IT204 SIGNALS AND SYSTEMS MINI PROJECT
MUSICAL INSTRUMENT DETECTOR

By:
Lucy Kumari 211IT037
Shreya Suresh Jindrali 211IT064
Shikha Reji 211IT061

INDEX:

INTRODUCTION

Knowledge of the instrumentation of a musical signal at any given time could be useful for major audio signal processing problems such as sound source separation and automated music transcription. Knowing which instruments are playing is a first step toward more intelligently designed solutions to these very important and largely unsolved challenges.

Instrument classification not only has the potential to benefit musical instrument learners, but also could have more widespread implications towards widespread organisation of musical audio clips based on instruments.

DATASET

The dataset is taken from
https://philharmonia.co.uk/resources/sound-samples/
The samples are 1 second in length. We have taken 500 audio samples in which there are 50 audio samples each for the musical instruments banjo, bass, cello, clarinet, flute, guitar, mandolin, saxophone, trumpet, violin.

We have divided the dataset into training and testing dataset. 25% of the dataset is the data for testing and the remaining 75% of the data is for training. The training dataset will be used to train the machine learning model whereas the testing data is used to check the accuracy of the model.

```
train_set shape: (375, 13)
test_set shape: (125, 13)
train_classes shape: (375,)
test_classes shape: (125,)
```

METHODOLOGY

We have prepared the labels from filenames of the audio samples and it is encoded to convert it into the machine readable form using the label encoder.
The sampling frequency is taken as 44100.
The length of the FFT window is 2048.

MFCC

MFCC is Mel-frequency cepstral coefficients. These coefficients collectively make up an MFC(Mel-frequency cepstrum). They are derived from a cepstral representation of an audio clip.

The frequency bands are equally spaced in a mel scale which approximates the human auditory system's response more closely than the linearly spaced frequency bands used in the normal spectrum.

This frequency warping can allow for better representation of sound, for example, in audio compression that might potentially reduce the transmission bandwidth and the storage requirements of audio signals.

MFCCs are used extensively in speech and speaker recognition. Essentially, they represent the Discrete Cosine Transform of the log spectrum of a signal analysed on an auditory frequency scale (the Mel scale). The process creates a 13-dimensional vector that summarises the signal’s spectrum

Some of the features calculated by MFCC are;

- ❖ **Channels:** The number of channels.
- ❖ **Sample Width:** The number of bytes per sample.
- ❖ **Frame rate/Sample rate:** The frequency of samples used.
- ❖ **Frame width:** The number of bytes for each frame.
- ❖ **Length:** The length of the audio file in milliseconds.
- ❖ **Frame count:** The number frames from the sample.
- ❖ **Intensity:**The loudness in dbFS.

Calculation of MFCC features:

- ❖ Fourier Transform of the windowed excerpt signal.
- ❖ Apply Discrete Fourier Transform
- ❖ Apply log of the magnitude
- ❖ Warp the frequencies on mel scale
- ❖ Apply inverse Discrete Fourier Transform

STANDARDIZATION

Standardising the data values into standard format.
Zero mean:The values for an attribute are normalised based on the mean.
Unit Variance:It gives the Standard deviation of the sample.

K-NEAREST NEIGHBOUR ALGORITHM

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

EVALUATION

- ❖ **Recall:** It refers to the percentage of total relevant results correctly classified by your algorithm.
- ❖ **F1 Score:** combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

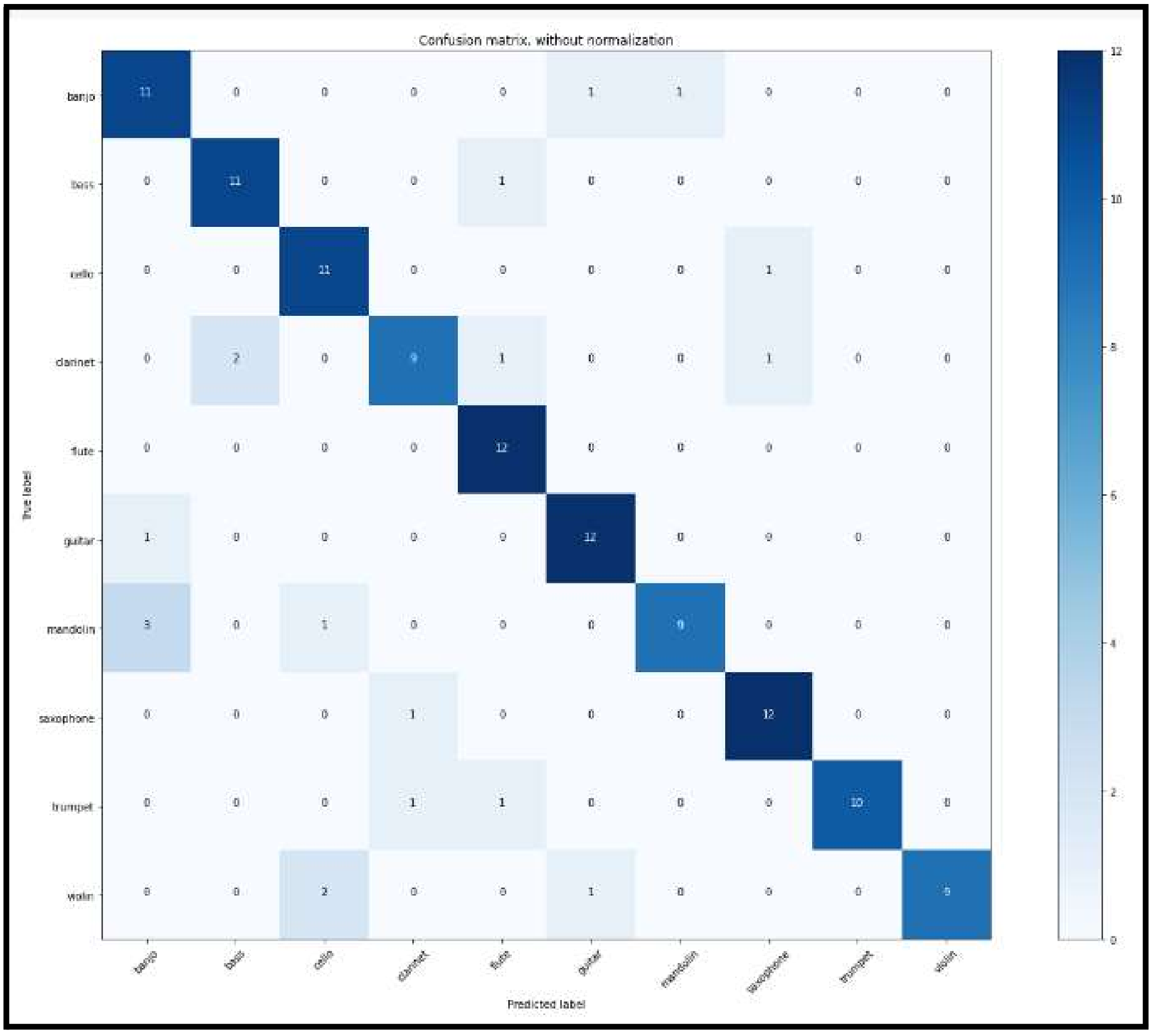
- ❖ **Accuracy:** It is the degree to which the result of a measurement conforms to the correct value or a standard.
- ❖ **Precision:** the quality or state of being precise.

```
Recall:  [0.85 0.92 0.92 0.69 1.   0.92 0.69 0.92 0.83 0.75]
Precision:  [0.73 0.85 0.79 0.82 0.8  0.86 0.9  0.86 1.   1.  ]
F1-Score:  [0.79 0.88 0.85 0.75 0.89 0.89 0.78 0.89 0.91 0.86]
Accuracy: 0.85  , 106
Number of samples: 125
```

CONFUSION MATRIX WITHOUT NORMALISATION

A confusion matrix is a technique for summarising the performance of a classification algorithm.Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

Classification accuracy is the ratio of correct predictions to total predictions made.Classification accuracy can also easily be turned into a misclassification rate or error rate by inverting the value. The number of correct and incorrect predictions are summarised with count values and broken down by each class. This is the key to the confusion matrix.


Confusion matrix, without normalization
True label
Predicted label
banjo
bass
cello
clarinet
flute
guitar
mandolin
saxophone
trumpet
violin

REFERENCES:

- ❖ Greg Sell, Gautham J. Mysore, and Song Hui Chon. Musical Instrument Detection. Thesis. Centre for Computer Research in Music and Acoustics, 2006. N.p.: n.p., n.d. Web. 17 Feb. 2017.
- ❖ Christian Simmermacher, Da Deng, and Stephen Cranefield. Feature Analysis and Classification of Classical Musical Instruments: An Empirical Study. Thesis. Department of Information Science, University of Otago, 2006. N.p.: n.p., n.d. Web. 17 Feb. 2017.
- ❖ Sabin, Manuel. Musical Instrument Recognition With Neural Networks. Thesis. 2013. N.p.: n.p., n.d. Web. 17 Feb. 2017
- ❖ Agostini, Giulio, Maurizio Longari, and Emanuele Pollastri. Content-Based Classification of Musical Instrument Timbres. Thesis. Laboratorio Di Informatica Musicale - L.I.M, n.d. N.p.: n.p., n.d. Web. 17 Feb. 2017.

THANK YOU