AUTOMATIC MUSIC INSTRUMENT RECOGNITION

by

G. MOHAN TEJA	19BEC1133
LAVANYA SINGH	19BEC1188
THEJASVINI.S	19BEC1372
AKSHAJ PRASAD	19BEC1444

A project report submitted to

Dr. JOHN SAHAYA RANI ALEX

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

ECE2006 – Digital Signal Processing

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

OCTOBER 2021

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. John Sahaya Rani Alex**, Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. A. Sivasubramanian,** Dean of the School of Electronics Engineering (SENSE), VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Programme Chair **Dr. P. Vetrivelan** for their support throughout the course of this project. We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course. We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

G. MOHAN TEJA | **LAVANYA SINGH**

THEJASVINI. S | **AKSHAJ PRASAD**

ABSTRACT

Musical instruments come in various sizes and shapes and the characteristics of them can sound very similar or sometimes very distinct. The same type of musical instrument can make a wide variety of sounds based on the material it is made of and the way it is used. Automatic sound source recognition plays an important role in developing automatic indexing and database retrieval applications. These applications have potential in saving the humans from time taking searches through huge amounts of digital audio material available today.

In this project we have shown that musical instruments come in various sizes and shapes and can make a wide variety of sounds based on the material it is made and the way it is used. We have used machine learning models to compare different characteristics of musical instruments and study its ability to distinguish different instruments. The project pipeline involves obtaining musical instrument data by using MFCC and DFT in processing frequency domain and feature classification. Monophonic and monotimbral signals were classified while eliminating noise and silence.

.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVES AND GOALS

The main aim of the project is to achieve Musical Instrument Recognition using signal processing in frequency domain and feature classification. Computer audition is the general study of the systems and methods necessary for audio understanding by a machine. In a sense, computer audition concerns itself with the study of designing computers that can “hear” as humans do.

The ultimate goal is a machine that can organize what it hears; learn names for recognizable objects, actions, events, places, musical styles, instruments, and speakers; and retrieve sounds by reference to those names. We aim to classify monophonic and monotimbral signals from the input audio while eliminating the presence of noise and silence in the same.

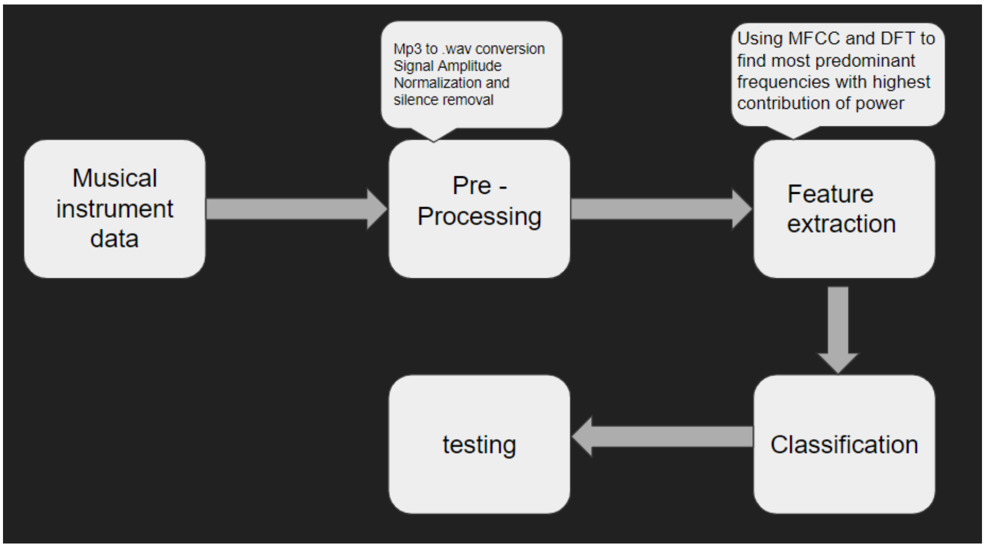
1.3 APPLICATIONS

Musical instrument recognition and sound source recognition are essential parts of computational auditory scene analysis (CASA). In this field, the goal is to analyse complex acoustic environments, including the recognition of overlapping sound events, and thus their sources. In musical synthesis, the model parameters are often analysed from an acoustic signal. There might be potential in combining these two fields, using physical model synthesis parameters for musical instrument recognition and bringing new methods for feature extraction from musical instrument recognition to physical modeling.

Musical instrument recognition is related to many other fields of research. The methods used in implementing musical instrument recognition systems are drawn from different technical areas. The pre-processing and feature extraction techniques can be taken from speech and speaker recognition. Commonly, classification is performed with statistical pattern recognition techniques. Neural networks and other soft computing techniques have also been applied in many cases.

2. DESIGN

2.1 BLOCK DIAGRAM



2.2 SOFTWARE ANALYSIS

The datasets were trained and tested after which four machine learning models namely, SVM, Ridge, Linear Regression and Lasso, were employed in order to calculate precision.

DATA SET

The dataset used was derived from the following website :
https://philharmonia.co.uk/resources/sound-samples/

This data set consists of 7 musical instruments which are :

Cello - 889 samples

Clarinet - 846 samples

Flute - 878 samples

Guitar - 106 samples

Saxophone - 732 samples

Trumpet - 485 samples

Violin - 1502 samples

LANGUAGES & TOOL KIT

These were the Languages/Modules that were used for this project

- Bash
- Matlab
- Python
- Communications Toolbox (Matlab)

4. SOFTWARE OVERVIEW

4.1 MATLAB CODE

NOISE ADDITION:

```
function count = NoiseAdd(snr)

    count = 0 ;

    %directory_in=input('enter directory path of the dataset of each instrument: ','s');
    directory_in = 'D:\Matlab\DSP project\data\';

    %directory_noisy=input('enter directory path for the noisy data of each instrument:
','s');
    directory_noisy = 'D:\Matlab\DSP project\n_data';

    d = dir(directory_in);
    isub = [d(:).isdir];
    nameFolds = {d(isub).name}';
    nameFolds(ismember(nameFolds,{'.','..'})) = [];
```

```
    for i = 1:size(nameFolds)
        nums = [nameFolds{i,:}];
        s = strcat(directory_in,nums);
        count = count+1;
        allFiles = dir(s);
        allNames = {allFiles.name};
        allNames(ismember(allNames,{'.','..'})) = [];

        [rows, columns] = size(allNames);

        for j = 1:columns
            count = count+1;
            audio_path = strcat(s,'\',[allNames{:,j}]);
            [y,Fs] = audioread(audio_path);

            %adding white guassian noise to y

            yy = awgn(y,snr);
            c = strcat(directory_noisy,"\",nums,"\",[allNames{:,j}],".wav");

            %writing a new file

            audiowrite(c,yy,Fs);
        end
    end
end
```

MACHINE LEARNING MODELS USED:

SVM*:*

```
function precision = SVM()
    [train,test,res] = createtbl();
    mdl = fitcecoc(train,res);
    res_test = predict(mdl,test);
    precision = PRECISION(res,res_test);
end
```

RIDGE:

```
function precision = RIDGE()
    [train,test,res] = createtbl();
    b = ridge(table2array(res),table2array(train),5,0);
    res_test = round(b(1) + table2array(test)*b(2:end));
    precision = PRECISION(res,res_test);
end
```

LINEAR REGRESSION:

```
function precision = LR()
    [train,test,res] = createtbl();
    mdl = fitlm([train res]);
    res_test = round(predict(mdl,test));
    precision = PRECISION(res,res_test);
end
```

LASSO:

```
function precision = LASSO()
    [train,test,res] = createtbl();
    [B,FitInfo] = lasso(table2array(train),table2array(res),'Alpha',0.75,'CV',10);
```

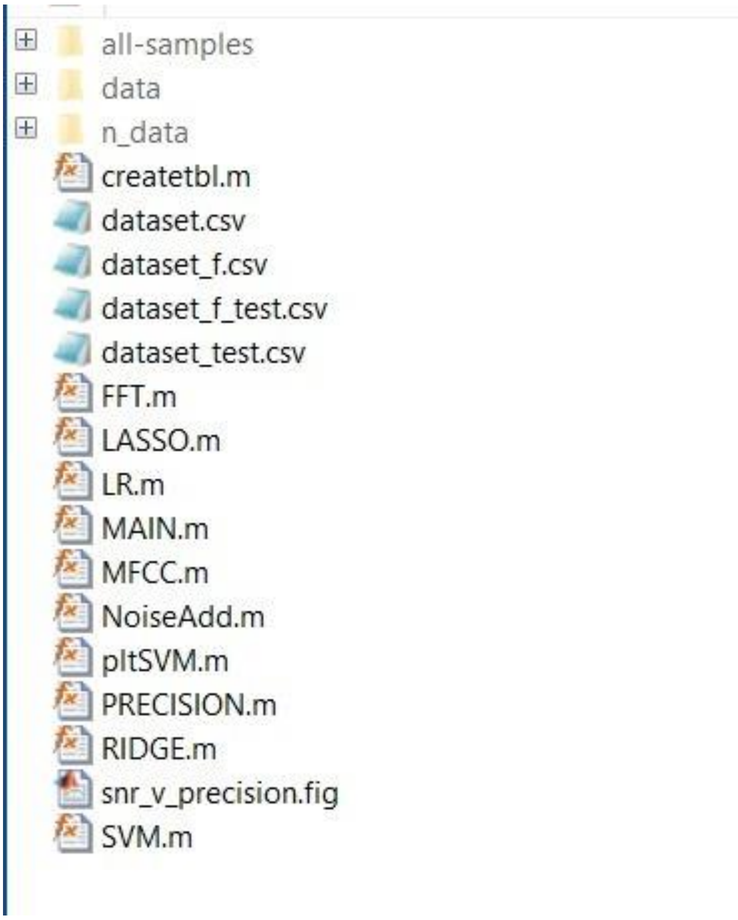
```
    idxLambda1SE = FitInfo.Index1SE;
    coef = B(:,idxLambda1SE);
    coef0 = FitInfo.Intercept(idxLambda1SE);
    XTest = table2array(test);
    yhat = round(XTest*coef + coef0);
    precision = PRECISION(res,yhat);
end
```

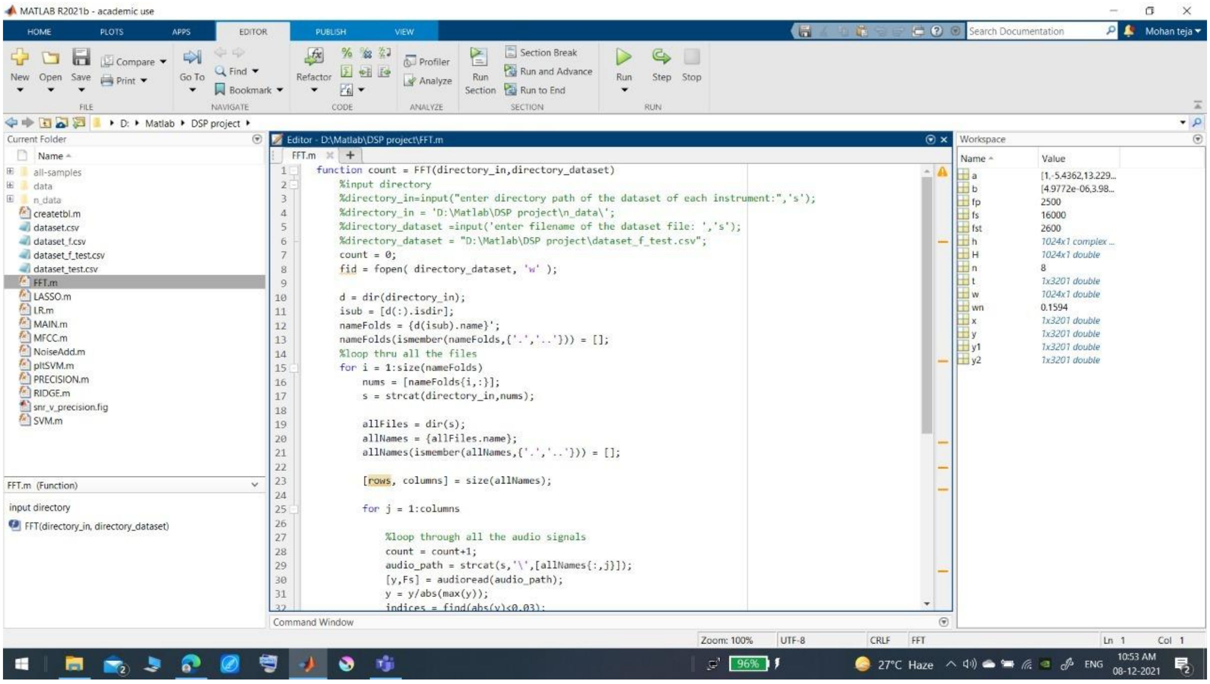
PRECISION CALCULATION:

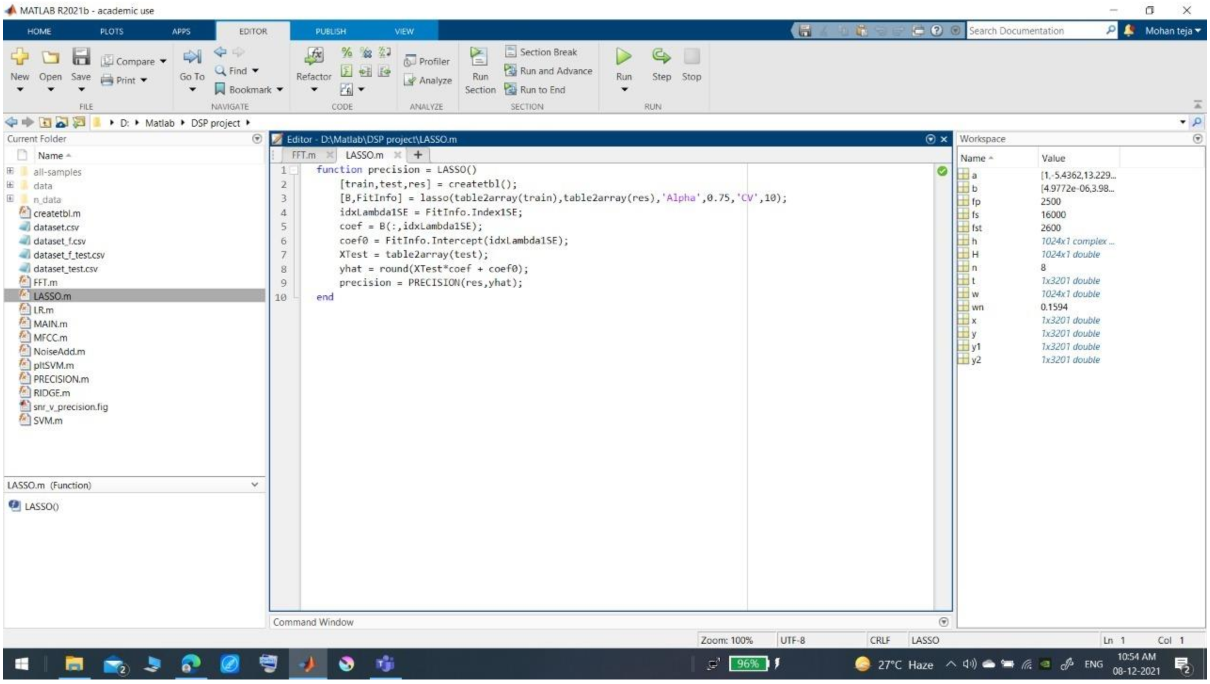
```
function precision = PRECISION(res,res_test)
    count=0;
    same =0;
    for i=1:height(res)
        count= count+1;
        if(res_test(i,:)==res{i,:})
            same= same+1;
        end
    end
    precision = same*100/count;
end
```

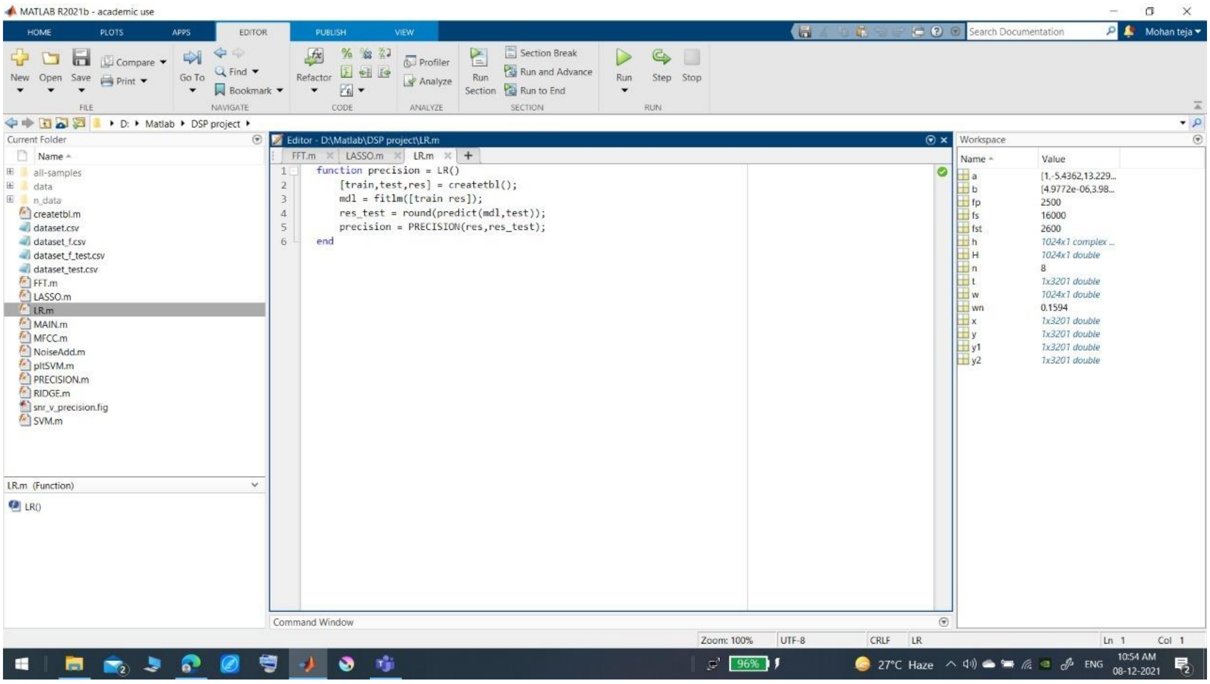
5. RESULTS

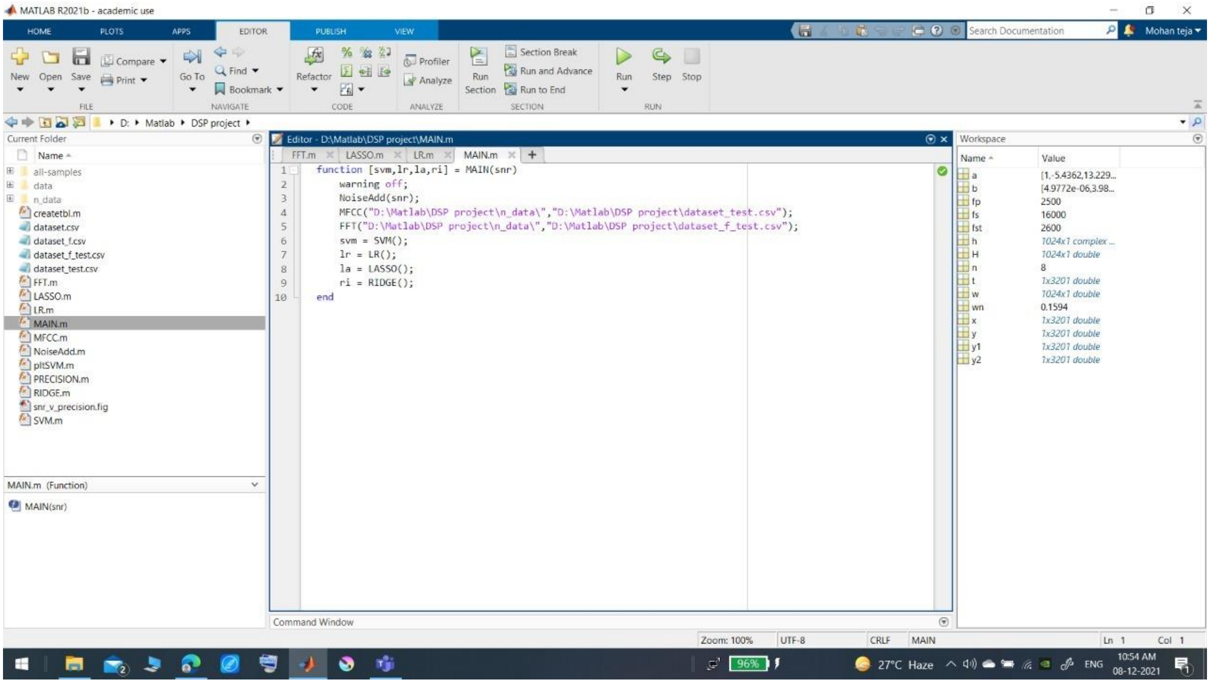
MATLAB SNAPSHOTS:

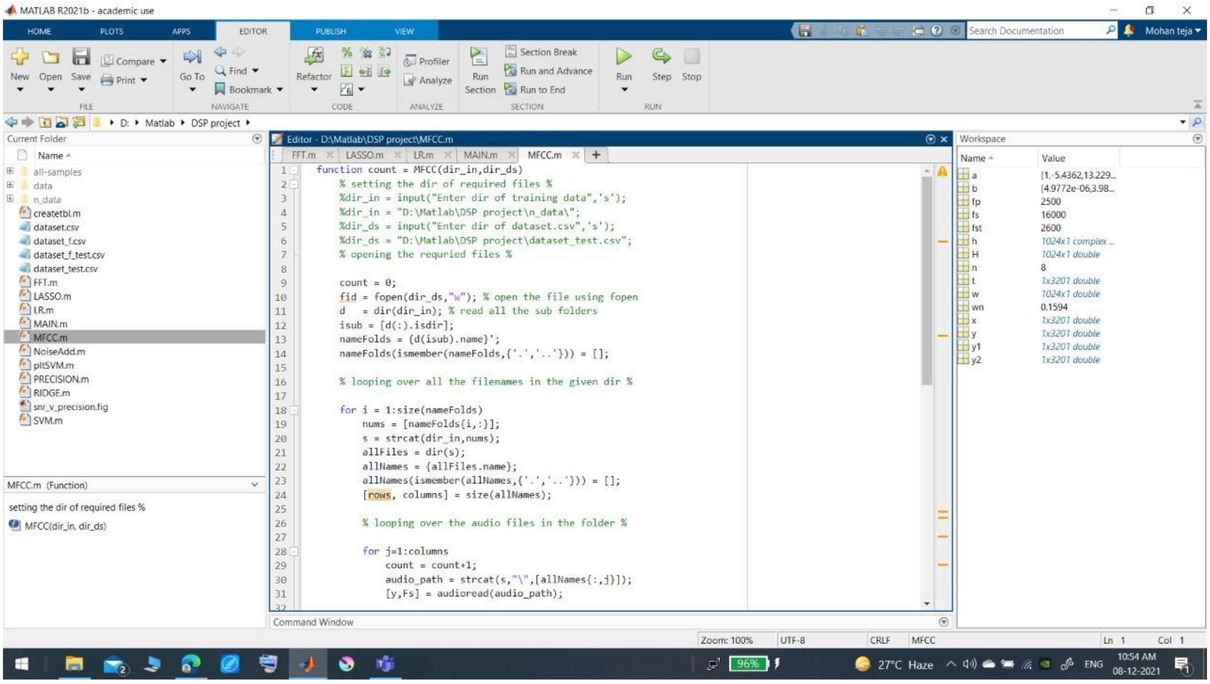


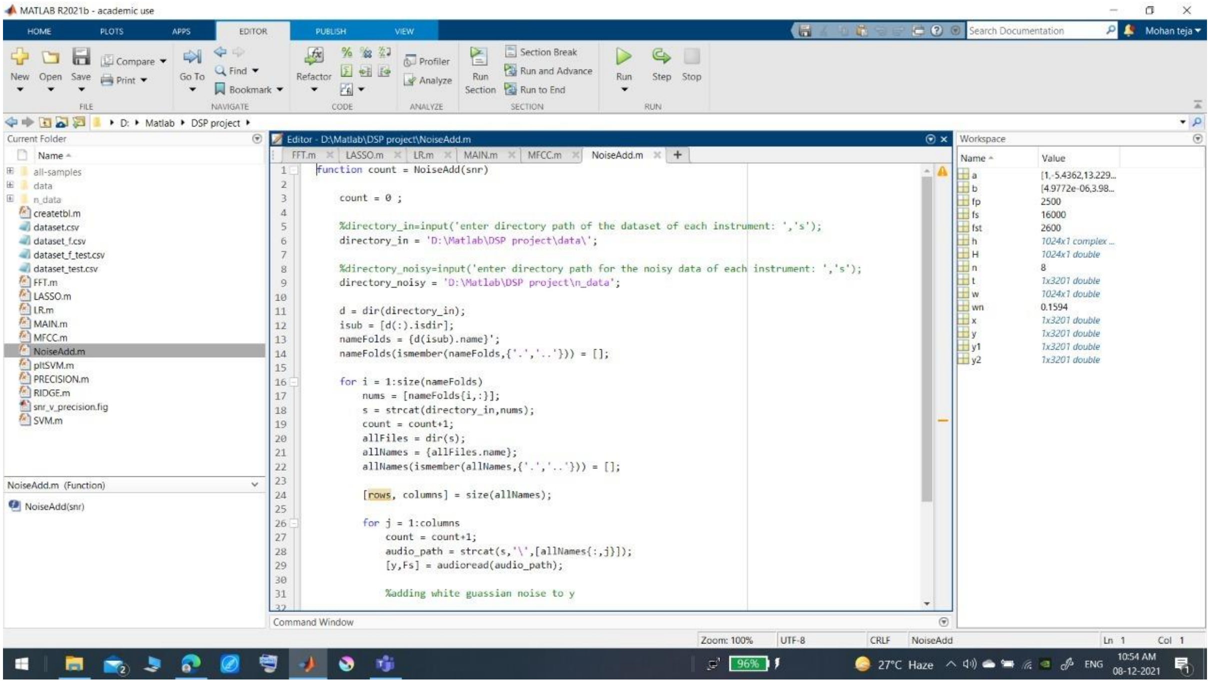


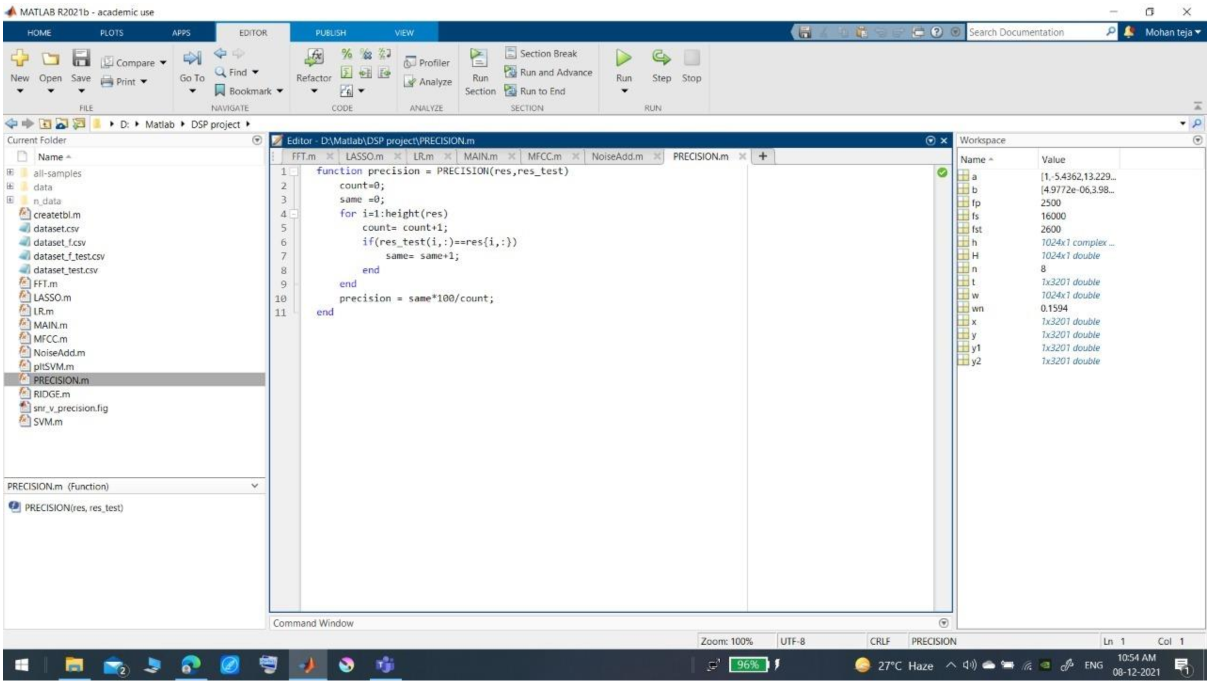

MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\LASSO.m
FFT.m
LASSO.m
function precision = LASSO()
[train,test,res] = createtbl();
[B,FitInfo] = lasso(table2array(train),table2array(res),'Alpha',0.75,'CV',10);
idxLambda1SE = FitInfo.Index1SE;
coef = B(:,idxLambda1SE);
coef0 = FitInfo.Intercept(idxLambda1SE);
XTest = table2array(test);
yhat = round(XTest*coef + coef0);
precision = PRECISION(res,yhat);
end

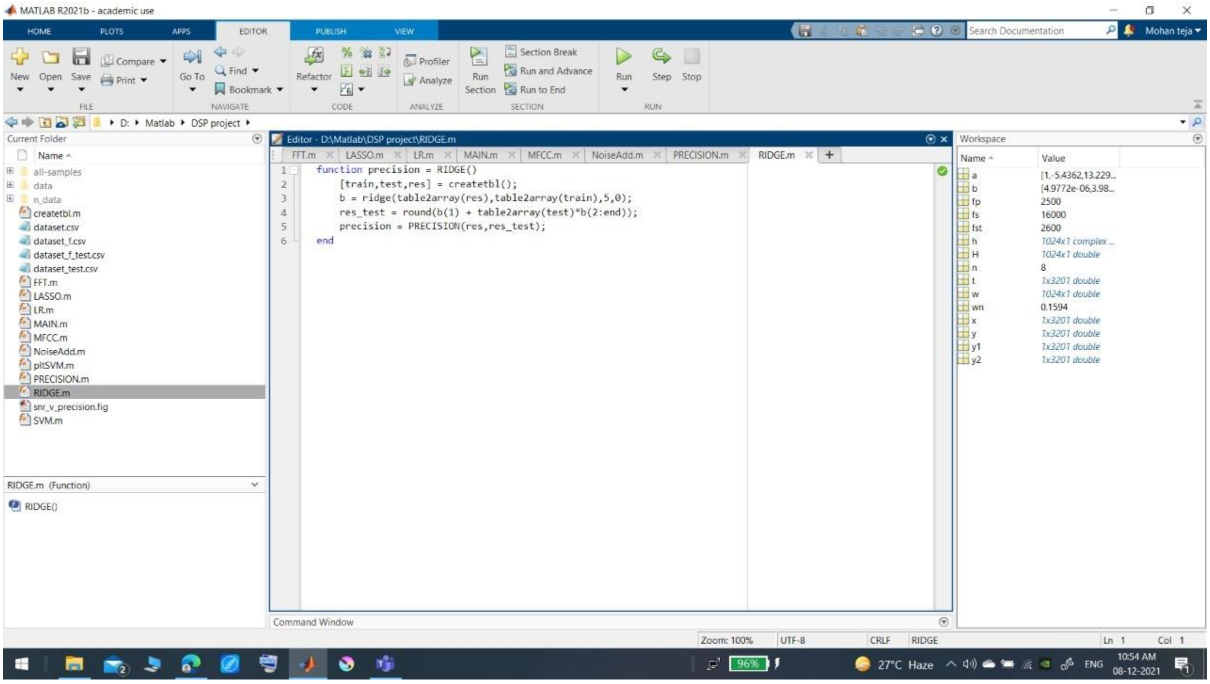

MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\LR.m
FFT.m
LASSO.m
LR.m
function precision = LR()
[train,test,res] = createtbl();
mdl = fitlm([train res]);
res_test = round(predict(mdl,test));
precision = PRECISION(res,res_test);
end

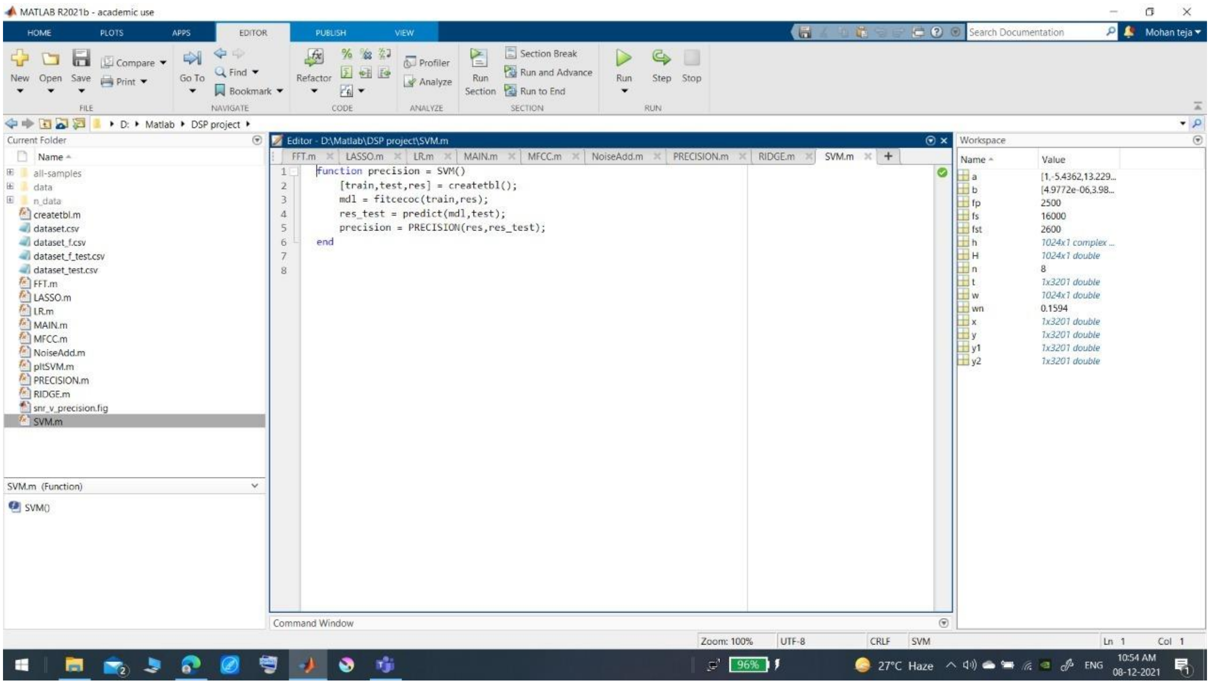

MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\MAIN.m
function [svm,lr,la,ri] = MAIN(snr)
warning off;
NoiseAdd(snr);
MFCC("D:\Matlab\DSP project\n_data\","D:\Matlab\DSP project\dataset_test.csv");
FFT("D:\Matlab\DSP project\n_data\","D:\Matlab\DSP project\dataset_f_test.csv");
svm = SVM();
lr = LR();
la = LASSO();
ri = RIDGE();
end


MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\MFCC.m
function count = MFCC(dir_in,dir_ds)
% setting the dir of required files %
%dir_in = input("Enter dir of training data",'s');
%dir_in = "D:\Matlab\DSP project\n_data\";
%dir_ds = input("Enter dir of dataset.csv",'s');
%dir_ds = "D:\Matlab\DSP project\dataset_test.csv";
% opening the requried files %
count = 0;
fid = fopen(dir_ds,"w"); % open the file using fopen
d = dir(dir_in); % read all the sub folders
isub = [d(:).isdir];
nameFolds = {d(isub).name}';
nameFolds(ismember(nameFolds,{'.','..'})) = [];
% looping over all the filenames in the given dir %
for i = 1:size(nameFolds)
nums = [nameFolds{i,:}];
s = strcat(dir_in,nums);
allFiles = dir(s);
allNames = {allFiles.name};
allNames(ismember(allNames,{'.','..'})) = [];
[rows, columns] = size(allNames);
% looping over the audio files in the folder %
for j=1:columns
count = count+1;
audio_path = strcat(s,"\",[allNames{:,j}]);
[y,Fs] = audioread(audio_path);


Editor - D:\Matlab\DSP project\NoiseAdd.m
function count = NoiseAdd(snr)
count = 0 ;
%directory_in=input('enter directory path of the dataset of each instrument: ','s');
directory_in = 'D:\Matlab\DSP project\data\';
%directory_noisy=input('enter directory path for the noisy data of each instrument: ','s');
directory_noisy = 'D:\Matlab\DSP project\n_data';
d = dir(directory_in);
isub = [d(:).isdir];
nameFolds = {d(isub).name}';
nameFolds(ismember(nameFolds,{'.','..'})) = [];
for i = 1:size(nameFolds)
nums = [nameFolds{i,:}];
s = strcat(directory_in,nums);
count = count+1;
allFiles = dir(s);
allNames = {allFiles.name};
allNames(ismember(allNames,{'.','..'})) = [];
[rows, columns] = size(allNames);
for j = 1:columns
count = count+1;
audio_path = strcat(s,'\',[allNames{:,j}]);
[y,Fs] = audioread(audio_path);
%adding white guassian noise to y


Editor - D:\Matlab\DSP project\PRECISION.m
function precision = PRECISION(res,res_test)
count=0;
same =0;
for i=1:height(res)
count= count+1;
if(res_test(i,:)==res{i,:})
same= same+1;
end
end
precision = same*100/count;
end


MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\RIDGE.m
function precision = RIDGE()
[train,test,res] = createtbl();
b = ridge(table2array(res),table2array(train),5,0);
res_test = round(b(1) + table2array(test)*b(2:end));
precision = PRECISION(res,res_test);
end
Current Folder
Workspace
Command Window

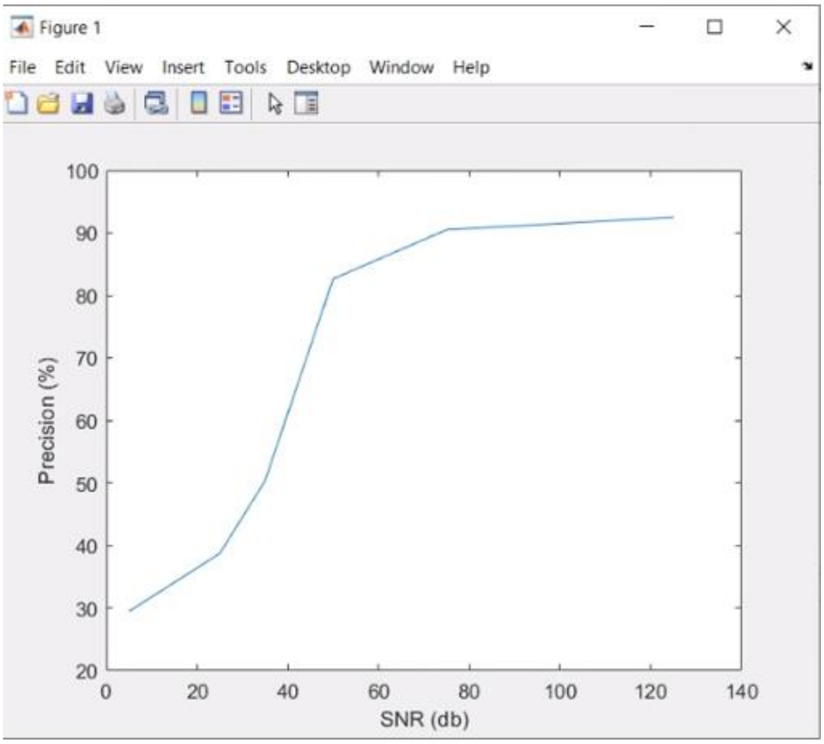

MATLAB R2021b - academic use
Editor - D:\Matlab\DSP project\SVM.m
function precision = SVM()
[train,test,res] = createtbl();
mdl = fitcecoc(train,res);
res_test = predict(mdl,test);
precision = PRECISION(res,res_test);
end
Current Folder
Workspace
Command Window

PRECISION PERCENTAGES:

```
>> [svm,lr,la,ri] = MAIN(100)

svm =

   83.5476


lr =

   96.7609


la =

   95.0643


ri =

   81.1311
```

In comparison, it is observed that maximum precision was achieved using the linear regression model.

PRECISION VALUES WITH RESPECT TO SNR:



6. CONCLUSION AND FUTURE WORK

This paper presents different types of classification schemes used to identify musical instruments. It can be said that these techniques are not the verge of the story of musical instrument recognition, and that there is lot of scope for expansion.

Based on the test, the MFCC and machine learning methods can classify the sound source of the instrument with an accuracy of 83.54%, 96.76%, 95.06% and 81.13%, with linear regression being the most accurate. It can be concluded that MFCC and machine learning models can be implemented in classifying sound sources on musical instruments with good accuracy.

FUTURE WORK:

This project can be further expanded to design and employ models and algorithms that have an optimized performance and work with more accuracy. It can also be expanded to classify and identify instruments from real-time music using the spontaneous data that is being obtained from the same.

Music content analysis in general has many practical applications, including structured coding, automatic musical signal annotation, and musicians' tools. Automatic musical instrument recognition is a crucial subtask in solving these difficult problems, and may also provide useful information in other sound source recognition areas, such as speaker recognition and much more.

7. REFERENCES

[1] M. S. Nagawade and V. R. Ratnaparkhe, "Musical instrument identification using MFCC," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 2198-2202, doi: 10.1109/RTEICT.2017.8256990.

[2] Toghiani-Rizi, Babak and Marcus Windmark. "Musical Instrument Recognition Using Their Distinctive Characteristics in Artificial Neural Networks." ArXiv abs/1705.04971 (2017): n. pag.

[2] http://cs229.stanford.edu/proj2015/010_report.pdf

[3] Tetsuro, Kitahara & Goto, Masataka & Komatani, Kazunori & Ogata, Tetsuya & Okuno, Hiroshi. (2007). Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps. EURASIP Journal on Advances in Signal Processing. 2007. 10.1155/2007/51979.

[4] Saini, Manish & Bhargava, Vipin. (2013). Signal Processing Techniques for Musical Instrument Recognition.

[5] Mazarakis, Giorgos & Tzevelekos, Panagiotis & Kouroupetroglou, Georgios. (2006). Musical Instrument Recognition and Classification Using Time Encoded Signal Processing and Fast Artificial Neural Networks. 246-255. 10.1007/11752912_26.

[6] Patel, Jay and E. S. Gopi. "Musical Notes Identification using Digital Signal Processing." Procedia Computer Science 57 (2015): 876-884.

8. BIODATA

1. NAME : G. MOHAN TEJA
   MOBILE : +91 6303710025
   EMAIL ADDRESS: mohanteja.g2019@vitstudent.ac.in

2. NAME : LAVANYA SINGH
   MOBILE : +91 9818299222
   EMAIL ADDRESS: lavanya.singh2019@vitstudent.ac.in

3. NAME : THEJASVINI.S
   MOBILE : +91 7904382573
   EMAIL ADDRESS: thejasvini.s2019@vitstudent.ac.in

4. NAME : AKSHAJ PRASAD
   MOBILE : +91 9868185283
   EMAIL ADDRESS: akshaj.prasad2019@vitstudent.ac.in