Music Genre Classification

Project-II (MA67102) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science And Data Processing

by

Rohan Kumar

(19MA60R23)

Under the supervision of

Dr. Pawan Kumar

Department of Mathematics

Indian Institute of Technology Kharagpur

Spring Semester, 2020-21

April 15, 2021

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 15, 2021 (Rohan Kumar)
Place: Kharagpur (19MA60R23)

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Music Genre Classification** " submitted by **Rohan Kumar** (Roll No. 19MA60R23) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Master of Technology in Computer Science And Data Processing is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2020-21.

Date: April 15, 2021
Place: Kharagpur

Dr. Pawan Kumar
Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Today data are increasing rapidly, So the number of songs releasing each day is relatively increasing. Each song is labelled with some genre, Like **Kuch Kuch hota hai** is labelled as romantic genre. Till late 20's century classification of genre of each song was done manually by human. In the real-time, like Wynk music, Spotify which have over more than 5 crore songs and each song has been categorized into different genre. In this Project, the objective is to classify the given songs into different genres from **GTZAN** dataset which contain 10 genres. We, first, implement K-NN classifier which has accuracy of about 70%. Short-term Fourier Transform (STFT), Mel-spectogram and Mel-frequency cepstrum coefficient (MFCC) are the most common feature extraction technique and each feature extraction technique has been successful in their own various audio applications. Then, these feature extractions of the audio fed to the Convolution Neural Network (CNN) model and VGG16 Neural Network model, which consist of 16 convolution neural network. We perform different feature extraction with different CNN and VGG16 model with or without Recurrent Neural Network and evaluated performance measure. In this model, we have achieved overall accuracy 90% for this task.

Keywords: GTZAN, Short-term Fourier Transform (STFT), Mel-spectrogram, Mel-frequency cepstrum coefficient (MFCC), Convolution Neural Network, VGG16, Recurrent Neural Network (RNN)

Acknowledgements

Every project is successful largely due to the the efforts of a number of wonderful people who had always given their valuable advice or lent helping hand. I sincerely appreciate the inspiration, support and guidance of all those people who have been significant in making this project a success.

I wish to express my sincere gratitude to my project guide, Dr. Pawan Kumar, Department of Mathematics, IIT Kharagpur, for his valuable guidance and motivation without which, it would not have been possible to bring out this project in the present form. Working with Dr. Pawan Kumar has been a pleasure experience. I am also grateful to him for his elaborate and detailed discussion on the subject and for providing sufficient help.

I wish to express my sincere thanks to Prof. Somnath Bhattacharyya, Head of the Department of Mathematics for the facilities and infrastructures and all the faculty members for their moral cooperation. I would also like to thank Prof. Pratima Panigrahi and Dr. Bodhayan Roy, our faculty advisor, for his immense support and helpful advice.

My heartfelt thanks to my family members and my friends for their love and support.

Rohan Kumar
M.Tech.(CSDP)
Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Music is categorized into subjective categories called genres. Humans have been the primary tool in attributing genre-tags to songs. Using a machine to automate this classification process is a more complex task. Machine learning excels at deciphering patterns from complex data. Genre classification is an important task with many real world applications. As the quantity of music being released on a daily basis continues to sky-rocket, especially on internet platforms such as Soundcloud and Spotify a 2019 number suggests that tens of thousands of songs were released every month on Spotify the need for accurate meta-data required for database management and search/storage purposes climbs in proportion. Being able to instantly classify songs in any given playlist or library by genre is an important functionality for any music streaming/purchasing service.

Despite being an interesting endeavour in and of itself, there are practical, realworld applications of automated music genre recognition, with music streaming services representing the most pertinent example. Pandora is a US-based streaming and

recommendation platform that employs musicologists to annotate songs with genre and rhythm features, with the aim of creating better recommendation playlists.

1.2 Motivation

With the advanced artificial intelligence, deep learning has shown better performance in music or voice classification. Automatic Speech Recognition (ASR), for example, OK Google in Android phone or Siri in IOS phone can recognize and understand what you are speaking.

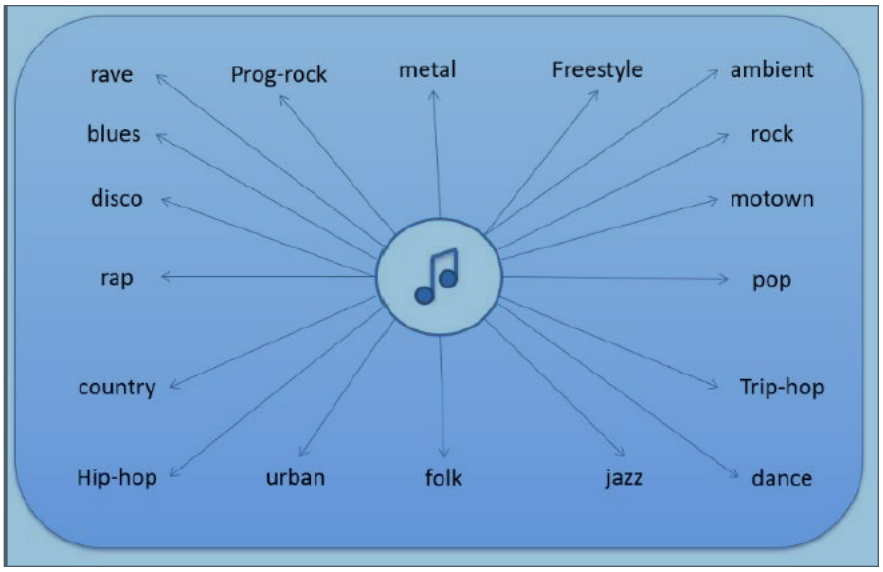

FIGURE 1.1: Various Music Genres

Researchers had applied different proposal algorithm and techniques to solve this kind of problem which can put it in deployment production. Spectrogram of audio data has proved and has considered to be the one very effective feature extraction techniques than others. For the classifier algorithm, several algorithms are common to work in this field like Gaussian Mixture Model (GMM), Support Vector Machine

(SVM), etc. Convolution Neural Network (CNN) had shown the prominent result with this spectrogram features as it feature taken into consider as an image.

1.3 Related Work

Music genre classification has been a widely studied area of research since the early days of the Internet. Tzanetakis and Cook (2002) addressed this problem with supervised machine learning approaches such as Gaussian Mixture model and k-nearest neighbour classifiers. They introduced 3 sets of features for this task categorized as timbral structure, rhythmic content and pitch content. They achieved 61% accuracy. As a benchmark, human accuracy averages around 70% for this kind of genre classification work . Tzanetakis and Cook used MFCCs, a close cousin of mel-spectrograms, and essentially all work has followed in their footsteps in transforming their data in this manner.

In recent years, however convolutional neural networks have shown to be incredibly accurate music genre classifiers, with excellent results reflecting both the complexity provided by having multiple layers and the ability of convolutional layers to effectively identify patterns within images (which is essentially what mel-spectrograms and MFCCs are). These results have far exceeded human capacity for genre classification, with our research finding that current state-of-the-art models perform with an accuracy of around 91% when using the full 30s track length. Many of the papers which implemented CNNs compared their models to other ML techniques, including k-NN, mixture of Gaussians, and SVMs, and CNNs performed favorably in all cases.

Nilesh M. Patil, they had done with the idea of applying Machine Learning Techniques. SVM and k-NN are the most frequent applied machine learning techniques

in many fields like image processing, twitter sentimental analysis, etc. In this work, the dataset they used GTZAN. To obtain the feature extraction from the audio file, it's done by Mel-Frequency Cepstral Coefficients (MFCC), then feature vectors are obtained. Now, these feature vectors are classified with supervised learning approaches i.e. k-NN, Linear SVM and Poly SVM. Each classifier have achieved overall accuracy with 64.4%, 60% and 77.78%.

Caifeng Liu , they proposed to develope a convolution neural networks architecture which takes the long related to context of information into considerations and transfers further suitable information to decision-making layer. To develop the new neural network, they used the idea of Inception model by combining convolutions with different kernel size and layers. So, the dataset they used were GTZAN, Ballroom and Extended Ballroom. The pre-processing steps done to extract feature called mel-spectogram with 128 Mel filters from audio signal. Then this feature input of size 647x128 fed to BBNN. They achieved overall accuracy 93.9%, 96.7 %, 97.2%.

1.4 Problem statement

Now Our Problem Statement is how to make a audio signal into a meaningful information. And from that information how we will able to find music genre of that particular clip or audio signal. To classify these things we will use Machine Learning classifiers and Deep neural Networks to improve its accuracy to classify it's genre. To design a model on given music data which can classify to specific music genre.
So, our first information is to classify songs genre. The genres are - blues, classical, country, disco, pop, jazz, reggae,rock, metal. We will work with GTZAN data set which is availabe on `http://marsyas.info/downloads/datasets.html`

Chapter 2

Background

2.1 Audio Features

Panagakis et al. (2008, Introduction) state that there are three main types of audio feature employed for music classification: timbral texture features, rhythmic features, and pitch content features. Detailing all possible feature is outside the scope of this project. Following are the most used features for classifiaction.

- MFCCs (timbral texture feature)
- Zero Crossing Rate (timbral texture feature)
- Spectral centroid (timbral texture feature)
- Chroma Features (pitch content feature)

2.2 Spectral Analysis Techniques

Spectral Analysis is to analyse the spectrum to find the pattern or properties from the source. A spectrum means to differentiate between amplitude and frequency

domain. When we say spectral analysis or techniques, which involves with Fourier Transform or any other technique which transform into frequency domain because after applying it to the source, we get the spectrum of that source.

2.2.1 Short Term Fourier Transform

As we all know, when we want to plot the waveform of any audio, it differentiates with amplitude and time. But not only time importance but also frequency. Time tells us when we hear something, and frequency tell us what we heard. The waveform of the audio is expressing as the sum of sinusoidal function, i.e., the waveform can be breaking down into sinusoidal function which is also frequency.

Suppose, we have a waveform of time domain function. This waveform shows the summation of sinusoidal function with 50 and 80 Hz frequency So, in order to obtain

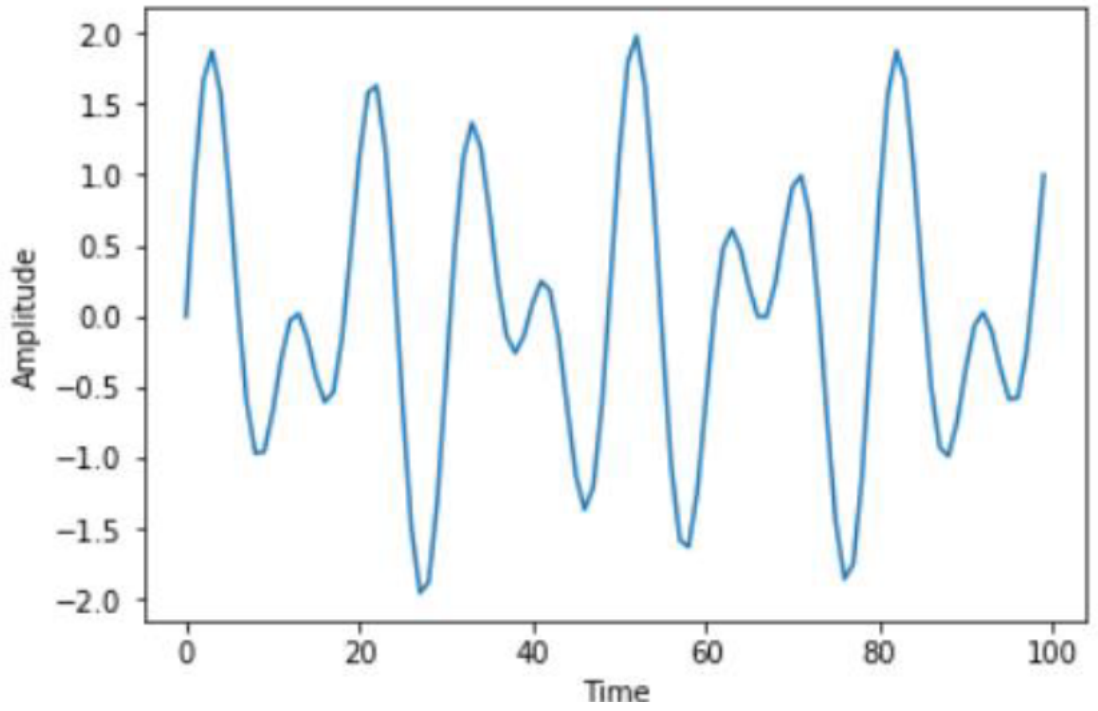

FIGURE 2.1: Waveform of Time Domain Function

frequency domain from time domain, this transformation is called Fourier Transform.

Because analysing at frequency domain is much easier and simpler than time domain. There is another term called Discrete Fourier Transform (DFT) because we are dealing with discrete values not continuous values. By definition of DFT, And to

$$X(k) = \sum_{n=0}^{N-1} x(n) * e^{-j2\pi nk/N}$$

convert time domain from frequency domain, its called inverse of DFT. By definition of IDFT,

$$x(n) = \sum_{k=1}^{N-1} X(k).\, e^{j2\pi nk/N}$$

So, here x(n) is discrete time function and X(k) is Fourier transform of x(n).
But looking them make it complex to evaluate with summation. Let assume that

$$D_N = W_N^{nk} = e^{-i2\pi nk/N}$$

This is called twiddle factor. So, in order to compute faster, we create the generalize form after putting value k=0,1,2,3 to form twiddle matrix.

$$X(k) = D_N \times x(n)$$

Where X(k) a Fourier Transform of x(n) in column vector (N×1) and x(n) is discrete time signal in column vector (Nx1) and D_N Twiddle matrix (kxn) which is represented by

$$n = 0,1,2,3 \ldots (N-1)$$

$$W_N^{nk} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

$$k = 0,1,2,3 \ldots (N-1)$$

Twiddle Factor Matrix

After finding the Fourier transform of x(n),we get the result in complex number. To do that, we take the absolute value of each X(k). If we apply Fourier transform for signal Fig 2.1., we get

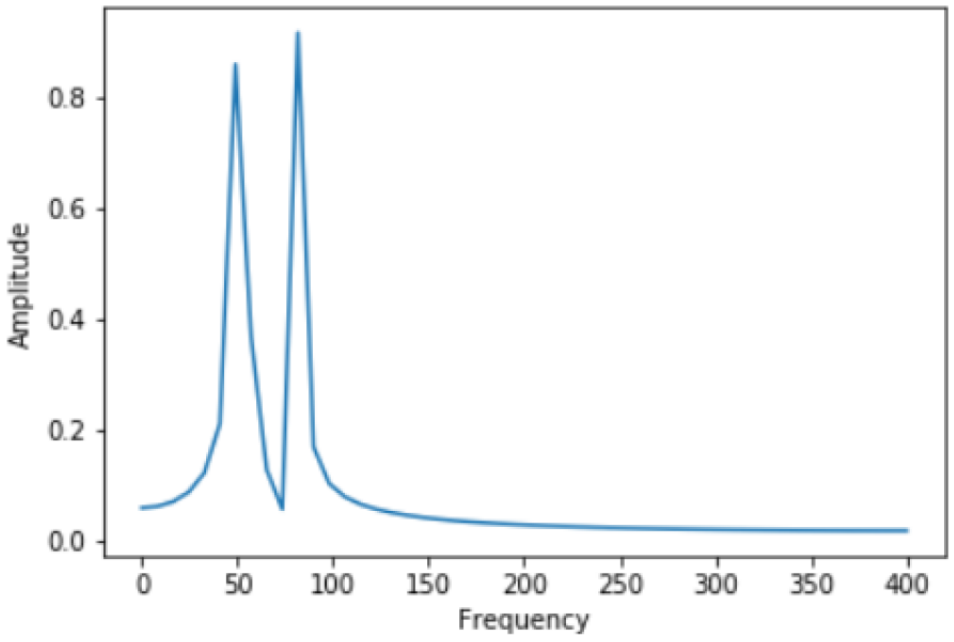

FIGURE 2.2: Fourier Transform of signal

In the Fig 2.2, we observe that two frequencies are having high amplitude and peak

value which telling us that these two frequencies are present in this signal with respect to other frequencies (showing no changes). At the peak value, we found that frequencies of 50 and 80 Hz are there this signal.

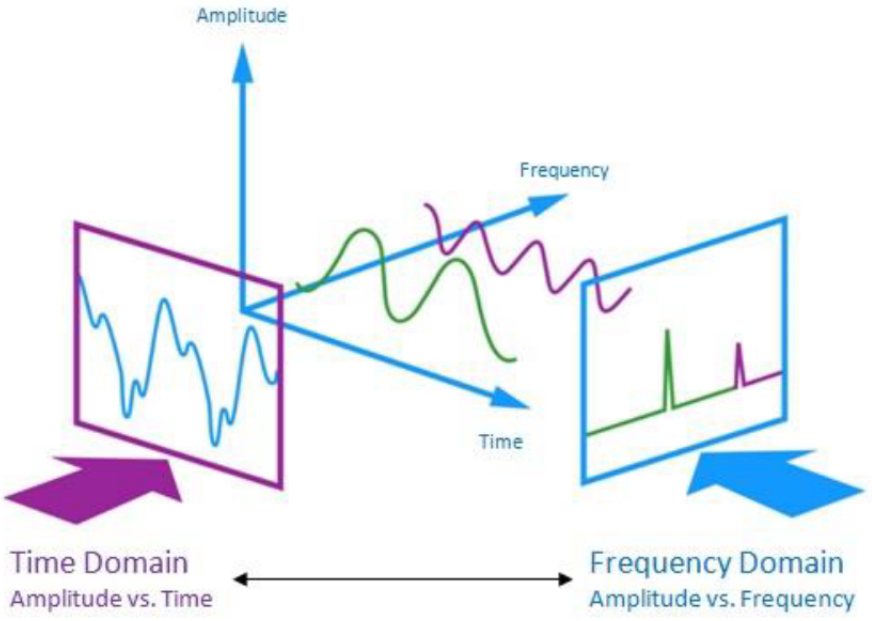

FIGURE 2.3: Visualization of Time Domain and Frequency Domain

But there is a limitation of applying DFT that for different signal with time showing same frequency. For example, let take a chirp signal and observe the figures below

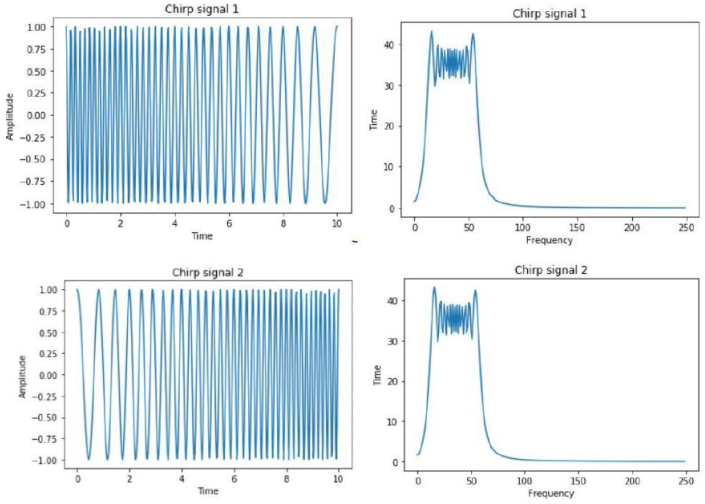

FIGURE 2.4: Illustrating Different Time Signal showing same Frequency

In Fig 2.4, Even though the signals are different but shows same frequency. This means that Fourier Transform only give us what are the frequency present in this signal, but it will not tell us when this component of frequency occurs.

Here we use short term fourier transform [10, 21]. As the name suggested "short term" means signal are split into small fixed duration of signal and then apply fourier transform of each block. Moving the sliding to create every block in signal called framing. This is computed frequency change with signal over time. Before computing, each block is multiplied with windowing function in order to enhance the ability of Fourier Transform to extract spectral data from signals. We used Hann window which look like bell curved shape (see function g(t) below Fig 2.5). Now

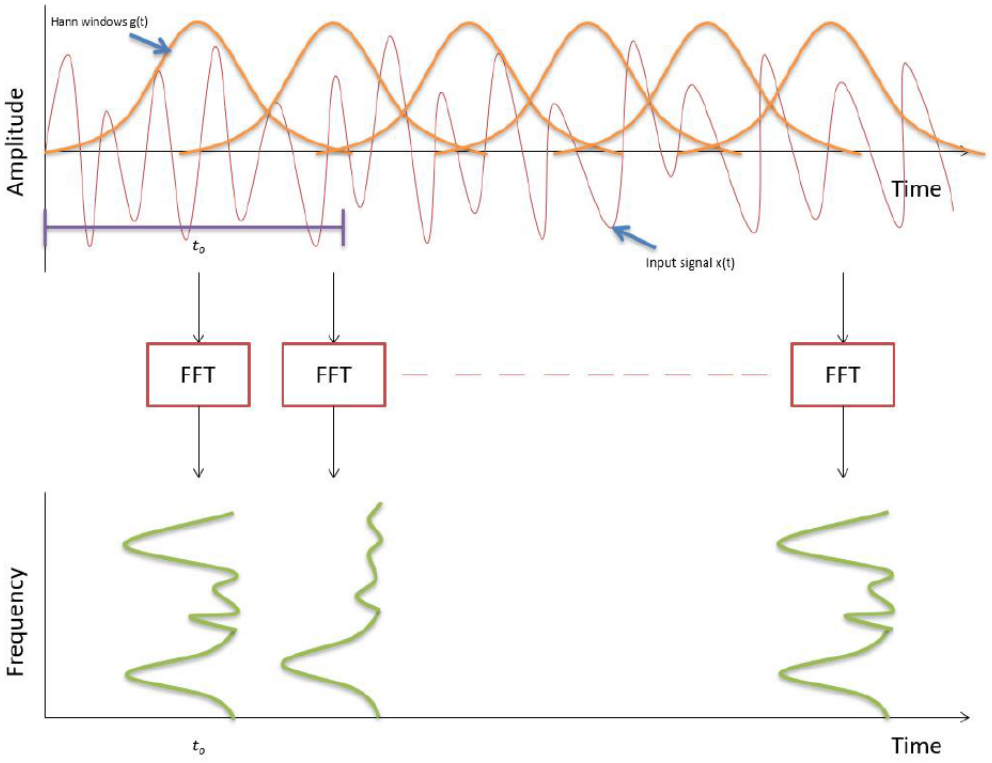

FIGURE 2.5: Visualization on STFT

after applying FFT on each block, we got frequency representation. Now this all frequency representation with time for this signal will be consider as their features which is also called as spectrogram.

2.2.2 Mel-Spectrogram

What differentiates Mel-spectrograms from regular spectrograms is their frequency spacing on y-axis. Mel-frequency spacing better approximates the hearing scale for human ears where lower frequencies are emphasized and higher frequencies are compressed. This approach seemed especially appropriate because our heat map results from our fast-Fourier transforms showed that most active sounds occured in lower frequencies. We used Librosa to produce a mel-spectrogram for each track. The processed tracks were split into smaller clips as individual samples. We varied the length of such samples to find the optimal partitioning, which is discussed in the results section.

2.2.3 Mel-Frequency Cepstral Coefficients

MFCCs are one of the most common feature types used in audio classification of all kinds, having been used in speech and ambient noise recognition as well as music genre classification. Unlike the estimated tempo of a given audio signal, which consists of a single value, ordinarily between 13 and 30 MFCCs are extracted, with the exact number of coefficients a decision made by the researcher. In this project, of the 54 features extracted from the audio file dataset, 30 were MFCCs; thus, it is worth examining the feature type in some detail.
MFCCs can be better understood by understanding the process by which they are derived from an audio file, which ordinarily runs as follows (Lyons, 2012):

1. An audio file is divided into small frames (a process known as 'windowing'), each lasting 20-40ms. The resulting vector is a time series that represents the overall audio file.

2. For each frame, an estimation of the power spectrum is calculated, by use of a Fourier Transform. The power spectrum of an audio frame is roughly

equivalent to the power present for each of the frequency bands within that frame.

3. A 'Mel-spaced Filterbank' is used on each of the output sets from step 2. The Filterbank is spaced according to the mel scale, to give emphasis to the frequency bands that are most relevant in human perception (humans find it easier to perceive differences in the lower registers, i.e., the bassier frequencies, so the filters are more narrowly spaced in these lower registers than in the higher registers). Each of the filters corresponds to a particular frequency band, and removes all of the values that fall outside this range. The result is a series of'filterbank energies'; one value per filter.

4. After computing the filterbank energies, their logarithms are calculated.

5. A Discrete Cosine Transform is performed on the logarithms, and approximately half of the resulting values are dropped. The result is a time-series vector of MFCCs: one set of MFCCs for each frame of the audio file.

It is further possible to find the mean values for each MFCC of an audio file. For example, to find a signal's mean MFCC5 value, the values of MFCC5 across all windows of the signal would be added up, and this value would then be divided by the number of windows in the signal. This process can be applied for each MFCC to find the mean MFCC values.

MFCCs are used in music genre classification to detect timbre (Jensen et al., 2006),which can be defined as the 'quality' or 'colour' of a sound. Different genres of music use different sets of instruments, resulting in striking timbral differences - compare the soft sound of Chopin's piano compositions with the harshness of punkrock. Even when two genres use the same set of instruments, they are often used to totally sonic different effect; e.g. Electronic Dance Music often features a bass-heavy and punchy kick drum, in comparison to jazz music, which typically uses the kick drum in a

much more subdued way. The timbral qualities of a song are detectable in its spectral information, and use of MFCCs has proven to be a powerful way of representing this content for use in machine learning and Deep Learning.

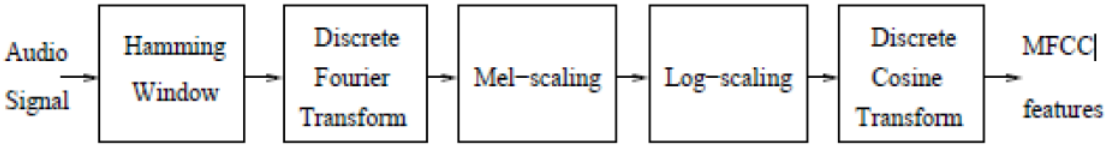

FIGURE 2.6: Block diagram for calculation of MFCC

2.3 Other Features

Chroma Frequency vector discretizes the spectrum into chromatic keys and represents the presence of each key. We take the histogram of present notes on a 12-note scale as a 12 length feature vector. The chroma frequency has a music theory interpretation. The histogram over the 12-note scale actually is sufficient to describe the chord played in that window. It provides a robust way to describe a similarity measure between music pieces.

Spectral Centroid It describes where the "centre of mass" for sound is. It essentially is the weighted mean of the frequencies present in the sound. Con sider two songs, one from blues and one from metal. A blues song is generally consistent throughout it length while a metal song usually has more frequencies accumulated towards the end part. So spectral centroid for blues song will lie somewhere near the middle of its spectrum while that for a metal song would usually be towards its end.

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n) x(n)}{\sum_{n=0}^{N-1} x(n)}$$

Zero Crossing Rate It represents the number of times the waveform crosses 0. It usually has higher values for highly percussive sounds like those in metal and rock.

$$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{I}\{s_t s_{t-1} < 0\}$$

Spectral Roll-off It is a measure of the shape of the signal. It represents the frequency at which high frequencies decline to 0. To obtain it, we have to calculate the fraction of bins in the power spectrum where 85% of its power is at lower frequencies.

Spectral Crest Factor (SCF) Crest factor indicates how extreme the peaks are in a waveform. Crest factor 1 indicates no peaks, such as direct current. Higher crest factors indicate peaks, for example sound waves tend to have high crest factors. Another measure of noisiness, closely related to SFM.

Spectral Flatness Measure (SFM) Measures the noisiness of an audio signal. High values indicate high noise levels where all frequencies have similar amount of power, while low values indicate "cleaner" sound.

Chapter 3

k-NN Classifier

3.1 Classifier

k-NN classifier is a supervised learning algorithm where each feature vector from a reference collection is treated as a point in multi-dimensional feature space and k-NN retrieval is subsequently used to classify the unclassified feature vectors. Classifying songs can be cast into k-NN query processing for large collections of high dimensional vectors. Such k-NN classification has two major advantages. First, approximate k-NN algorithms can cope very successfully with scale. Second, k-NN classification has the added advantage of handling fuzzy class assignments very well, as different near neighbors can give weights to different classes.

3.2 Feature extraction

It is the process of generating a set of numerical descriptors that characterize the audio. In order to extract features, Short Time Fourier Transform (STFT) is used to break the audio signal into short analysis windows or time frames, which can possibly overlap. Both the size of the analysis window, w, and the hop size, h, can

be changed from the default of 512 samples. The actual features are then calculated as running means and variances over several such analysis windows. This forms another window, called texture window, which can be thought of as a memory of the last m descriptors (by default, m = 40).

3.2.1 Dataset

The dataset that we used in this project is GTZAN dataset. This dataset was created by Tzanetakis and Cook. It contains 1000 audio excerpts of 30 sec distributed in 10 musical genres (Blues, Reggae, Classical, Country, Disco, Hip-Hop, Jazz, Metal, Pop, Rock).

Marsyas (Music Analysis Retrieval and Synthesis for Audio Signals) is a free software framework for audio analysis, synthesis and retrieval. This software has been written by George Tzanetakis and used for a variety of both academic and industrial projects. The major underlying theme under design of Marsyas software has been to provide an efficient and flexible framework for Music Information Retrieval. It is also said to be regularly maintained by its author and there are plans to extend its functionality in the future. The Marsyas implementations include the standard temporal and spectral lowlevel features like spectral centroid, spectral roll off, spectral flux, zero crossing rate, and mel frequency cepstral coefficients (MFCC). The framework includes a variety of building blocks for performing common audio tasks. Some representative examples are: sound file IO, audio IO, feature extraction, signal processing and machine learning.

3.2.2 Preprocessing data

Before giving audio data as an input to the system preprocessing of the signal is required. Preprocessing means filtering, cutting down the silent zone before the

signals are normalized. Preprocessing may include simple DSP operations such as sampling rate conversion. It is important to choose the correct sampling rate (a practical upper bound being, of course, the original recording rate). Too high a sampling rate may place unnecessarily high requirements on the feature extraction module, as a large portion of the information it receives is irrelevant for the goals of the automatic classification. Too low a sampling rate will make the antialiasing cutoff frequency so low as to discard useful information from the higher frequencies. All data which is being fed to the system is processed in the same manner for which the complete silent zone prefixing the sentence and post fixing the sentence is chopped out. The preprocessed data will be used both in training and testing phase. The preprocessing steps are as follows.

- Sample the audio signal at the rate of 28 KHz. At this sampling rate, signals over 14 KHz are ignored. As humans cannot hear signals over 20 KHz, those frequencies need not considered. And at sampling rate as low as 8 KHz, as in telephone facility, different genres of music can still be differentiated. So, we chose some sampling rate that lies between these two limits.

- Apply Fourier Transform to 256 samples, after multiplying them with Laplace smoothing function. The number of samples (256) was chosen such that it was large enough to capture the essential features of the music.

- Get the amplitude spectrum from the result of the Fourier Transform at the first 128 frequencies. Thus, we have a vector of length 128.

- Then we slide the window of samples, take the next 256 samples, repeat the same process and so on. We do this till the end of the audio clip.

- Now we make groups of 50 consecutive vectors and take average of each group. Such vectors (each of length 128) are used as inputs to our learning algorithms.

3.3 The Classification Process

In a supervised genre classification process there are two sets of songs. The first set, called the training set, has a solid set of genre labels, which are used to guide the classification process. The features extracted from the training set are typically preprocessed in some manner. For SVM classification, e.g., the SVM is applied to the set to learn the classifications, while for k-NN classification, the training set may be indexed to facilitate the subsequent classification.

The second set is the set to be classified, or the classification set. For each song in this set, the classification process is applied to all features of the song, resulting in a genre assignment. If each song is represented by a single feature, the genre of the feature is automatically applied to the song. If each song is represented by multiple features, however, the genres of the features are aggregated into a genre assignment for the song, e.g., through a voting process, where the genre assigned to the most features is assigned to the song.

In k-NN classification, each query feature is assigned the genre of its k nearest neighbors. Distance between points can be calculated using any standard distance calculation such as Euclidean Distance, Manhattan Distance or Hamming Distance. When k = 1, the classification is based on the one nearest neighbor, but when k ¿ 1 the votes of all k nearest neighbors have equal weight. The number of high-dimensional feature vectors can be hundreds or thousands per song, resulting in a large number of votes per song.

To give an example, assume that each song is represented by 1,200 descriptors, and that each feature retrieves k = 3 nearest neighbors. Then each song will retrieve 3,600 similar features from songs in the training set, where each feature votes for the genre of the song it came from. These 3,600 votes are then aggregated and the genre with the most votes is assigned to the song to be classified. There can be two ways of applying k-NNclassification algorithm.

3.3.1 Sequential Scan

The sequential scan is the most straight-forward method for k-NN classification. As the name suggests, this classifier sequentially scans the collection of features from the training set and computes the distance to each feature of the song to be classified. The advantage of a sequential scan is that no training phase is required; the disadvantage is that scanning the training set features and computing distances takes time. Nevertheless, no k-NN classifier exists that is more efficient, while guaranteeing the same "accurate" results as a sequential scan would give.

3.3.2 Cluster Purning

Cluster Pruning algorithm starts by randomly selecting C cluster leaders from the training set feature collection. The number of clusters is typically chosen such that the average cluster fits with one disk read. The features in the training set collection are then read, one by one, and assigned to the closest cluster leader. For large feature collections, where there are many cluster leaders, they are organized into a hierarchy for more efficient processing. During k-NN retrieval, the features from the song to be classified are considered one by one. For each feature, the b closest cluster leaders are identified, again through the hierarchy, and the clusters they represent are retrieved and scanned for the k approximate nearest neighbors. Experience from different application domains has shown that b = 1 often gives excellent results. If the training set feature collection is divided into C clusters, then the time required for k-NN retrieval is roughly b/C times the time required for a sequential scan.

3.4 Classification Result

3.4.1 Evaluation Metrices

Confusion Matrix: The columns of the confusion matrix represent the predictions, and the rows represent the actual class. Correct predictions always lie on the diagonal of the matrix. Equation shows the general structure of confusion matrix. Below equation shows the structure of confusion matrix.

$$\begin{bmatrix} TP & Fn \\ FP & TN \end{bmatrix}$$

wherein, True Positives (TP) indicate the number of instances of a class that were correctly predicted, True Negatives (TN) indicate the number of instances NOT of a particular class that were correctly predicted NOT to belong to that class. False Positives (FP) indicate the number of instances NOT belonging to a class were incorrectly predicted belonging to that class and False Negatives (FN) indicate the number of instances that were incorrectly predicted belonging to other class.
Recall is a metric that gives a percentage of how many of the actual class members the classifier correctly identified. (FN + TP) represent a total of all minority members. Recall is given by below equation.

$$Recall = TP/TP + FN$$

Precision It gives us the total the percentage of how many of a particular class instances as determined by the model or classifier actually belong to that particular class. (TP + FP) represents the total of positive predictions by the classifier. Precision is given by equation below.

$$Precision = TP/TP + FP$$

Thus, in general it is said that Recall is a Completeness Measure and Precision is an exactness Measure. The ideal classifier would give value as 1 for both Recall and Precision but if the classifier gives higher (closer to one) for one of the above metrics and lower for the other metrics in that case choosing the classifier is difficult task.

Below Table shows the "confusion" matrix, for k = 3. Classic is classified with 100% accuracy and Blues, Jazz, and Metal all have over 90% accuracy. These are all well-defined genres. We see that in most cases the mistakes the program does are similar to those humans commonly make, and in general these results agree with others reported in the literature. The table shows that 3 blues songs are classified as jazz, 7 country songs are classified as pop (which is a very fuzzy classification genre), 10 disco songs are classified as pop, 8 jazz songs are classified as classic, and 7 reggae songs are classified as pop. Just as in (Tzanetakis Cook, 2002) the lowest accuracy is for the rock genre. We were, however, surprised to see how many rock songs were classified as disco songs. Country also receives a low accuracy score, and it is interesting to note how few songs are also wrongly classified as either rock or country. This indicates that these genres are very broad in nature.

	Blues	Classic	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	93	0	0	3	0	3	0	0	0	1
Classic	0	100	0	0	0	0	0	0	0	0
Country	4	7	66	11	0	1	0	7	2	2
Disco	1	1	2	80	3	0	0	10	1	2
Hiphop	2	0	1	0	80	0	2	8	5	2
Jazz	0	8	0	0	0	91	0	1	1	0
Metal	2	0	0	1	1	0	92	0	0	4
Pop	1	3	3	2	1	0	1	87	1	1
Reggae	1	1	1	8	5	0	0	7	75	2
Rock	3	7	3	20	4	3	5	7	4	44
Total	107	127	76	125	94	98	100	126	88	58

3.5 The effect of feature selection

For this experiment, we start by extracting the default timbral features. We then experimented with adding features; the results are summarized in Below Table. Below Tableshows that adding Spectral Flatness Measure (SFM) increases the accuracy for this test set from 75.4% to 80.4%. Adding further information to the feature vector, however, did not improve the results, and in fact we observed that it actually hurts the results slightly. We conclude that in addition to the default timbral features it is beneficial to include Spectral Flatness Measure to achieve best results.

Features	Dimensions	Accuracy
Timbral features (TF)	34	75.4%
TF + SFM	82	80.4%
TF + SFM + SCF	130	80.0%
TF + SFM + LSP	118	80.0%
TF + SFM + LPCC	106	79.8%
TF + SFM + CHROMA	106	79.4%
TF + SFM + SCF + LSP	166	79.8%
TF + SFM + SCF + LPCC	154	79.4%
TF + SFM + SCF + CHROMA	154	79.9%

3.6 The effect of neighbors retrieved

We experiment with the value the k parameter, which indicates how many neighbors from the collection are considered for each query descriptor. We ran the k-NN search for values of k ranging from 1 to 10.

Neighbors k	Accuracy	Time
1	80.4%	207.4 min
2	80.7%	208.3 min
3	80.8%	209.3 min
4	80.6%	210.1 min
5	80.5%	212.3 min
10	80.5%	217.5 min

Chapter 4

Deep Learning Used

4.1 Convolution Neural Network (CNN)

Convolution Neural Network is a class of Neural Network which has been successfully in image classification, recognition and segmentation. In this architecture of

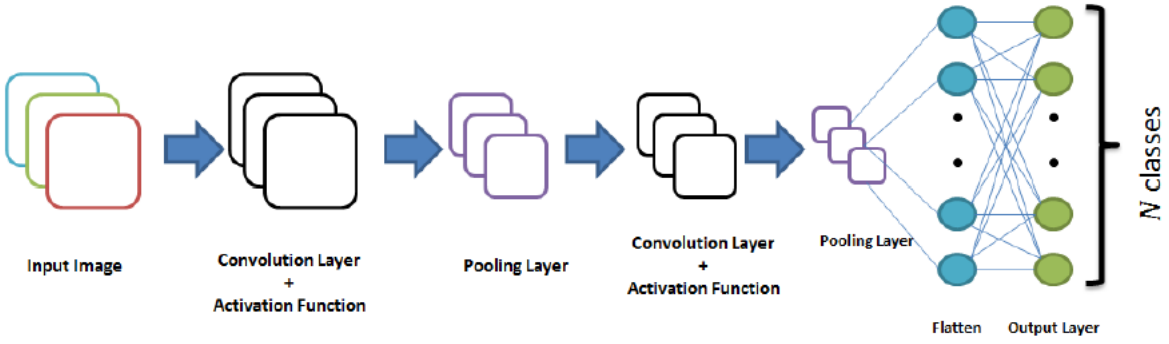

FIGURE 4.1: CNN Architecture

Convolution Neural Network, there are four operational stages:

1. **Convolutions** When the input image is fed to this network, its convolute with the convolution kernel. It is noticed from architecture (Convolution Layer) that there are many outputs after performing convolute to every convolution

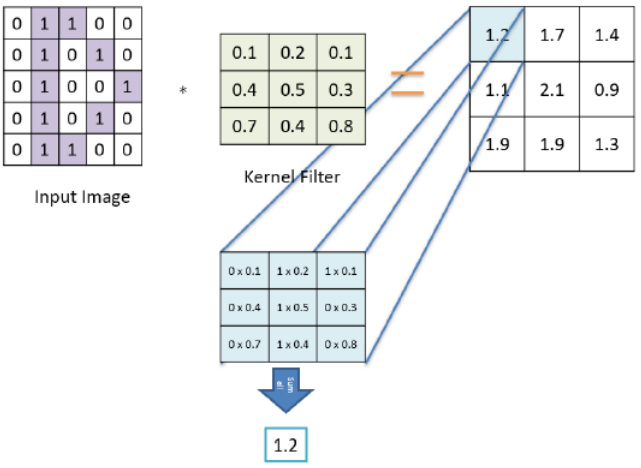

FIGURE 4.2: Illustration of computing Convolution of Input image (5 x 5) with convolution kernel (3 x 3)

kernel which are not same to other kernels so that for a input image, we can find several features to identify unique from other classes.

2. **Rectilinear Unit (ReLU)** These convolution outputs are then passed through non-linear activation such as ReLU, sigmoid and tanh. The most commonly activation function used as ReLU. Mathematically, it is defined as $R(Z) = max(0, Z)$ where z is pixel value. The reason is that, after applying convolution filter with input, it is possible that pixel may contain negative pixel value. Generally, image pixel value is range from $0 to (2^g rayscale - 1)$ applying these, all negative pixel value will be replaced with 0.

3. **Pooling** In this pooling layer, it basically down sample the image which reduce the dimensionality of the feature input by factor of 2. The most commonly in this layer is MaxPooling. In maxpooling operation, we take the maximum ones with the kernel size from feature map. Illustration of MaxPooling

4. **Fully Connected Layer** In this layer after performing several convolutions and/or maxpooling layer based on hyper tuning, the feature input are flattened into 1-D array. Now, from these 1-D array input to last layer i.e. output unit. They can now treat as back-propagation neural network. In the output unit,

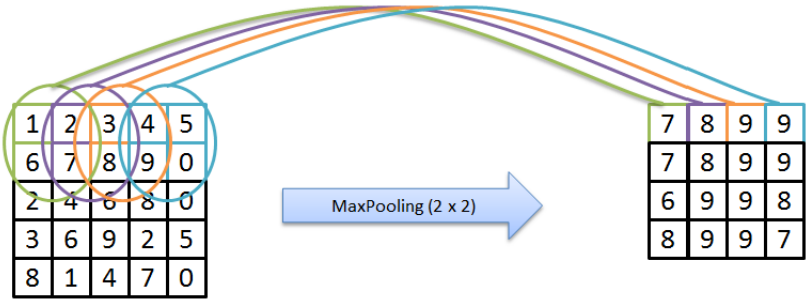

FIGURE 4.3: Illustration of MaxPooling

we apply activation function as sigmoid if it is binary classification. And for activation function as SoftMax if it is multiclass classification.

4.2 Recurrent Neural Networks

The main idea behind Recurrent Neural Networks is to use sequential information. In a simple neural network we keep all inputs and outputs independent of each other. But for most of the tasks it is not a idea. Now you need to now which words came before if you want to predict upcoming words.Now for each element of a sequence RNNs perform same task that is why they are called recurrent, with the output being depended on the previous computations. An another way to think about RNNs is that they have a memory which captures information about what has been calculated so far. In theory RNNs can make use of information in arbitrarily long sequences,but in practice they are limited to looking back only a few steps (more on this later). Here is what a typical RNN looks like:

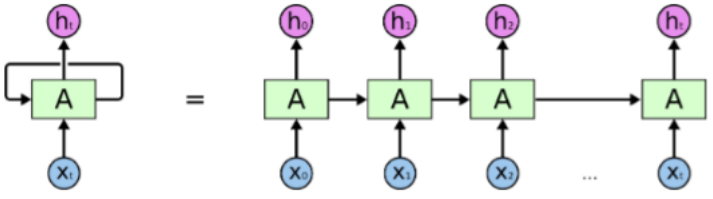

FIGURE 4.4: RNN Architecture

4.2.1 LSTM Networks

Now let us take a look at a special kind of Recurrent neural networks, that is LSTMs (Long Short Term Memories), which are capable of learning long-term dependencies. They were introduced by Hochreiter Schmidhuber (1997), They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but

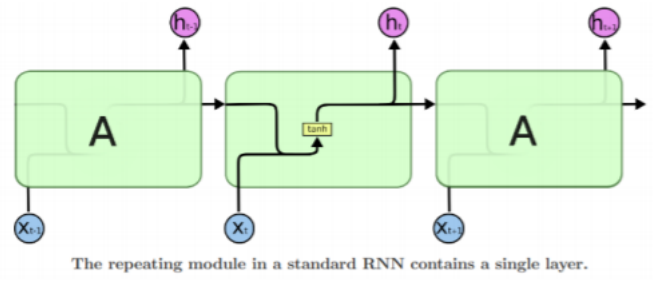

FIGURE 4.5: LSTM

the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way

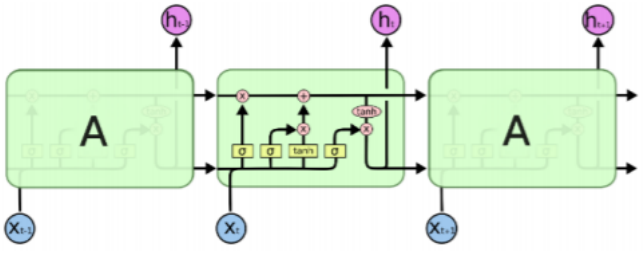

FIGURE 4.6: LSTM

4.3 VGG16

This model usually in the field of deep convolution neural network for object recognition which was developed and trained by Oxford's renowned Visual Geometry Group (VGG) and achieved better performance on ImageNet Database. It is totally appealing, however not only that it works well, because Oxford team has also stored the structure of pretrained network and weights which can further use for the other dataset that are available for free online to save time. We've used VGG16 i.e. Con-

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

FIGURE 4.7: Configuration of ConvNet

vNet Configuration of 16 weight layers (D) and added final layer as corresponding number of classes present in the dataset. So, instead of using pretrained weight network, we configure the network manually with ConvNet Configuration D with initializing random weights (not trained ones).

Chapter 5

Experimental Results

5.1 Proposed Architecture

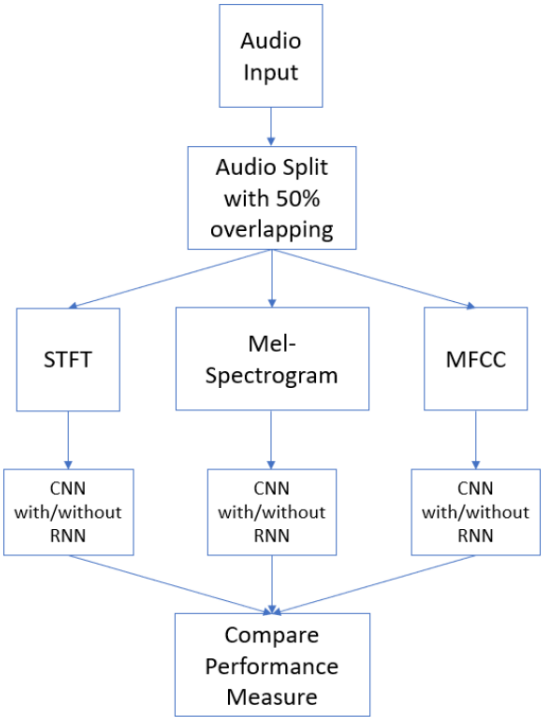

FIGURE 5.1: Proposed Architecture

5.2 Dataset

We used GTZAN dataset. It contains 10 classes and each class contains 100 audios. Each audio contains 30 seconds of track. The tracks are in 22050 Hz with mono 16-bit audio file in .wav format.

5.3 Data Pre processing

We read the audio with the sampling rate of 22050 Hz. After that, we split the audio of 30 seconds durations into 3 seconds durations of audio clips. The problem is that when split the audio, the computer will consider that each clip are not relate to each other and are independent. So, in order to avoid this, we have taken 50% of previous duration of data and next 50% of next duration of data. This will help to understand the computer that the first 50% duration of data are from the previous audio clip and from that data, it continues to next 50% duration of data.

5.4 Feature Extraction

Each clip has performed different feature extraction to see the analysis which one been performed better. First, we applied Short Term Fourier Transform (STFT) where FFT window size (frame size) is 1024 and hop length (frame increment) is 512 and windowing function is Hann function. Each clip gets (513, 129) dimensions that is 513 frequency bins and 129 frames in time. Second, we applied Mel-Spectrogram o each clip where FFT window size is 1024 and hop length is 512. Each clip gets (128,129) dimensions. Lastly, we applied further extended to Mel-Spectrogram to MFCC. I observed with different top k coefficient values after applying DCT, I got 13 to be the best optimal choice.

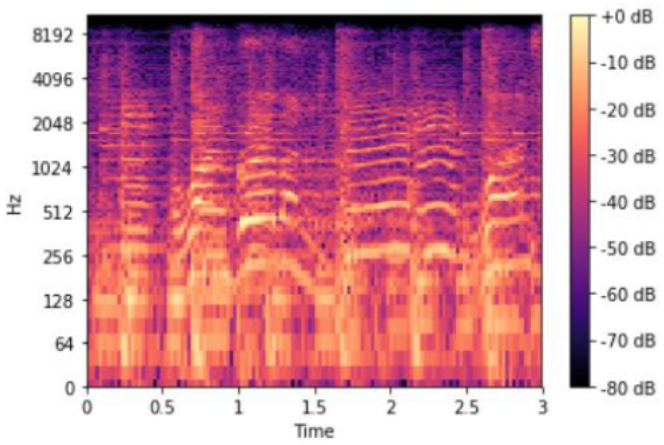

FIGURE 5.2: STFT for a blue class audio with 3 seconds duration

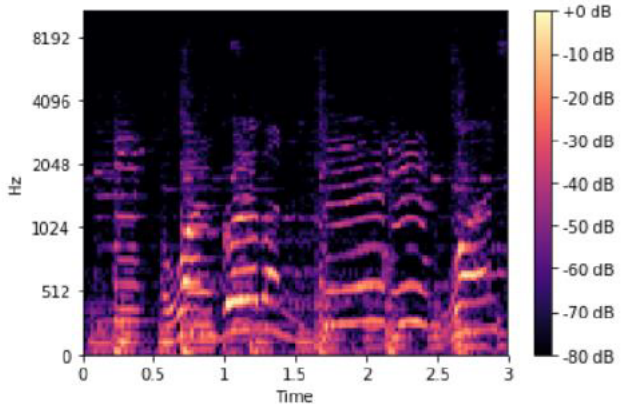

FIGURE 5.3: Mel-Spectrogram for a blue class audio with 3 seconds duration

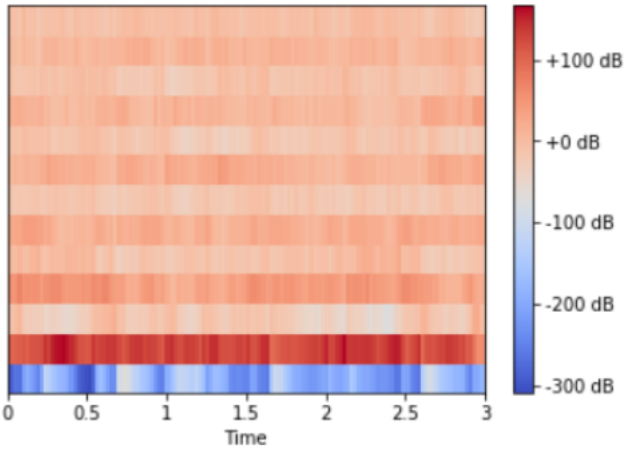

FIGURE 5.4: MFCC for a blue class audio with 3 seconds duration

5.5 Learning Algorithm

Data are divided into three sections: training data, validation data and test data. And these data are split into 80% of training data, 10% of validation data and 10% of testing data. These training data are then fed to Convolution Neural Network (CNN) before reshaping the data. Once an epoch is done, we check the performance of validation data to see how much can generalize the unknown data after learnt from training data. Once it completed all trained network, we evaluate for the test data after seeking training and validation data performance.

VGG16 has been trained on Mel-Spectrogram feature extraction data only because of the limited resource present in our system. This network has outperformed than CNN in these features. You will see the result in tabular form clearly in the next section part.

In configuration of proposed CNN network, we create five Convolution Block layer. Each block layer contain Convolution Kernel with different kernel size followed by MaxPooling Layer followed by Dropout with droprate 25%. Then we flatten them into 1D array and output layer.

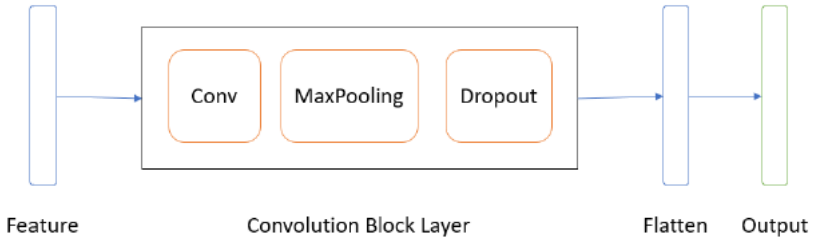

FIGURE 5.5: Proposed CNN Architecture

For the configuration of proposed CNN with LSTM (or GRU) network, we create five Convolution Block layer. After that we followed by three LSTM/GRU layer with two 128 lstm (or gru) unit and one 64 lstm (or gru) unit then we flatten them into 1D array and apply one dense layer and finally, that last one as, output layer.

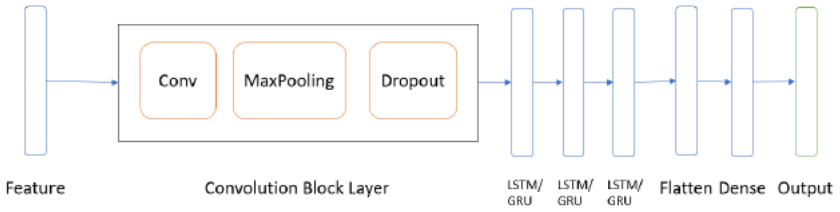

FIGURE 5.6: Proposed CNN with LSTM Architecture

5.6 Result with different feature extraction and learning algorithm

Feature Extraction Techniques	*Training Method*	*Epoch*	*Train*	*Valid*	*Test*
Mel-Spectrogram	CNN	120	0.8788	0.9023	0.901
	CNN + LSTM	120	0.9214	0.914	0.903
	VGG16	25	0.9831	0.9351	0.919
	VGG16 + LSTM	20	0.9335	0.8719	0.877
MFCC (13)	CNN	200	0.8949	0.8491	0.862
	CNN + LSTM	200	0.8559	0.8474	0.853
STFT	CNN	70	0.9589	0.9485	0.955
	CNN + LSTM	70	0.9283	0.9117	0.922

5.7 Graph between Training Data and Validation Data with respect to Accuracy and Loss

In this section, we are showing the graph for the best ones of each feature extractions data. With Mel-Spectogram Feature extractions data, we performed two networks (CNN and VGG16). Graph plots are shown below Fig 5.7 Fig 5.8.

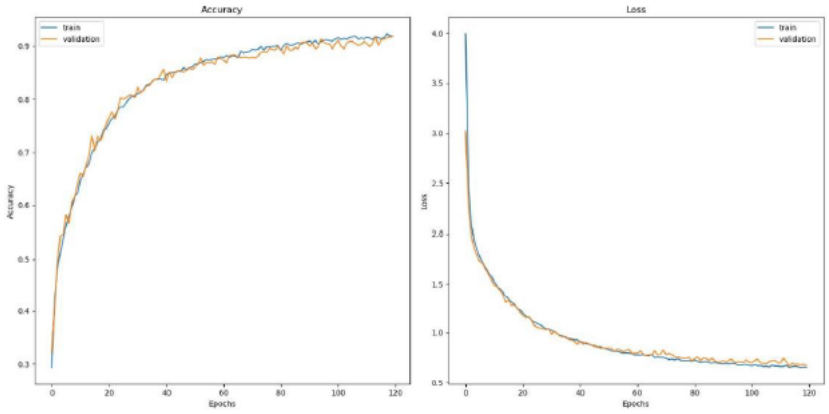

FIGURE 5.7: Graph plot for CNN with GRU trained network with Mel-Spectrogram

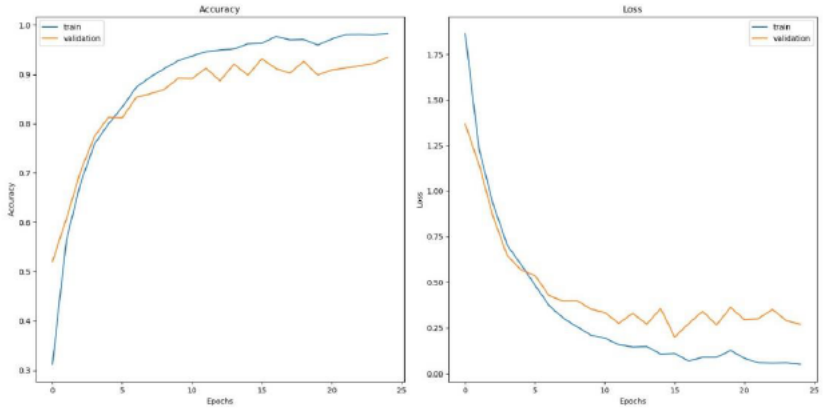

FIGURE 5.8: Graph plot for VGG16 trained network with Mel-Spectrogram input features

Chapter 6

Conculsion And Future Work

In this Project, we have applied k-NN machine learning algorithm, followed by CNN and RNN. we have used LSTM+CNN model for better Accuracy. Due to Limitation if my Laptop Configuration was not able to perform more no. of epochs. It was time consuming. For better achievement of result we can also use Bidirectional RNN. We have performed on few features like MFCCs, STFT. Other types of feature could also be investigated in terms of their relevance for this type of classification. Since genre classification between fairly different genres is quite successful, it makes sense to attempt finer classifications. The exact same techniques used in this project could be easily extended to classify music based on any other labelling, such as artist. In addition, including additional metadata text features such as album, song title, or lyrics could allow us to extend this to music mood classification as well.

Finally more work can be done on high quality datasets. We can also rreduce the size of pre-existing datasets combining with other datasets. Overall Music genre classification remains an interesting and challenging for both academic institution and business alike, and there is plenty of room for further study and analysis.

BIBLIOGRAPHY

.

[**1**] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, Shenglan Liu,*"Bottom-up Broadcast Neural Network For Music Genre Classification", ScienceDirect,Elsevier,Pattern Recognition, Jan, 2019*

[**2**] George Tzanetakis and Perry Cook , "Musical genre classification of audio signals. Speech and Audio Processing", IEEE transactions on, 10(5):293–302,(2002).

[**3**] Hareesh Bahuleyan. "Music genre classification using machine learning techniques". CoRR, abs/1804.01149, 2018

[**4**] N. Scaringella, G. Zoia, and D. Mlynek.*"Automatic genre classification of music content: a survey", IEEE Signal Process. Mag., 23(2):133141, 2006* .

[**5**] Nilesh M. Patil, Dr. Milind U. Nemade,*" Music Genre Classification Using MFCC, K-NN and SVM Classifier"*, International Journal Of Computer Engineering In Research Trends, Vol. 4, Issue. 2, Feb, 2017

[**6**] Mingwen Dong. "Convolutional neural network achieves human-level accuracy in music genre classification" .CoRR, abs/1802.09697, 2018.

[**7**] Omar Diab, Anthony Manero, and Reid Watson.*"Musical Genre Tag Classification With Curated and Crowdsourced Datasets". Stanford University, Computer Science, 1 edition, 2012.*

[**8**] Thomas Lidy and Alexander Schindler.*" Parallel convolutional neural networks for music genre and mood classification". MIREX2016.*

[**9**] Ahmed Elbir, Hilmi Bilal Cam, Mehmet Emre lyican, Berkay Ozturk, Nizamettin Aydin,*" Music Genre Classification and Recommendation by using Machine Learning Techniques",Innovations in Intelligent Systems and Applications Conference, Oct, 2018*

[**10**] Pradeep Kumar D, Sowmya B. J., Chetan, and K. G. Srinivasa ,*" A Comparative Study of Classifiers for Music Genre Classification based on Feature Extractors", IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics, Aug, 2016*