Music Instrument Recognition

TEAM 49

Arushi Singhal (201516178)

Avani Agrawal (201530194)

Simran Singhal (201516190)

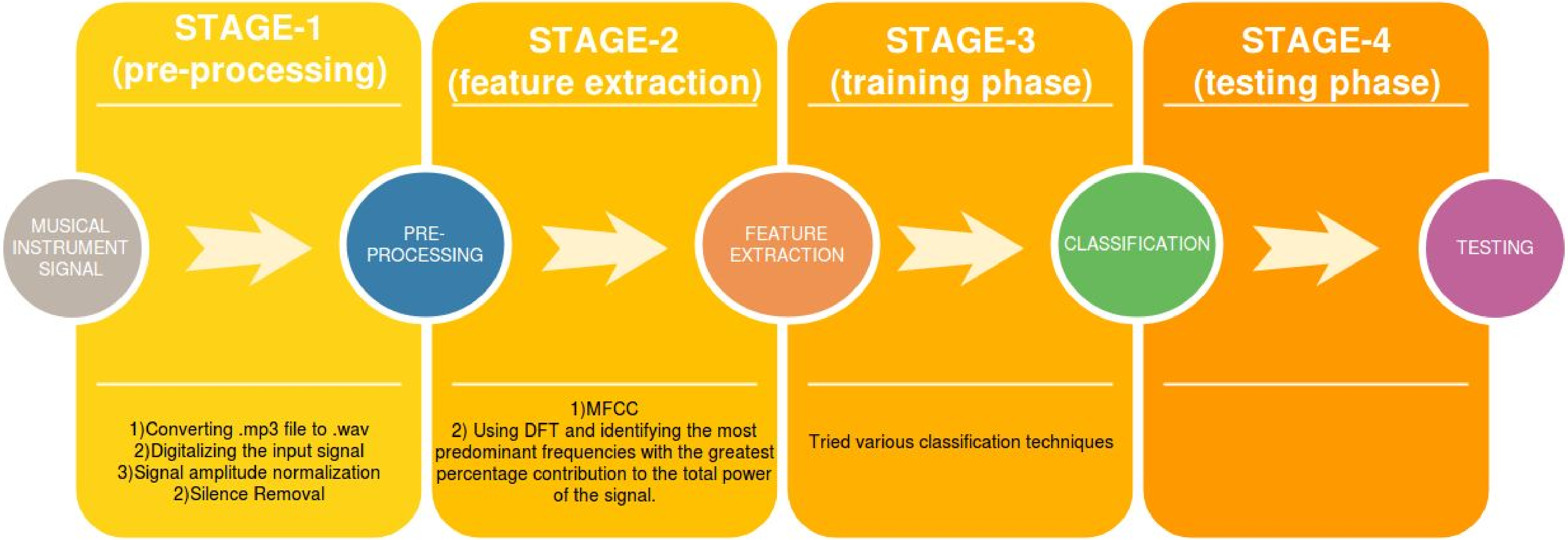
INTRODUCTION

- Musical instruments come in wide spectrum of shapes and size and the characteristics of ones sound can just as well be distinct or similar to some other instruments.
- Same type of instrument can give different sound based on the material from which it is constructed and even on the style used by different musicians to play the instrument.
- In this project we are using machine learning techniques to compare different characteristics of musical instruments and study their ability to distinguish a range of instruments. This was performed using the frequency spectrum of the audio signal, together with some classification methods.

PROBLEM STATEMENT

- Main objective of our project is **“MUSICAL INSTRUMENT RECOGNITION”** using the concepts of speech signal processing in frequency domain and feature classification using machine learning techniques.
- We are aiming at classifying **monophonic** and **monotimbral** signals.
- Some of the issues faced during recognition are the presence of noise and silence in the audio and challenge were to remove them.

PROJECT PIPELINE



DATASET

(http://www.philharmonia.co.uk/explore/sound_samples/banjo)

Our dataset comprises of 7 musical instruments

- Cello (889 samples)
- Clarinet (846 samples)
- Flute (878 samples)
- Guitar (106 samples)
- Saxophone (732 samples)
- Trumpet (485 samples)
- Violin (1502 samples)

Languages And Toolkit

- Bash scripting
- MATLAB
- Python
- Scikit learn (python library)

STAGE - 1

DATA PREPROCESSING

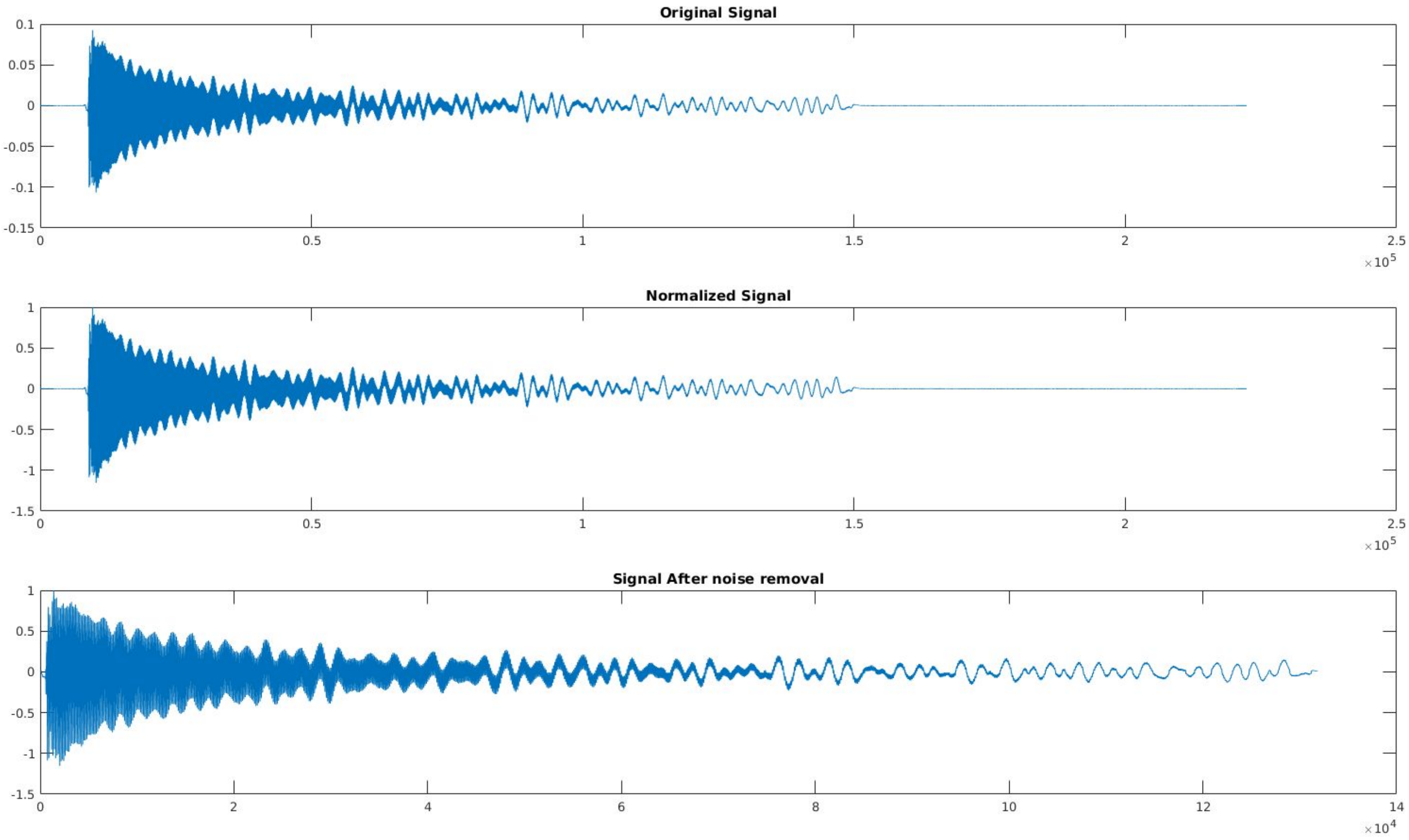
- Converting .mp3 to .wav
- Digitizing the input signal
- Silence removal
- Signal amplitude normalization

Input file conversion from .mp3 to .wav format

- Used bash scripting for file format conversion.
- .mp3 files are lossy and compressed files
- .wav files are lossless and uncompressed files.

Sampling .wav file data (converting continuous data to discrete) and further processing

- Used MATLAB for sampling the signal.
- Amplitude normalization
- Silence removal by setting an amplitude threshold.

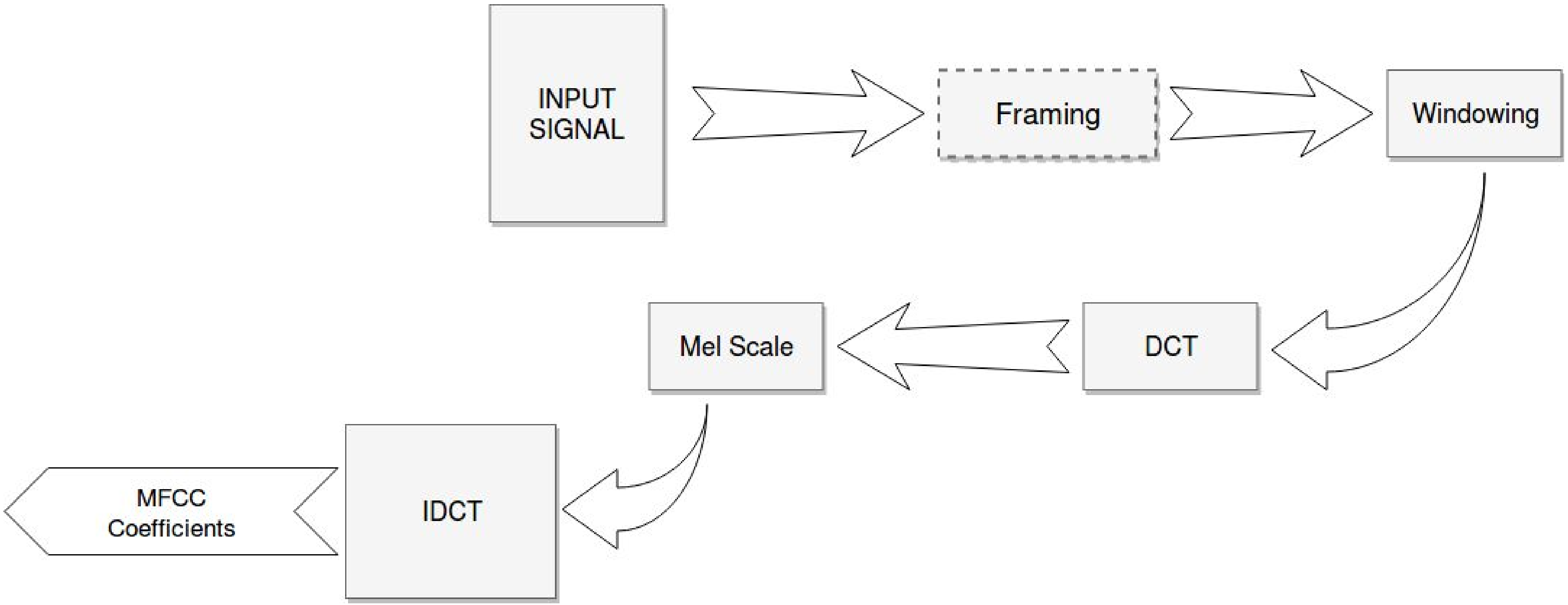


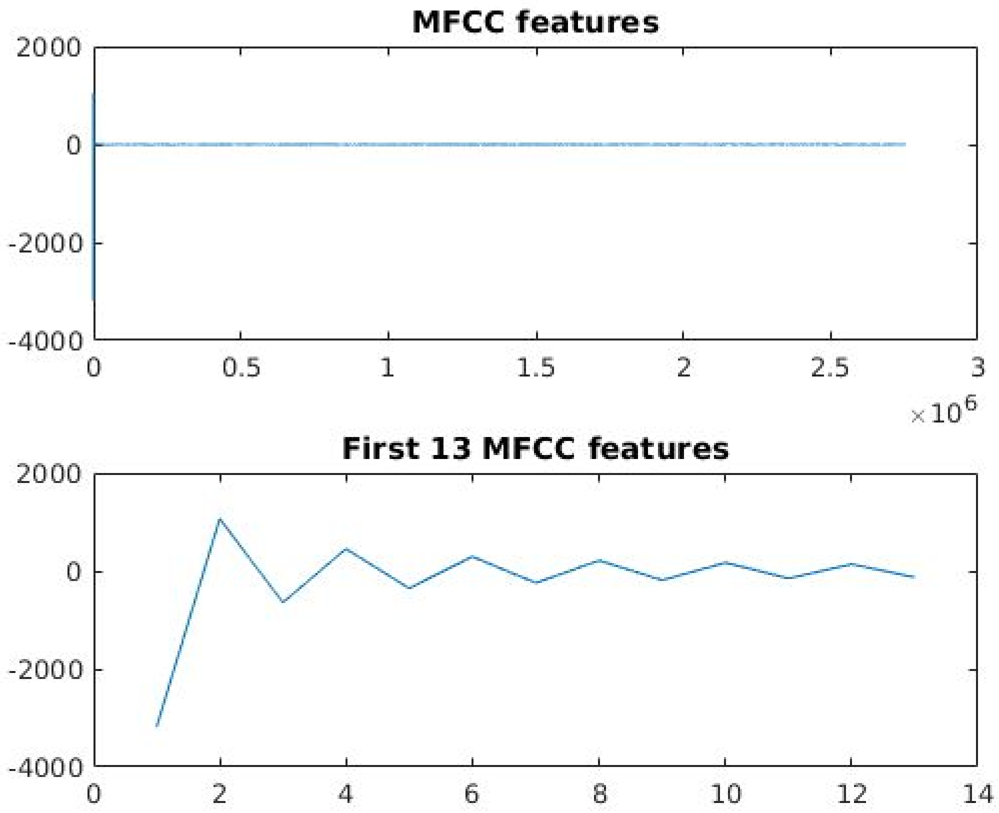
Original Signal
0.1
0.05
0
-0.05
-0.1
-0.15
0
0.5
1
1.5
2
2.5
×10^5
Normalized Signal
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
2
2.5
×10^5
Signal After noise removal
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
14
×10^4

Stage - 2
(features extraction)

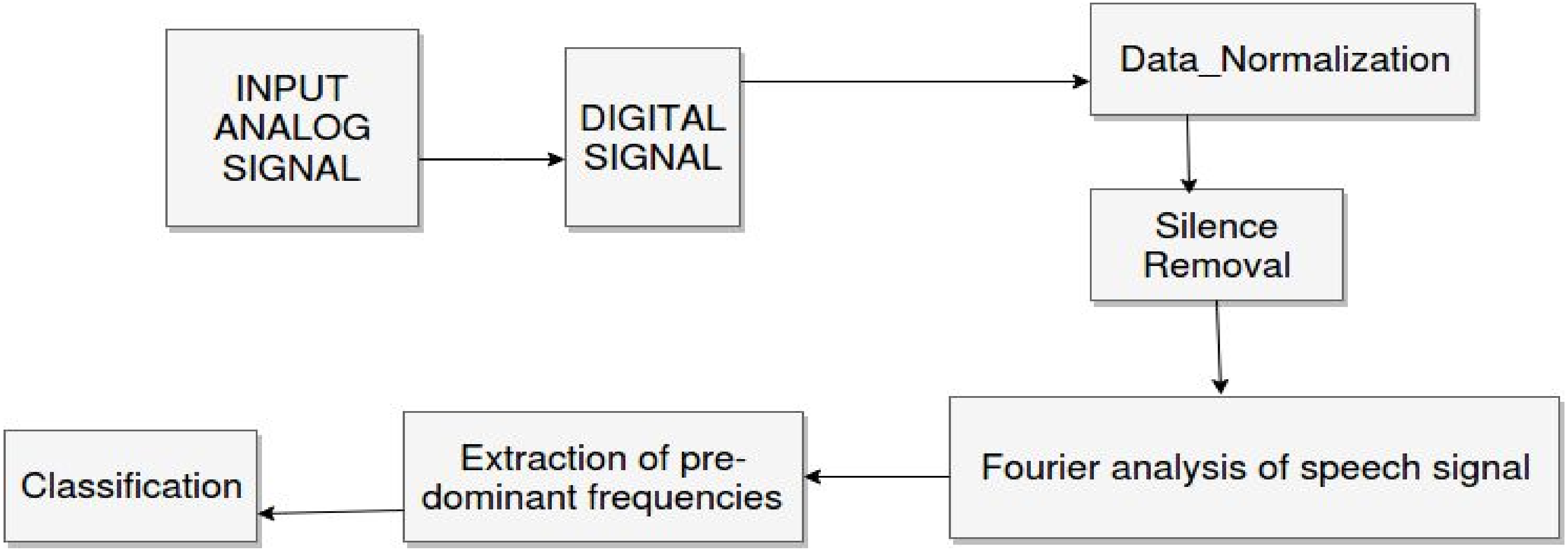
- Mel Frequency Cepstral Coefficients (MFCC)
- Using DFT and identifying the most predominant frequencies contributing greatest to the total signal power.

MFCC


INPUT SIGNAL
Framing
Windowing
DCT
Mel Scale
IDCT
MFCC Coefficients


MFCC features
2000
0
-2000
-4000
0
0.5
1
1.5
2
2.5
3
$\times 10^6$
First 13 MFCC features
2000
0
-2000
-4000
0
2
4
6
8
10
12
14

Feature extraction using DFT



Stage - 3

Classification using :-

- SVM (using sigmoid and RBF kernels)

Result of SVM for different kernels for MFCC without Noise

kernel='rbf' C=50.0 decision_function_shape='ovr' gamma=0.005 Accuracy:- 0.993235294118 Precision:- 0.993326159933 Recall:- 0.993235294118 F1-score:- 0.993280723962	kernel='sigmoid' C=0.1000 decision_function_shape='ovr' gamma=0.001 Accuracy:- 0.276470588235 Precision:- 0.0764359861592 Recall:- 0.276470588235 F1-score:- 0.119761452968

Result of SVM for different kernels for MFCC Noise

kernel='rbf' C=50.0 decision_function_shape='ovr' gamma=0.005 Accuracy:- 0.993235294118 Precision:- 0.993326159933 Recall:- 0.993235294118 F1-score:- 0.993280723962	kernel='sigmoid' C=0.1000 decision_function_shape='ovr' gamma=0.001 Accuracy:- 0.276470588235 Precision:- 0.0764359861592 Recall:- 0.276470588235 F1-score:- 0.119761452968

Stage - 4

TESTING PHASE

CONCLUSIONS

After analyzing different classification techniques on the given data we got highest 99% accuracy with SVM by using kernel ‘rbf’.

REFERENCES

- http://cs229.stanford.edu/proj2015/010_report.pdf
- https://arxiv.org/pdf/1705.04971.pdf
- http://aircconline.com/sipij/V4N4/4413sipij08.pdf