing, we decided to try two approaches namely split by song and split by album. Splitting data by songs involves the dataset to be split by stratified sampling on the

artists Splitting by albums involves randomly selecting two albums for each artist and putting them into testing and validation set, while sending the rest to the training set. Splitting by song keeps the salient production details for each album intact along with the music style. However, this may lead to inflated model performance when compared to splitting by album, as it is only able to remember the music style of a particular artist. For song split, the train, validation and test split is 89% / 9% / 10%.

Once the audio files have been transformed and split into different sets, we will slice the files into smaller samples of length *t* seconds. These slices samples will be the final dataset for our model on which it will be trained. However we need to clarify that we do not mention the time when slicing our files. The transformed files do not store information for every second, but for every frame. So for slice the songs to a desirable time, we mention *FR*t,* where FR is the frame rate for the audio file. This can be calculated using the sampling rate and the hop length, which results in the frame rate of 31 frames per sec.

Finally the artists names are label encoded followed by one hot encoding so that it can be read and understood by the model.

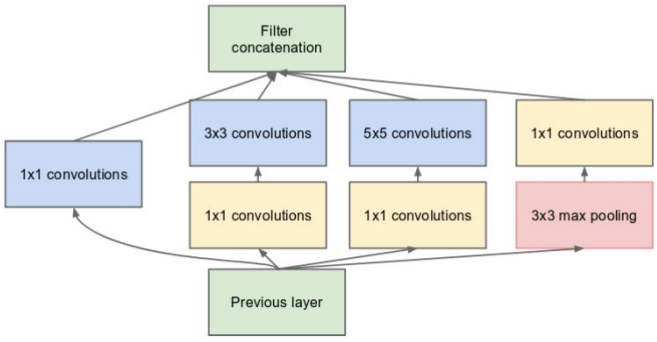
---

## 4. Model & Training Considerations

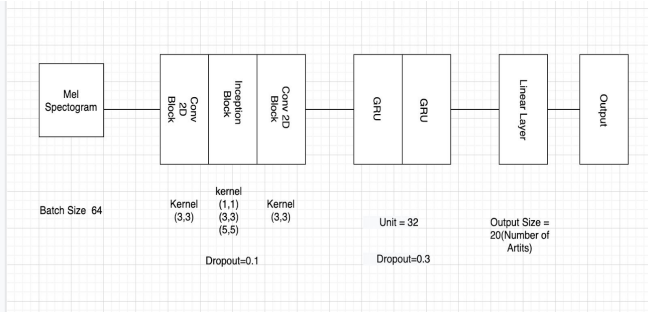
For this project, our aim was to improve the performance of artist detection compared to models which used MFCC as a preprocessing technique and beat the state of the art models which use the mel-spectrogram images and CRNN network . We have used mel-spectrogram as a preprocessing technique for our model because we think that neural networks would be able to learn better by considering how frequency content changes over time and also considering all the temporal features . CRNN networks excel at this task , where convolution networks detect the frequency pattern structures and recurrent networks detect any temporal pattern in the mel-spectrogram image. In this project we have modified the convolution part of the previous state of the art CRNN architecture with new techniques in convolution like Inception and ResNet.

### 4.1 CRNN - Using Inception Block

Prior to inception , inorder to improve the performance of the neural network models people would use deeper models which suffer from a vanishing gradient problem and makes the training of the neural network difficult . Invention of inception net improved the accuracy and training time which motivated us to try inception blocks for music artist detection problems. “Salient parts in the image can have extremely large variation in size . Which makes it extremely difficult to choose right kernel size . Larger kernel size detects the features which are distributed more globally while small kernel size detects the features which are distributed more locally. Inception blocks solve this problem by choosing the multiple kernel size which operate on the same level and output of all those kernels are all stacked together as an input for the next layer. “[5] So rather than using the entire inception net we have used inception blocks for our network. In the next few images we will see the architecture of basic inception block and architecture of the model for artist detection.(1 image taken from [5])



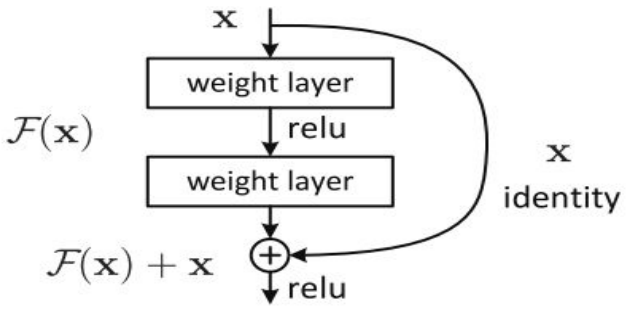

(b) Inception module with dimension reductions




Network Architecture

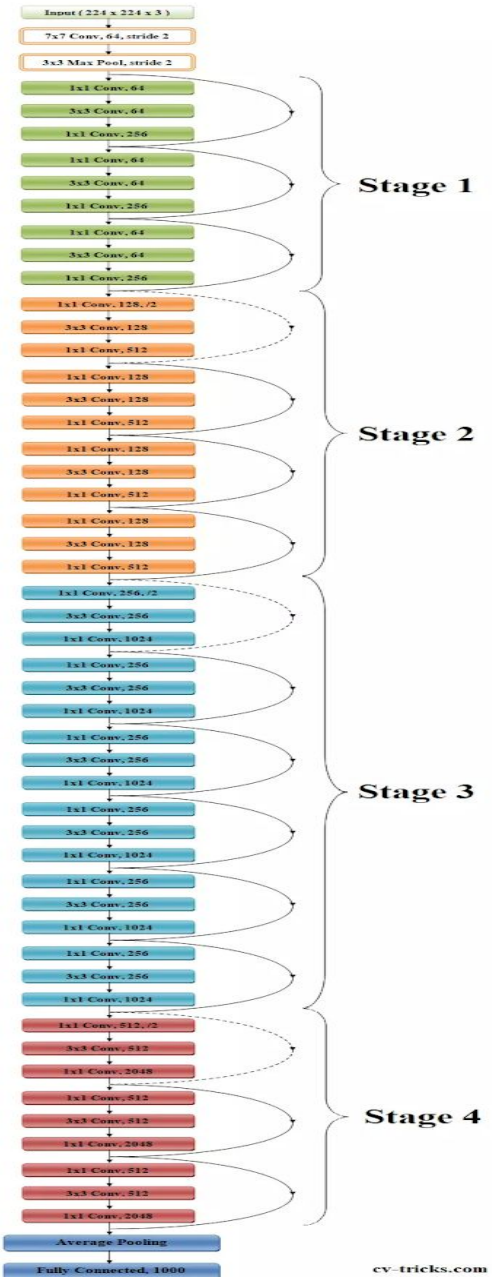
## 4.2 Using Resnet50

Resnet is one the recent work in the field of deep learning which makes it possible to train deeper networks without any problems. “As we know from universal approximation theorem that we can learn any function using a single feedforward neural network but that layer would be massive and might overfit to data. So We prefer the deeper architecture"[6] . But as we know training the deeper network is much more difficult due to vanishing gradient problem. Resnet solves this problem by adding “identity shortcuts” which copies the output of the current layer to the input of the 3rd next layer as we can see in the image.




(Image Source :[6])

The blocks shown in above images are called residual blocks. Before ResNet When people tried to train deeper networks they got training error which is higher than shallow networks. “The authors of ResNet argued that deep networks should not have higher error than shallow networks because you can just add identity mapping to shallow networks and performance would remain the same. Authors hypothesized that letting stacked layers map residual mapping is easier than letting them fit the entire mapping . And Residual blocks allows network to do that”.[6]




(Image Source : [9])

Using variants of residual blocks as a basis there have been numerous ResNet architectures developed in recent times. From those architectures we have decided to use ResNet 50 with consideration of training time and computation resources available to us.

### 4.3 Using Resnet50 +RNN

As we said earlier in order to get the best performance for the artist detection task from Mel-Spectrogram we have to detect frequency patterns as well as temporal patterns in the image. ResNet50 detects the frequency patterns but lacks the ability to detect temporal patterns. So we have decided to use the first 3 stages of RestNet50 structure in the image (we need at least some time related data in the image, using all the stages will reduce the features in the time dimension ) and 2 stacked layers of GRU after that. This model was deep enough to learn the frequency patterns in the image compared to the model based on the inception block . It also detects the temporal patterns so it performs better than just the ResNet50 model. For all the models we tried to optimize the performance using the regularization techniques like dropout and batch normalizations.

Since this is a multi-class classification problem, we will be using cross-entropy as the loss function for our experiment. A standard learning rate of 0.0001 is set for all the models with the number of epochs set to 300. Adam optimizer is chosen as the model optimizer due to the fact that its has shown the best performance in CNN tasks.

---

## 5. Result

In this section, we will tabulate and discuss the result of our models runs. The models run in this project are trained for varying sample length {1s,3s,6s} using two types of data split techniques which have been mentioned above. The resulting scores are an average of 3 runs and the scoring metric used is F1 score. This is because the sample size for each artist is not the same, hence there will be class imbalance issues. The F1 score in each run is in itself the average of the F1 scores for the 20 classes used in this project. Our Methodology involved running all the models for 1s samples and tabulating the result. Due to the lack of processing time available to us, we decided to run the rest of the samples using the best performing model.

Here are the results for the 1s sample:

| **Model** | **Split** | **1s** |
|---|---|---|
| Inception + RNN | Song | 0.5916 |
| | Album | 0.3689 |
| Resnet 50 | Song | 0.592 |
| | Album | 0.354 |
| Resnet 50 + RNN | Song | 0.6642 |
| | Album | 0.451 |

As can be observed, Resnet 50 with RNN gives the best average F1 score at the song level split with a score of 66.42%. Inception block and Rest50 models give a comparatively similar performance. Even when it comes to album level split, Resnet 50 with RNN gives the best scores out of the 3 models.

Now that we know which model performs the best, we can evaluate the best model's performance on the other sliced samples

**Average Test F1 scores for Resnet50 + RNN Model**

| **Split** | **1s** | **3s** | **6s** |
|---|---|---|---|
| Song | 0.6642 | 0.6543 | 0.6792 |
| Album | 0.451 | 0.5169 | 0.4873 |

As mentioned earlier, song level split have given better performance results than album level split. As mentioned earlier, this could be because of the influence of production details along with the musical style.

We can observe that we are getting the best overall results at the 6s mark. This could mean that having longer samples could mean better results. However the performance might diminish beyond because the noise rejection from voting using a large number of test samples outweighs additional temporal benefits.

Computation Effort : Below is the specification of the server we used to train our model.

Number of Cores: 4 cores

RAM size: 16 GB RAM

GPU Spec: 13 GB GPU RAM with 2 cores

| **Model** | **Average Time required to train** |
|---|---|
| Inception + RNN | 7 hours |
| Resnet 50 | 9 hours |
| Resnet 50 + RNN | 11 hours |

## 6. Conclusion & Future Scope

The aim of this paper was to develop and experiment with different CRNN architectures and decide which one would be the best for music artist classification on the artist20 dataset. Using the data and our 3 models, we concluded that the Resnet50 + RNN model is the best model for the classification problem. We also realized that the best results are derived using the song split technique of data splitting. We even concluded that the best sample length for all the samples we experimented with was the 6s length. Putting all these constraints together, our mean F1 score for the best model came out to be 67.92%, which is better than the models discussed in the above sections.

Future approaches can include experimentation with extensive sample lengths, looking at song level evaluation, audio augmentation and further model improvements.

## 7.References


[1]Zain and Yue Zhao “Music Artist Classification With Convolutional Recurrent Neural Networks”

arxiv: 1901.04555 , 2019

[2] K. Choi, G. Fazekas, and M. Sandler, “Explaining deep convolutional neural networks on music classification,” arXiv preprint arXiv:1607.02444, 2016.

[3] A. Wang et al., “An industrial strength audio search algorithm.” in International Society for Music Information Retrieval Conference (ISMIR), vol. 2003. Washington, DC, 2003, pp. 7–13.

[4] J. Panksepp and C. Trevarthen, “The neuroscience of emotion in music.” Communicative Musicality:

[5] A simple guide to Inception Net by bharath Raj , Towards Data Science Blog

[6] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in Proceedings of the 14th python in science conference, 2015, pp. 18–25

[7] An overview of Resnet and it’s variants by Vincent Fung,Towards Data Science Blog.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385,2015.

[9] Detailed Guide to Understand and Implement ResNets ,https://cv-tricks.com/

[10] B. Whitman, G. Flake, and S. Lawrence, “Artist detection in music with minnowmatch,” in IEEE Signal Processing Society Workshop. IEEE, 2001, pp. 559–568.

[11]D. P. Ellis, “Classifying music audio with timbral and chroma features.” in International Society for Music Information Retrieval Conference (ISMIR), vol. 7, 2007, pp. 339–340

[12] M. I. Mandel and D. P. Ellis, “Song-level features and support vector machines for music classification,” International