

Song Content Labeling using an Incremental Learning Approach

by

Anand Pol
202011006

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

INFORMATION AND COMMUNICATION TECHNOLOGY

to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY



May, 2022

Declaration

I hereby declare that

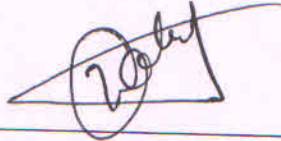
- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.

Anand

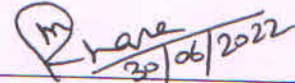
Anand Pol

Certificate

This is to certify that the thesis work entitled "Song Content Labeling using an Incremental Learning Approach" has been carried out by **Anand Pol (202011006)** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.



Dr. Bakul Gohel
Thesis Supervisor



Dr. Manish Khare
Thesis Co-Supervisor

Acknowledgments

Throughout my two-year M.Tech programme, I learned how to identify real-world computer science challenges and devise effective solutions. I would like to express my gratitude to the Dhirubhai Ambani Institute of Information and Communication Technology for this great opportunity and I assure my best efforts to contribute in this thesis.

First and foremost, I would like to express my heartfelt gratitude to my mentors, Dr. Bakul Gohel and Dr. Manish Khare, who have been a continual encouragement and motivation. Their research knowledge is immense, and it facilitated me in my work. They guided and mentored me with the right ideas and great insights when required. Lastly, I would like to express my gratitude to my family, professors, and friends for investing their time in helping me become a better professional. I appreciate all of your time and effort in assisting me in several situations. This opportunity, as well as your time, will be sincerely valued.

Contents

Abstract	v
List of Principal Symbols and Acronyms	vi
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Problem Statement	1
1.2 Objective and Contributions	2
1.3 Thesis Organization	2
2 Related Work	3
2.1 Video Classification Domain	3
2.2 Music Genre Classification Domain	3
2.3 Artist Classification Domain	4
2.4 Incremental Learning Approach	4
3 Methodology	7
3.1 Dataset	7
3.1.1 Splitting Songs into Vocal and Accompaniment components	9
3.1.2 Song Content labeling	9
3.1.3 Feature Extraction on Tagged Data	11
3.1.4 Format of the Dataset	11
3.2 Data Processing	12
3.3 Incremental Training Approach	13
4 Experiments, Results and Analysis	14
4.1 Binary Classification - Vocal/Music	14
4.1.1 Data Formation - Binary Classification	14

4.1.2	Incremental Learning - Binary Classification	15
4.1.3	Machine Learning/Deep Learning Approaches	16
4.1.4	Results - Binary Classification	16
4.2	Cross Language Experiment	19
4.2.1	Results - Cross-Language Experiment	19
4.3	Multi-Artist Classification	20
4.3.1	Data Formation - Multi-Artist Classification	20
4.3.2	Incremental Learning - Multi-Artist Classification	20
4.3.3	Machine Learning/Deep Learning Approaches	20
4.3.4	Similarity Based Approaches	21
4.3.5	Results - Multi-artist Classification	21
5	Conclusion and Future Work	25
	References	26

Abstract

Multimedia Data Content Labeling has been a research area for a relatively long period. Researchers have spent a substantial amount of time focusing on classifying video sequences, music genres, and artists' classification. Artist tagging is prevalent in this field, but it is addressed chiefly to Western music. This thesis mainly focuses on the artist classification and tagging of the Indian songs, specifically in the Hindi language. Looking at the increasing amount of data being recorded by the Hindi song industry, old methods are not efficient enough for the artist tagging. Incremental learning has been used quite widely now in multimedia data content labeling. The proposed algorithm is developed for music-vocal and multi-artist classifications using an incremental learning approach on the Hindi songs dataset. This incremental learning approach showed significantly good results. Cross-Language testing is also performed for Vocal/Music classification to check if the model generalizes over other language songs. Moreover, there are not enough datasets for performing such tasks for Hindi songs. Thus, a novel dataset is also introduced, containing window-based information labeled for each song for each artist. The dataset is primarily designed for tagging around twenty well-known Indian artists. This proposed method achieved remarkable accuracy of 83.6% and 55% for music-vocal and multi-artist classification on the test set, respectively.

List of Principal Symbols and Acronyms

ANN Artificial Neural Network

CS Cosine Similarity

SVC Support Vector Classifier

SVM Support Vector Machine

List of Tables

3.1	Artist Songs MetaData : Music/Vocal binary classification	8
3.2	Artist Songs MetaData : Multi-artist classification	8
3.3	Tagged Songs - 1	10
3.4	Dataset Format - 1	12
4.1	Incremental Training results 1 - SVC - Binary Classification	17
4.2	Incremental Training results 1 - ANN - Binary Classification	17
4.3	Incremental Training results 2 - ANN - Binary Classification	17
4.4	Incremental Training results 3 - ANN - Binary Classification	17
4.5	Incremental Training results 4 - ANN - Binary Classification	17
4.6	Classification Report of Binary Classification using ANN	18
4.7	Cross-Language Testing	19
4.8	Incremental Training results on set S3 Accuracy - Multi-Artist - Win- dow Level	22
4.9	Incremental Training results on set S3 Accuracy - Multi-Artist - Song Level	22
4.10	Similarity Based Approaches Test on set S3 - Multi-Artist	22
4.11	Classification Report of Multi-Artist Classification using CS - Ap- proach 1	23

List of Figures

2.1	Class Incremental Workflow [1]	6
4.1	Incremental Training Workflow	15
4.2	Confusion Matrix of Binary Classification using ANN	18
4.3	Confusion Matrix of Multi-Artist Classification using CS - Approach 1	24

CHAPTER 1

Introduction

Data Content labeling has been in the spotlight for quite a long period. Data is available in huge quantities and is not all tagged. Hence, tagging data is a challenging task to perform. Video classification involves classifying video clips into categories such as comedy, romantic, thriller, et cetera as well as emotion detection [2] and facial expression recognition domains [3], [4]. Music Genre classification is another domain where music is classified into different genres especially in western music as Pop, Rock, Indie Rock, EDM, Jazz, Country, Hip Hop & Rap, Classical Music, Latin Music, K-Pop et cetera [5], [6]. In Indian songs, music can be classified into genres like Pop, Carnatic, Gazhal, Classical, Semi-classical, Sufi, et cetera.

Substantial literature is available for western music, and very few on our Indian songs, especially Hindi. Datasets or relevant resources are not available for Hindi Songs, so a new Hindi Song dataset has been introduced, and its details are described in the Dataset section. The methodology section describes further processing of this dataset, tagging the data, and extracting features from this labeled data. Model selection, training, testing, and results are shown in the subsequent sections.

1.1 Problem Statement

The current requirement in the industry demand data to be tagged with accurate tags, and such a dataset is not available in the public domain. We have many songs available in the Hindi language, but the songs are not tagged with artist or music information corresponding to separate timestamps. Moreover, it is not possible to label all data manually. So, one of the solutions is to tag small data with tags, build a model, train it on this small data, and use that model to tag unknown data. If the model performs well, add that data to training data, train the model, and tag more unknown data again, essentially performing incremental

learning to train the model to learn different data patterns and help us simplify the BigData problem.

1.2 Objective and Contributions

This thesis aims to label music data using an incremental learning approach. It aims to achieve the following two objectives. 1) Music/Vocal Classification using Incremental Learning approach 2) Multi-artist Classification using Incremental Learning approach. The analysis of the Cross-Language experiment for Music/Vocal binary classification on the English songs dataset using the model obtained by Music/Vocal Classification on Hindi songs dataset is also performed. The novel contribution to this thesis work also includes the dataset generation by downloading songs, labeling every 15 seconds duration of a song with an appropriate label and extracting features thereafter using supported libraries. The incremental learning approach for both objectives mentioned earlier intends to address the BigData problem which is mentioned in the Section 1.1.

1.3 Thesis Organization

The thesis is further divided into chapters, with details as described. Chapter 2 describes the related work that is carried out in multimedia data content labeling and incremental learning approach domain. Chapter 3 describes the methodology and dataset description. Chapter 4 describes the experiments carried out for the music-vocal classification and multi-artist classification and the testing results for those experiments. Chapter 5 describes the conclusion of the thesis and the work that can be done further in this thesis in future.

CHAPTER 2

Related Work

The related work has been divided into four parts based on review performed on different types of multimedia content labeling and incremental learning approaches.

2.1 Video Classification Domain

Emotion detection using Deep CNN is introduced by Haider Riaz and Usman Akram in [2]. It uses multiple networks which operate in parallel and are merged, followed by relu, max-pool, drop-out, dense and soft-max layers. These multiple networks helped the model capture the global and local features from video sequences for emotion detection purposes. A comprehensive survey of Deep Facial Expression Recognition is covered in [3]. The authors have reviewed popular Datasets available for deep FER, also explained FER pipelines and static and dynamic image sequences. In [4], the authors proposed a FER technique in video sequence via Hybrid Learning. Two Deep CNN networks are employed, processing static facial images and other temporal CNN networks processing optical flow images. As a model, Deep Belief Model is used and SVM for classification purposes. In [7], the authors use Viola-jones algorithm to detect faces followed by tracking through Kanade-Lucas-Tomasi (KLT) algorithm. It used (HOG) features with a support vector machine (SVM) classifier for facial recognition. And, recognize facial expressions using the proposed light-weight convolutional neural network (CNN).

2.2 Music Genre Classification Domain

Convolutional recurrent neural networks (CRNNs) have been extensively used for Music Classification purposes. This CRNNs was introduced in [5], which take

advantage of CNNs for local feature extraction and RNNs for temporal summarization of the features which are extracted. The authors have shown that CRNNs show a strong performance concerning the number of parameters and training time, indicating the effectiveness of its hybrid structure in music feature extraction and feature summarisation. In [6], authors have proposed a hybrid architecture named the parallel recurrent convolutional neural network (PRCNN), which is an end-to-end training network that combines feature extraction and time-series data classification in one stage. In [8], the authors exploited the low-level information from spectrograms of audio. They developed a novel CNN architecture which considers multi-scale time-frequency information, transferring more appropriate semantic features to the decision-making layer to recognize the genre of an unidentified music clip.

2.3 Artist Classification Domain

For artist classification, the authors of paper [9] have used temporal structure in audio spectrograms using deep CNN and RNN models. Audio clip length is a new contribution in this paper, as are previously identified considerations like dataset split and feature level. Under the above conditions, a Convolutional Recurrent Neural Network (CRNNs) is deployed to the artist20 music artist identification dataset. In [10], the authors have used Cappella singing segments of 3-s duration for feature extraction. Each analyzing singing voice segment is divided into frames of 25 ms, with 10 ms overlapping in the preprocessing stage. Four Acoustic features are significant for artist identification, namely 1) Vibrato, 2) Harmonic spectral envelope, 3) Formants, 4) Timbre. We can extract Vibrato and the harmonic spectral envelope features from the pitch of the song. We can extract formants features from the shape of the vocal tract (where sound is filtered) and Timbral features from the shape and size of vocal cavities. In this project, I have extracted formants features using the OpenSmile library [11] available in Python. Timbral features are extracted using the Librosa library [12] available in Python.

2.4 Incremental Learning Approach

Incremental learning is a sound strategy of learning data patterns by the model incrementally as the data increases. In work [13], authors have introduced a new training strategy, iCaRL, that allows learning in a class-incremental way. Only the training data for a few classes must be present simultaneously, and new classes

can be added progressively. iCaRL learns robust classifiers and a data representation simultaneously. Experiments were carried out on CIFAR-100 and ImageNet ILSVRC 2012 datasets. In work [14], the authors first thoroughly analyzed the current state-of-the-art (iCaRL) method for incremental learning and demonstrated that the system's good performance was not because of the reasons presented in the existing literature. They concluded that the success of iCaRL is primarily due to knowledge distillation and recognized a fundamental limitation of knowledge distillation, i.e., it often leads to bias in classifiers. They proposed a dynamic threshold moving algorithm that can successfully remove this bias. In work [15], the authors proposed a Mahalanobis hyperelliptic multi-class incremental learning algorithm based on a support vector machine (MSVMIL). Only new samples are involved in successive training rounds in the incremental process. When there are new samples, the samples are classified by a historical classifier unrelated to the new sample and the Mahalanobis distance. In work [16], the authors proposed a support vector machine classifier based on online kernel optimization for sentiment classification. This classifier uses the proposed fuzzy boundary to modify the weight of the incoming review database for a specific duration.

In work [1], the authors have focussed on incremental class learning and tried to address the problem of catastrophic forgetting. They have introduced a novel framework called Broad Class Incremental Learning System (BCILS), which can address this issue. Authors claim this model is flexible and straightforward in structure and preserves old data patterns while learning new ones. The incremental work flow described in this paper is as shown in Figure 2.1. In work [17], the authors have proposed a algorithm called Learn++, which emphasizes on Ensemble learning technique to learn new data patterns while keeping intact the previous data patterns even if previous data is not available now. This algorithm can be applied on any neural network classifier and is independent of classifier used. In work [18], the authors state that various algorithms have been proposed for incremental learning without catastrophic forgetting. However, tests used to vary quite much, which were taken over small datasets only. So, the authors of this work have proposed a new evaluation system that compares five different mechanisms that can reduce catastrophic forgetting. In work [19], the authors have proposed an incremental support vector based on the Markov Resampling technique, abbreviated as (MR-ISVM). The authors infer that this approach shows high accuracy in classification tasks and runs faster than other ISVM-based approaches.

After doing a literature review on all the three multimedia domains described

above, we decided to move towards Artist Classification Domain. Specifically, the approach is to initially do binary classification on the new song by tagging every 15-second window of the song as artist or music class. Then tagging that window belonging to the artist class with a specific artist who has sung in that window, essentially multi-artist classification. It is an exciting problem to work with, and there is no substantial literature yet available in the community. Moreover to achieve the above mentioned goal, incremental learning is used to help resolve BigData problem to some extent. Incremental learning approach hasn't been used yet for Hindi songs dataset. Also, a novel dataset is constructed, details of which are described in the following section.

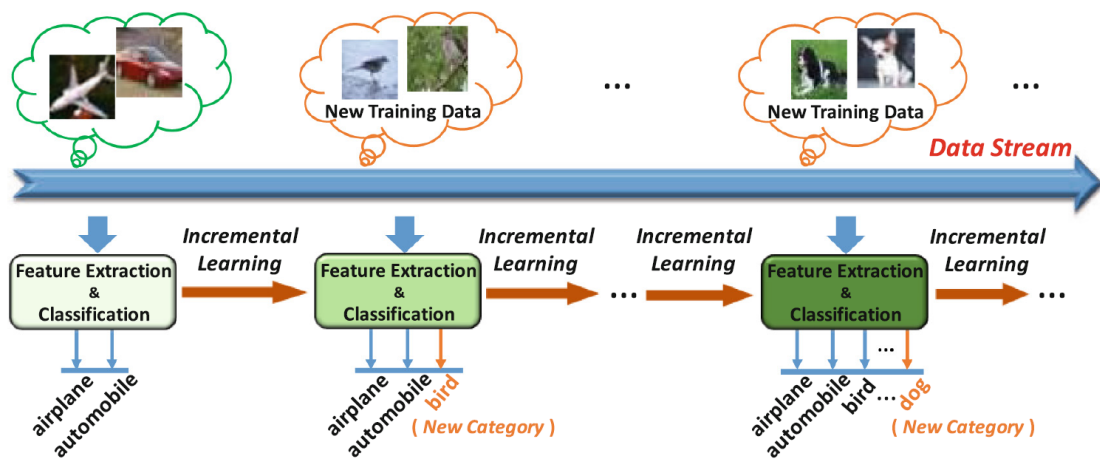


Figure 2.1: Class Incremental Workflow [1]

CHAPTER 3

Methodology

The methodology section consists of the following components: dataset generation and its details, dataset preprocessing, and the incremental learning approach used in music-vocal and multi-artist classification.

3.1 Dataset

Dataset introduced in this project is a novel contribution, and its relevant details are described hereafter. This dataset consists of song samples classified according to the artist. To hold enough data for each artist, around 7-15 songs are included according to their availability. Songs are downloaded using library Fmdpy available in python. All the songs are downloaded in either wav or opus format. Artists included in initial part of this work i.e. Binary Classification - music/vocal classes and its details regarding the numbers of songs per artist can be found in Table 3.1.

For multi-artist classification, more artists and number of songs per artist was required. So increased the number of artists from 13 to 20, and songs from 118 to 240. The updated table of number of songs per artist is as shown in Table 3.2.

Table 3.1: Artist Songs MetaData : Music/Vocal binary classification

Artist	Number of Songs
Amit Trivedi	6
Anuv Jain	6
Arijit Singh	13
Armaan Malik	10
Atif Aslam	8
Jasleen Royal	16
Kishore Kumar	11
Prateek Kuhad	7
Shreya Ghoshal	5
Sunidhi Chauhan	10
Mukesh	8
Mohammed Rafi	9
Mohit Chauhan	9

Table 3.2: Artist Songs MetaData : Multi-artist classification

Artist	Number of Songs
Amit Trivedi	11
Anuv Jain	7
Arijit Singh	16
Armaan Malik	13
Atif Aslam	12
Jasleen Royal	19
Kishore Kumar	14
Prateek Kuhad	10
Shreya Ghoshal	9
Sunidhi Chauhan	13
Jubin Nautiyal	11
Mukesh	10
Mohammed Rafi	12
Manna Dey	12
Udit Narayan	11
Sonu Nigam	11
Papon	12
KK	13
Mohit Chauhan	12
Kumar Sanu	14

3.1.1 Splitting Songs into Vocal and Accompaniment components

Artist Classification essentially requires information about an artist's vocal parts, so considered splitting every song into two parts. a) Vocal b) Accompaniment. For splitting the song into two parts, the Spleeter library [20] available in python serves the purpose. It has various configurations like two stems, three stems, four stems. This work prefers two stems configuration to split the song into two parts, vocal and accompaniment. In three stems, four stems, it splits the songs into more parts like drums, vocal, other music instrument stems, et cetera. The vocal part obtained after splitting is exclusively used in binary classification task described in further sections.

3.1.2 Song Content labeling

Songs are tagged manually for generating training data for the model. Each song of a particular artist is chosen and tagged every 15 seconds window with 0 or 1. 0 indicates that a particular artist whose chosen song is not singing in that 15 seconds window or there is a music part. 1 indicates that the particular artist is singing in that 15 seconds window at least more than 50 percent of the duration. So this way, the entire song is tagged with 0 and 1. This exact process is followed for around 118 songs and around 13 artists. So, songs corresponding to these 13 artists are manually tagged for generating training dataset. Subset of metaData about songs tagged is shown in Table 3.3.

Table 3.3: Tagged Songs - 1

Artist Name	0-15	15-30	30-45	45-1	1-1:15	1:15-1:30	1:30-1:45	1:45-2	2:00-2:15	2:15-2:30	2:30-2:45
Jasleen Royal	0	1	1	1	0	0	0	1	1	1	0
Jasleen Royal	1	1	1	1	1	1	0	1	1	1	0
Jasleen Royal	1	1	1	1	1	1	1	0	1	1	
Jasleen Royal	0	1	1	1	1	0	1	1	1	0	1
Jasleen Royal	0	1	0	0	0	1	0	1	1	0	0
Jasleen Royal	1	1	1	1	0	1	1	1	1		
Jasleen Royal	1	1	1	1	0	1	0	1	0	1	1
Jasleen Royal	1	1	1	0	1	1	1	0	1	1	1
Jasleen Royal	0	1	1	0	1	0	0	0	0	0	0
Jasleen Royal	1	0	0	0	1	0	0	0	0	0	0
Jasleen Royal	0	0	1	1	1	0	1	1	1	1	0
Jasleen Royal	0	1	1	1	0	0	1	1	1	0	1
Jasleen Royal	0	1	1	1	0	0	0	0	0	0	0
Jasleen Royal	1	1	1	1	1	1	1	0	0	1	1
Jasleen Royal	0	0	0	0	0	0	1	1	0	0	0
Jasleen Royal	0	1	1	1	1	0	0	0	0	1	1
Anuv Jain	0	1	1	1	0	1	1	1	1	1	1
Anuv Jain	0	1	1	1	1	1	1	0	1	1	1
Anuv Jain	0	1	1	1	1	1	1	1	1	1	1
Anuv Jain	0	1	1	1	1	1	0	1	1	1	1
Anuv Jain	0	1	1	1	1	1	1	1	1	1	1
Anuv Jain	0	1	1	0	1	1	1	1	1	1	1
Shreya Ghoshal	0	0	1	1	1	0	1	0	0	1	1
Shreya Ghoshal	0	0	1	1	0	1	1	1	0	0	0
Shreya Ghoshal	0	0	0	1	1	1	0	0	1	1	0
Shreya Ghoshal	0	1	1	1	0	0	0	0	1	1	1
Shreya Ghoshal	0	1	1	1	0	1	1	1	1	1	0
prateek kuhad	0	0	0	0	1	0	1	0	1	1	1
prateek kuhad	0	1	1	1	1	1	1	1	0	1	1
prateek kuhad	0	1	1	1	1	1	1	0	1	0	
prateek kuhad	0	1	1	1	1	1	1	1	1	1	1
prateek kuhad	0	1	1	1	1	1	1	1	1	0	0
prateek kuhad	0	1	1	1	0	0	0	0	1	1	1
prateek kuhad	0	1	1	1	1	1	1	1	0	1	1
Sunidhi Chauhan	0	0	0	0	0	1	1	0	0	0	1
Sunidhi Chauhan	0	0	0	0	1	0	0	0	0	0	1
Sunidhi Chauhan	0	0	0	0	0	0	0	0	0	0	0
Sunidhi Chauhan	0	0	1	1	1	1	0	1	1	1	0
Sunidhi Chauhan	1	1	1	1	0	0	1	1	1	1	0
Sunidhi Chauhan	0	1	1	1	0	0	1	1	1	1	0
Sunidhi Chauhan	0	1	1	1	1	0	1	1	1	1	1
Sunidhi Chauhan	0	0	0	1	0	0	0	0	0	0	0
Sunidhi Chauhan	0	0	1	1	1	1	1	1	1	1	1
Sunidhi Chauhan	0	0	0	1	1	1	0	0	1	1	1
Armaan Malik	0	1	1	1	1	0	0	1	1	1	1
Armaan Malik	0	0	0	1	0	0	1	1	0	0	0
Armaan Malik	0	0	1	1	1	1	0	0	1	1	1
Armaan Malik	0	1	1	1	1	1	0	0	1	0	1
Armaan Malik	0	0	1	1	1	0	0	1	1	1	1
Armaan Malik	0	0	1	0	1	1	0	1	1	1	0
Armaan Malik	0	1	1	1	1	1	1	1	1	1	0
Armaan Malik	0	0	1	0	1	0	0	0	0	1	0
Armaan Malik	0	0	1	1	1	0	0	1	1	1	0
Armaan Malik	0	0	0	0	0	0	0	1	0	0	0
Atif Aslam	1	1	0	0	1	1	1	0	0	1	1
Atif Aslam	0	1	1	1	1	1	1	1	1	1	0
Atif Aslam	0	1	1	1	1	1	1	0	1	1	1
Atif Aslam	0	1	1	1	1	1	1	0	0	1	1
Atif Aslam	0	0	1	1	0	1	1	1	0	1	1
Atif Aslam	0	0	1	1	0	0	0	1	1	1	1
Atif Aslam	0	0	0	1	1	0	1	1	0	1	1
Atif Aslam	0	1	1	1	1	1	0	1	1	1	1

3.1.3 Feature Extraction on Tagged Data

Features extracted from vocal parts play a crucial role in identifying the artist. Features such as formants, timbral features are extracted from data. The OpenSmile library [11] is used to extract formants and pull F0, F1, F2, F3 frequencies and amplitude values. Timbral features such as loudness, spectral roll-off, spectral centroid, spectral bandwidth, mfcc, malspectrogram, chroma stft, spectral contrast, et cetera are extracted using the Librosa library [12]. Every 15 seconds of a song is classified yes for a particular artist if the tag for that duration is 1, and the features are extracted using OpenSmile and Librosa library. If the tag is 0, then features are extracted, addressing it as a music part. Total 30 features are generated for each 15 seconds window.

3.1.4 Format of the Dataset

Each record in the dataset consists of information like the target artist, and the rest of all columns are features extracted using OpenSmile and Librosa library. The dataset is created by extracting features for tag 1 in song as artist's features and tag 0 as music class features. So, the dataset has 14 classes having 13 artists classes plus one music class. However, since our initial aim is to perform binary music and vocal class classification, all artist classes can be merged into one class and named an artist class. So, the dataset is ready, and based on the steps mentioned in the subsequent section; the dataset is divided and processed for binary classification between artist and music segment. Format of dataset with some of the features can be referred in Table 3.4.

Table 3.4: Dataset Format - 1

Artist Name	Loudness_sma3	alphaRatio_sma3	hammarbergIndex_sma3	slope0-500_sma3	slope500-1500_sma3
Jasleen Royal	2.193476439	-22.72520638	31.24501228	0.0824297294	-0.0329759717
Jasleen Royal	0.491710037	-8.024303436	12.29613304	0.1010247841	0.002715123817
Jasleen Royal	1.318520069	-22.24769974	31.80542183	0.07193505764	-0.02149777301
Jasleen Royal	1.30722177	-22.248209	33.43047714	0.1220137775	-0.01553924941
Jasleen Royal	1.39886117	-25.65386009	38.44342041	0.09930077195	-0.02274668775
Jasleen Royal	0.7823596597	-28.13183784	40.45317078	0.07778877765	-0.01774762757
Jasleen Royal	1.653035164	-20.30538559	31.21156693	0.1215424836	-0.0207118988
Jasleen Royal	1.706402659	-17.09330559	30.40633011	0.1454815716	-0.01820234023
Jasleen Royal	1.467864394	-19.26259995	31.10281372	0.1166429296	-0.01224905532
Jasleen Royal	1.561077476	-20.9222908	35.86524963	0.1376594752	-0.01790407486
Jasleen Royal	1.312561035	-19.78407478	29.82011414	0.06587164104	-0.02677088045
Jasleen Royal	0.9381251335	-31.24310684	40.29401398	0.104214184	-0.01688401215
Jasleen Royal	0.602471292	-14.20999527	20.28425789	0.1166228801	-0.002041369211
Jasleen Royal	1.166097999	-8.394992828	16.20210648	0.1070541814	-0.01081278455
Jasleen Royal	2.065360069	-8.035049438	21.12445641	0.1439068466	-0.005001437385
Jasleen Royal	1.898235083	-9.34643364	20.15898705	0.1401290894	-0.02579152957
Jasleen Royal	1.197060347	-6.532259941	16.94603539	0.1340728253	0.001448100433
Jasleen Royal	1.135908246	4.258196831	28.98540306	0.08176780492	0.01659990288
Jasleen Royal	2.050433874	-11.65597153	19.21340942	0.08249179274	-0.0167331397
Jasleen Royal	1.781062484	-14.4419651	23.78380013	0.06729819626	-0.01454164088
Jasleen Royal	2.383005619	-9.055347443	20.06546783	0.1110791489	-0.006706658751
Jasleen Royal	1.774306536	-15.0092411	26.40476608	0.1323906034	-0.01260859426
Jasleen Royal	1.772139907	-16.87109184	28.34572983	0.1217311174	-0.01601335965
Jasleen Royal	1.962142825	-13.23904228	24.06114197	0.1079087108	-0.01115555596
Jasleen Royal	2.483844757	-11.67879295	24.91646385	0.1228075102	-0.01076346263
Jasleen Royal	2.806812048	-13.33424187	25.65607071	0.1258512437	-0.02204133011
Jasleen Royal	2.224917412	-15.93800545	27.26138496	0.1254665852	-0.01365080196
Jasleen Royal	2.732274771	-14.47491074	24.14958954	0.1231629401	-0.007566614542
Anuv Jain	1.722102642	-5.838017464	15.02845192	0.04240595922	-0.01212095283
Anuv Jain	1.677452326	-3.310821056	12.63654137	0.05841619894	-0.003796919482
Anuv Jain	1.308140993	-9.524647713	22.70072365	0.0167798046	-0.004729540087
Anuv Jain	1.178938746	-3.899378777	14.62257195	0.09410982579	8.77e-6
Anuv Jain	1.495494962	-6.06978178	17.31287956	0.08151956648	-0.006341732573
Anuv Jain	1.203957796	-7.204737186	16.80909348	0.06204603985	-0.006324884016
Anuv Jain	1.457585812	-8.154848099	16.42726326	0.05559688061	-0.004782228265
Anuv Jain	2.041921139	-8.703738213	18.30739212	0.09445524961	-0.0101905046
Anuv Jain	2.231897593	-7.298566818	15.14147854	0.10882999	-0.01292656828
Anuv Jain	1.370794654	-4.545657158	13.53403568	0.08130260557	-0.006178014446

3.2 Data Processing

As a part of data preprocessing, data standardization is performed using the StandardScaler object. All values are standardized, which helps for giving equal importance to all input variables and not biasing the feature's importance solely based on numerical values. The same StandardScaler object is used to standardize test dataset.

3.3 Incremental Training Approach

Incremental training is an approach to train the model initially with some preliminary data and then tag new unlabeled data incrementally with the obtained model. This approach involves continuous learning of the model and learning more and more data patterns. Incremental learning is used in this work to address the BigData problem, i.e., initially, we have very few labeled song data using which the model is trained. Now, the trained model predicts the labels for next set of data. After labeling this set of data with labels using the model, this data is also added to already available training data. So training data is now increased, and the model is now trained over this new training set. This process is repeated continuously. So, there is no human interference in labeling new data apart from labeling at a initial stage. Incremental learning is used in this work for binary classification of 15 seconds window in unlabeled data. Songs belong to different artists given for training, and training set initially don't have songs belonging to all the 13 artists included in this work. So, model learns different artists' data patterns with the help of incremental training. Model has to finally learn different artists' vocal part and yet has to classify it into same artist class, and also music component into music class which is a task here for binary classification.

CHAPTER 4

Experiments, Results and Analysis

This section includes experiments on vocal/music classification and multi-artist classification and their respective dataset formation. Cross-Language testing is also performed to check the model's generalization, and its details are described further in this section. The corresponding method's results and analysis are shown below in their respective sections itself.

4.1 Binary Classification - Vocal/Music

The task is to classify every 15 seconds duration of the song as vocal or music class. The further parts in this section explain dataset formation, incremental learning approach used in binary classification, cross-language experiment, and results obtained for the test dataset.

4.1.1 Data Formation - Binary Classification

The model aims to predict the class for every 15 seconds duration window of the song with either artist or music class. Training Data is ready as mentioned in section 3.1.4. A total of 30 features are extracted for each 15 seconds duration from the vocal part of the song. The empty part in the vocal segment denotes that the music is present in that window, so the feature extracted from such a window is tagged as music class features. After extracting the features for all the 118 songs, the dataset is generated of 2000 such pieces, with vocal records counting as 1226 and music class counting as 774, each having 30 features. The incremental training approach used for further training is described in the following section.

4.1.2 Incremental Learning - Binary Classification

The dataset obtained above is divided into four sets - D1, D2, D3, and D4 for incremental training. Each set having around 500 pieces of data containing vocal as well as music classes. The incremental training follows this workflow.

1. Train the model on the D1 dataset
2. Predict labels for D2, D3, D4 datasets using the obtained model and note the accuracy
3. Train the model on D1 + Predicted D2 dataset labels
4. Predict labels for D3, D4 datasets and note the accuracy
5. Train the model over D1 + Predicted D2 + Predicted D3 dataset labels
6. Predict labels for the D4 dataset and note the accuracy

Before training the model, the dataset is standardized with the StandardScaler object, and the same object is used to standardize the test data before predicting its labels. This entire above process involves incremental training on data and analysis of incremental training on D4 data test accuracy. For showcasing the trend, different combination of train and test dataset has been tried, and obtained expected results i.e. increasing trend of accuracy as shown in section. Refer figure 4.1 for incremental training workflow.

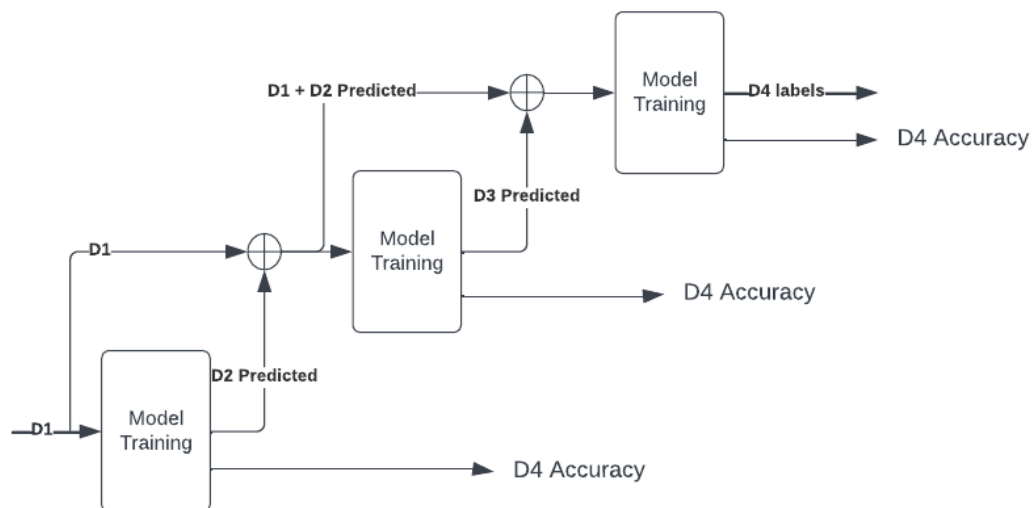


Figure 4.1: Incremental Training Workflow

4.1.3 Machine Learning/Deep Learning Approaches

Incremental training is performed as described in the above section. Various Experiments are carried out with Machine Learning and Deep Learning approaches. In first experiment, Incremental training was performed with SVC as a base model. The model was tuned with hyper-parameter tuning using the GridSearchCv method. Next, experiments were carried out with a deep learning approach using an Artificial Neural Network. Various experiments were carried out for finding optimum parameter values like batch size, learning rate, the number of dense layers, optimization techniques, regularization parameters, et. cetera, and the parameters corresponding to the best results were chosen at the end. ANN approach is implemented for multiple dataset combination for training and testing sets for the purpose of validating the analysis of incremental training approach using cross-validation. Some of the technical details for the best model in Music/Vocal Classification, i.e., the ANN model is the batch size as 4, learning rate 0.001, epochs as 100, number of units in dense layers as 128, 64, 32 ... 1, tanh regularizers for regularization purpose, Adam optimizer as an optimization technique and binary cross entropy loss function for loss estimation. For the SVC model, I used the GridSearchCv technique to find its best parameters for optimum results, and some technical details about the same is the kernel as rbf, which is a non-linear kernel, gamma value as 0.01, and regularization C value as 100.

4.1.4 Results - Binary Classification

The results obtained after hyperparameter tuning the SVC model for binary classification are shown in Table 4.1. Results of ANN approach for binary classification are as shown in Table 4.2-4.5. Results show that the ANN model performs exceptionally well than the SVC model in the incremental training approach for binary classification. SVC model also shows an increasing trend in the accuracy of test set D4, but the rate of increase is less. ANN model shows a gradual increase in the accuracy of test set D4 and outperforms the SVC model and can be seen from Table 4.1 and Table 4.2. Another observation was that the SVC model predicts the older data less accurately when trained on more new data. In contrast, the ANN model performs comparatively well in the same scenario. Find the classification report of Binary Classification using ANN in Table 4.6. Also, find the confusion matrix information of Binary Classification using ANN in Figure 4.2.

Table 4.1: Incremental Training results 1 - SVC - Binary Classification

Incremental Training	D4 set Accuracy
D1	0.742
D1 + D2 predicted	0.783
D1 + D2 predicted + D3 predicted	0.784

Table 4.2: Incremental Training results 1 - ANN - Binary Classification

Incremental Training	D4 set Accuracy
D1	0.782
D1 + D2 predicted	0.822
D1 + D2 predicted + D3 predicted	0.836

Table 4.3: Incremental Training results 2 - ANN - Binary Classification

Incremental Training	D3 set Accuracy
D4	0.643
D4 + D1 predicted	0.711
D4 + D1 predicted + D2 predicted	0.721

Table 4.4: Incremental Training results 3 - ANN - Binary Classification

Incremental Training	D2 set Accuracy
D3	0.655
D3 + D4 predicted	0.725
D3 + D4 predicted + D1 predicted	0.741

Table 4.5: Incremental Training results 4 - ANN - Binary Classification

Incremental Training	D1 set Accuracy
D2	0.735
D2 + D3 predicted	0.762
D2 + D3 predicted + D4 predicted	0.772

Table 4.6: Classification Report of Binary Classification using ANN

	precision	recall	f1-score	support
music	0.85	0.62	0.72	387
vocal	0.80	0.93	0.86	613
accuracy			0.81	1000
macro avg	0.83	0.78	0.79	1000
weighted avg	0.82	0.81	0.81	1000

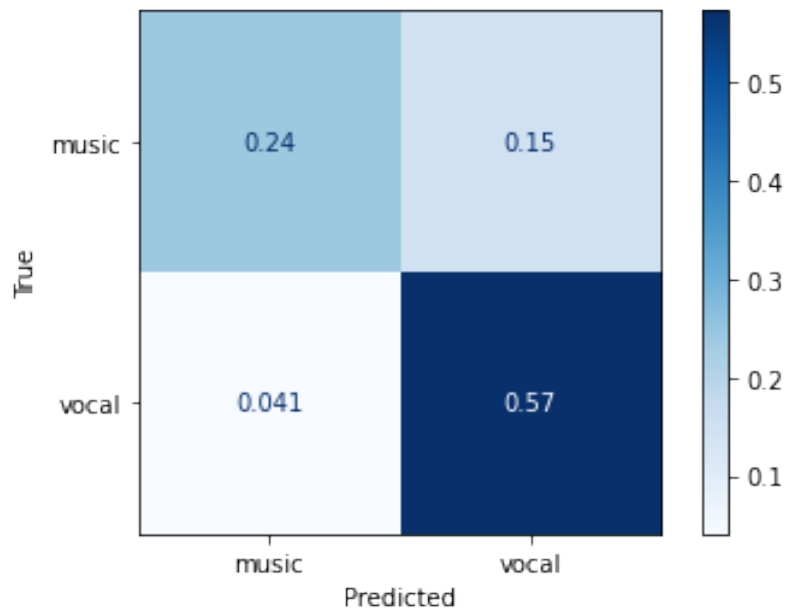


Figure 4.2: Confusion Matrix of Binary Classification using ANN

4.2 Cross Language Experiment

The model trained on the Hindi songs dataset is tested over other language songs like English songs for cross-language testing. This experiment aims to analyze if the model trained on Hindi songs can be used to solve the binary class problem on English songs. A total of 20 English songs are downloaded, and the actual labels of every 15-second window of English songs are obtained by manual labeling following the process in the section 3.1.2. The model predicts the labels following the same process for Hindi songs, i.e., the features are extracted for every 15-second window for each song using the libraries mentioned in the section 3.1.3. Features generated are then fetched to the model for label prediction by the first standard scaling the input with the training StandardScaler object. Dataset generated is similar as shown in Table 3.4. The results and analysis is as shown in following section.

4.2.1 Results - Cross-Language Experiment

Results of Cross-Language testing are shown in the Table 4.7, which states that the overall accuracy of the binary classification of all pieces of English songs is around 47%, which is relatively much low as compared to Hindi songs pieces - 83.6%. The F1 score and recall for the music class are pretty low, though the precision for the artist class is considerable; for the music class, it is low. The results infer that the model generated on the Hindi songs dataset is language-specific and cannot be generalized to other languages.

Table 4.7: Cross-Language Testing

Class	precision	recall	F1-score	Accuracy
artist	0.79	0.46	0.58	0.47
music	0.17	0.47	0.26	

4.3 Multi-Artist Classification

The task is to classify the 15 seconds duration of the song into a particular artist if that 15 seconds duration is earlier classified as a vocal class by the model. The further parts in this section explain dataset formation, experiments, and results obtained on the test dataset.

4.3.1 Data Formation - Multi-Artist Classification

The model obtained in the section 4.1.2 by training the model on the D1 dataset is used to label each record with either artist/music class in D2, D3, and D4 datasets. The music records from these datasets are discarded as they are not required for multi-artist classification. So the final dataset for multi-artist classification only consists of records of artists' classes, and they are labeled with their respective artist classes with the preliminary information we have about the songs, i.e., when the song is downloaded, we have information that this song belongs to a particular artist. So, all the windows in this song that are tagged as artist class by the model obtained above is tagged with this particular artist label. This process is repeated for every artist's song, and we get the final dataset. The final dataset is divided into 3 sets S1, S2 and S3. S1 contains artist pieces corresponding to nearly 120 songs, and S2 and S3 contains artist pieces corresponding to nearly 60 songs each. Each dataset contains songs segments of all 20 artists, which is required for training purposes to maintain low variance for the test set.

4.3.2 Incremental Learning - Multi-Artist Classification

Incremental training is similar as described in the previous section. The only difference is that there are 20 classes to classify here, whereas there were only 2 in the previous section. Artificial neural networks, SVM, XGBoost, and Cosine Similarity-based approaches have been used for multi-artists classification, which are described in successive sections.

4.3.3 Machine Learning/Deep Learning Approaches

Artificial neural network architecture is constructed and fine-tuned with the training dataset. Regularization techniques, optimization such as Adam optimizers with a suitable learning rate, and loss as a categorical cross-entropy are used. In the following approach, the SVM model is trained over the dataset. GridSearchCv

is used for hyperparameter tuning. In the following approach, XGBoost is trained over the dataset. The results of all these Machine Learning and Deep Learning approaches have been reported in the respective sections. Some of the technical details for the ANN model trained for Multi-Artist Classification is the batch size as 32, learning rate 0.001, epochs as 100, number of units in dense layers as 512, 256, 128, 64, 32 ... 4, 20, activity regularizers for regularization purpose, Adam optimizer as an optimization technique and categorical cross entropy loss function for loss estimation.

4.3.4 Similarity Based Approaches

Cosine Similarity is used in a similarity-based approach. In this approach, no training is required. The cosine similarity of the test segment with all available train segments of a single artist is taken. Three variants have been tried for Cosine Similarity based approach, CS - Max, CS - Mean, CS - Top 5 Mean. In CS - Max, we take the maximum of all cosine similarities obtained above and report that for the corresponding artist. In CS - Mean, we take the mean of all cosine similarities obtained above and report that for the corresponding artist. In CS - Top 5 Mean, we take the mean of top 5 cosine similarities obtained above and report that for the corresponding artist.

After choosing one of the approach from above, this process is performed for every artist, and the artist corresponding to maximum cosine similarity is given as a label for that test window. This process is repeated for all test segments, and test results are generated. The results of all three approaches described above are shown in Table 4.10.

4.3.5 Results - Multi-artist Classification

The results of various approaches for multi-artist classification are shown in Table 4.8. SVM and XGBoost models show lower accuracy than other two approaches. Results show that the similarity-based approach - cosine similarity performs comparatively well than the ANN, SVM & XGBoost approaches. The cosine similarity approach does not require any training, which saves training time and is computationally proficient. ANN also performs well than the other two machine learning approaches. This training is based on window level approach, while the comparison of window level and song level approaches is shown later in this section.

Table 4.8: Incremental Training results on set S3 Accuracy - Multi-Artist - Window Level

Incremental Training	ANN	SVM	XGBoost	Cosine Similarity
S1	0.52	0.49	0.49	0.55
S1 + S2 predicted	0.52	0.45	0.46	0.55

The above results on Multi-artist classification were on window level training, but if the model is trained on song level, i.e., no two segments of the same song will be in train, and test split, all the pieces of a single song will be in either train split or test split then the results decreases due to increase in test set variance. This trend is also observed in paper [9], where they find accuracies for song level and album level model training. Their results are also quite less in the case of the album level approach than the results at the song level approach due to an increase in test set variance. Find the results in the Table 4.9. The results are quite less and can be observed for all the models' performances.

Table 4.9: Incremental Training results on set S3 Accuracy - Multi-Artist - Song Level

Incremental Training	ANN	SVM	XGBoost	Cosine Similarity
S1	0.28	0.30	0.30	0.23
S1 + S2 predicted	0.29	0.28	0.28	0.24

The results of different approach of cosine similarity in Multi-artist classification problem can be seen in table below. It is quite prominent from the results that the first approach gives much better results than approach 2 and approach 3. Taking mean and Top 5 mean in cosine similarity approach gives much less results, so final results chosen are of approach 1. So for further usecases, approach 1 is selected as final approach.

Table 4.10: Similarity Based Approaches Test on set S3 - Multi-Artist

Incremental Training	CS using Max	CS using Mean	CS using Top 5 Mean
S1	0.55	0.16	0.12
S1 + S2 predicted	0.55	0.18	0.12

The classification report for Multi-artist Classification using cosine similarity based approach 1 is shown in Table 4.11 , the confusion matrix for the same can be found in Figure 4.3.

Table 4.11: Classification Report of Multi-Artist Classification using CS - Approach 1

	precision	recall	f1-score	support
Amit Trivedi	0.22	0.29	0.25	14
Anuv Jain	0.50	0.73	0.59	15
Arijit Singh	0.51	0.57	0.54	37
Armaan Malik	0.67	0.59	0.62	41
Atif Aslam	0.50	0.48	0.49	42
Jasleen Royal	0.58	0.71	0.64	31
Jubin Nautiyal	0.54	0.51	0.52	43
KK	0.60	0.54	0.57	57
Kishore Kumar	0.72	0.74	0.73	35
Kumar Sanu	0.38	0.42	0.40	57
Manna Dey	0.65	0.58	0.61	38
Mohammed Rafi	0.70	0.77	0.73	30
Mohit Chauhan	0.58	0.54	0.56	39
Mukesh	0.70	0.66	0.68	29
Papon	0.53	0.56	0.54	50
Shreya Ghoshal	0.64	0.48	0.55	29
Sonu Nigam	0.46	0.47	0.46	45
Sunidhi Chauhan	0.67	0.56	0.61	36
Udit Narayan	0.47	0.48	0.47	44
Prateek Kuhad	0.44	0.42	0.43	26
accuracy			0.55	738
macro avg	0.55	0.55	0.55	738
weighted avg	0.56	0.55	0.55	738

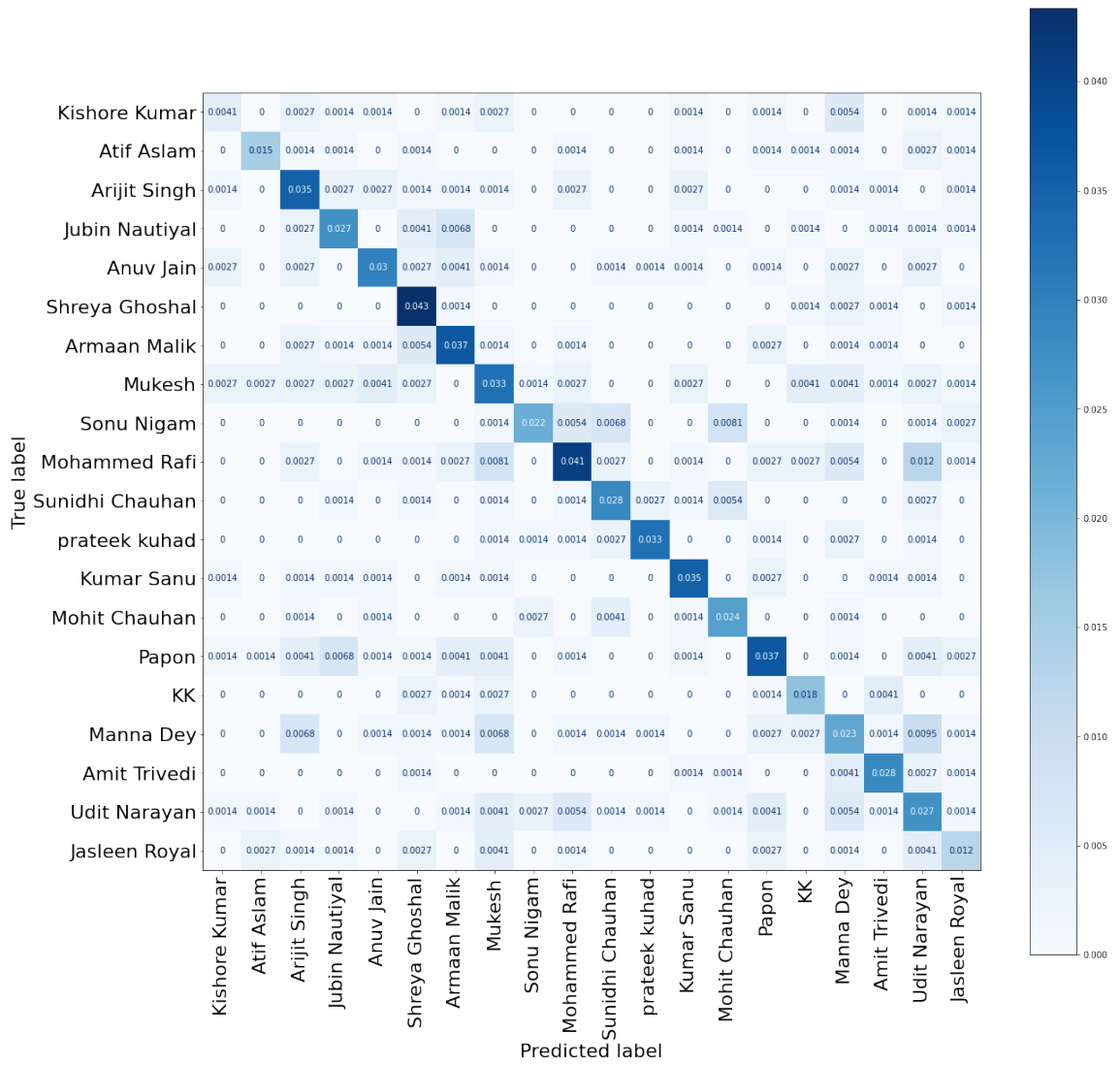


Figure 4.3: Confusion Matrix of Multi-Artist Classification using CS - Approach 1

CHAPTER 5

Conclusion and Future Work

The results are pretty prominent to showcase that incremental training shows an increasing trend in test set accuracy for Binary Classification. So, this infers that the BigData labeling problem can be addressed using the Incremental training approach. More such 15 seconds windows can be labeled in the new songs using the ANN model for vocal/music class tagging, showing good accuracy. This labeling can help extract information about vocal segments and help extract further details of the musical components like Genre et. cetera. Also, models for multi-artist classification shows moderate results, where the cosine similarity-based approach performed quite well even without training compared to other non-similarity based methods. Moreover, an experiment with Cross-Language songs concludes that the model is language-specific and cannot be generalized to other languages' songs.

The dataset created was 240 songs, out of which 120 pieces were manually labeled, and the rest 120 songs were labeled using the model generated in this thesis. This dataset is not enough for larger-scale testing, and hence future work requires increasing the number of songs in the training and testing set and adding more artists to increase variance in data. Also, a strategy can be developed for classifying an artist into some similar artist or classifying it as an unknown artist if the artist is not among the artists available in the dataset.

References

- [1] W. Liu, H. Yang, Y. Sun, and C. Sun. A broad neural network structure for class incremental learning. In *International Symposium on Neural Networks*, pages 229–238. Springer, 2018.
- [2] H. Riaz and U. Akram. Emotion detection in videos using non sequential deep convolutional neural network. In *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–6, 2018.
- [3] S. Li and W. Deng. Deep facial expression recognition: A survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.
- [4] S. Zhang, X. Pan, Y. Cui, X. Zhao, and L. Liu. Learning affective video features for facial expression recognition via hybrid deep learning. *IEEE Access*, 7:32297–32304, 2019.
- [5] K. Choi, G. Fazekas, M. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396, 2017.
- [6] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo. Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices. *IEEE Access*, 8:19629–19637, 2020.
- [7] M. Sajjad, S. Zahir, A. Ullah, Z. Akhtar, and K. Muhammad. Human behavior understanding in big multimedia data using cnn based facial expression recognition. *Mobile Networks and Applications*, 25(4):1611–1621, Aug 2020.
- [8] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu. Bottom-up broadcast neural network for music genre classification. *Multimedia Tools and Applications*, 80(5):7313–7331, Feb 2021.
- [9] Z. Nasrullah and Y. Zhao. Music artist classification with convolutional recurrent neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

- [10] D. Y. Loni and S. Subbaraman. Robust singer identification of indian playback singers. *EURASIP Journal on Audio, Speech, and Music Processing*, 2019(1):10, Jun 2019.
- [11] F. Eyben, M. Wöllmer, and B. Schuller. openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *Proc. of ACM Multimedia 2010*, pages 1459–1462, Florence, Italy, 2010. ACM.
- [12] 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*, volume 8, 2015.
- [13] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- [14] K. Javed and F. Shafait. Revisiting distillation and incremental classifier learning. In *Asian conference on computer vision*, pages 3–17. Springer, 2018.
- [15] Y. Qin, D. Li, and A. Zhang. A new svm multiclass incremental learning algorithm. *Mathematical Problems in Engineering*, 2015, 2015.
- [16] R. K. Thakur and M. V. Deshpande. Oko-svm: Online kernel optimization-based support vector machine for the incremental learning and classification of the sentiments in the train reviews. *International Journal of Modeling, Simulation, and Scientific Computing*, 9(06):1850054, 2018.
- [17] R. Polikar, J. Byorick, S. Krause, A. Marino, and M. Moreton. Learn++: A classifier independent incremental learning algorithm for supervised neural networks. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 2, pages 1742–1747. IEEE, 2002.
- [18] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [19] J. Xu, C. Xu, B. Zou, Y. Y. Tang, J. Peng, and X. You. New incremental learning algorithm with support vector machines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2230–2241, 2018.

- [20] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020. Deezer Research.