International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

An Overview of the use of Python for Audio Signal Processing

Shreshthi SIngh Parihar*

** Student, Dept. Computer Engineering,*
Bharati Vidyapeeth's College of Engineering, Lavale, Pune, Maharashtra, India
Savitribai Phule Pune University

---***---

Abstract -*Audio signal processing is one of the most researched fields of study dating back to the beginning of 20th Century. But, due to the emergence of superfast computers and affordable data storage the digital audio signal processing is the most sought after technology. With organizations trying to optimize the sound quality sent and received via. The internet calls, this field is the need of the hour. Python enables the programmer to do audio signal processing with the numerous libraries it already has.*

Key Words: Python, Audio Signal Processing, Libraries

1. INTRODUCTION

The study of Audio Signal Processing goes well back to the beginning of the 20th Century[1]. But, back then the digital systems that we use extensively today were at mere infancy. So the real use of Audio Signal Processing back in those days was via. Analog Audio Signals and all computations and alterations on it were made using electrical signal modifications. Now that we have devices with high computational powers, another branch of Audio Signal Processing emerged viz. Digital SIgnal Processing.

1.1 Analog Audio Signal Processing

Audio Signals basically are electronic representations of longitudinal sound waves which travel through the air. Wherein the direction of wave propagation is parallel to particle displacement. These motions can easily be stored as information or data and or manipulated using various electronic devices.

There has been extensive research on this field of study which is evident with the inventions of various devices like Telegram, Telephone etc. Most of the research has been on the successful receiving, storing as data, and recreating of the audio waves.

1.2 DIgital SIgnal Processing

Due to the advent of affordable, better accessibility and high speed computation devices the researchers have moved onto digital signal processing. Unlike Analog signal processing, digital signal processing works on the digital representation of the audio waves. It uses binary numbers to express the audio waveform in terms of a sequence of symbols. Therefore, one can use the digital circuits like special processors, microprocessors and general-purpose computers. This approach is more powerful than the traditional analog system.

2. Why use Python?

In the last decade python has seen an exponential rise in popularity as is evident from the stackoverflow trends in Fig 1. And considering the massive amount of users on stackoverflow this data can be of importance.

It is owing to the face that Python is (1) Beginner friendly, (2) Versatile - as there are various packages for every purpose one can think of, (3) Open Source (4) Requires less lines of code (loc) for even a complex problem (5) User friendly (6) High productivity.

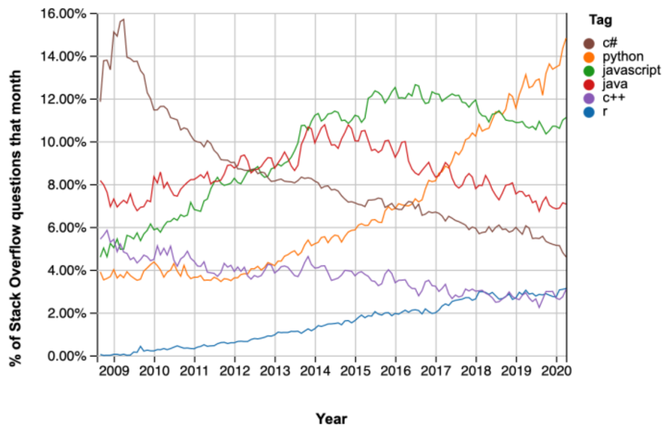


Fig. 1

Python, owing to its open source nature and highly active community, supports numerous packages for any purpose under the sun. Some of the most important packages and libraries under the python are listed below:

2.1 Numpy

NumPy, created in 2005 by Travis Oliphant, is the fundamental package for scientific computing with Python. NumPy, which stands for Numerical Python, provides an array object that is upto fifty times faster than traditional lists. NumPy is majorly used for it as an efficient multi-dimensional container of generic data. Allowing NumPy to seamlessly and speedily integrate with a wide variety of databases. It is licensed under the BSD license, enabling reuse with few restrictions.

2.2 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib tries to make easy things easy and hard things possible.

You can generate plots, histograms, power spectra, bar charts, error-charts, scatterplots, etc., with just a few lines of code.

2.3 Scipy

SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that it should work with NumPy arrays.

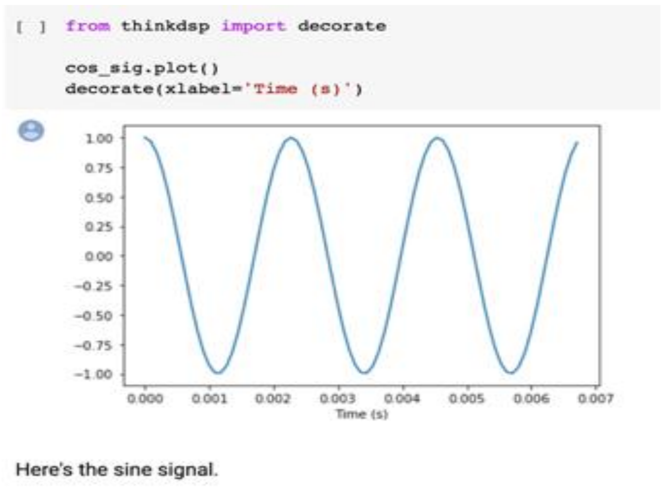
The combination of Numpy, Matplotlib and Scipy work as a replacement for MatLab, which is a popular platform for scientific computing. It easily provides many user friendly and efficient numerical practices such as routines for numerical integration and optimisation. For simple plotting the pyplot module provides a MatLab-like interface. You also have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MatLab users.

2.4 ThinkDSP

ThinkDSP is a python module, built by Allen Downey, that provides classes and functions for working with signals. It simplifies the task of audio signal processing by providing the most used algorithms as functions and is pretty easy to use.

Allen Downey has provided a complete how to use guide with his book named as 'ThinkDSP' which provides a good insight into the use of the various DSP algorithms, which are beginner friendly. A small code snippet from the ThinkDSP tutorial is shown in Fig. 2

This library uses Numpy, Scipy and MatplotLib extensively in the background to provide easy to use functions and algorithms like FFT and Silencer to apply proper modifications to the audio file, preferably in .wav format.



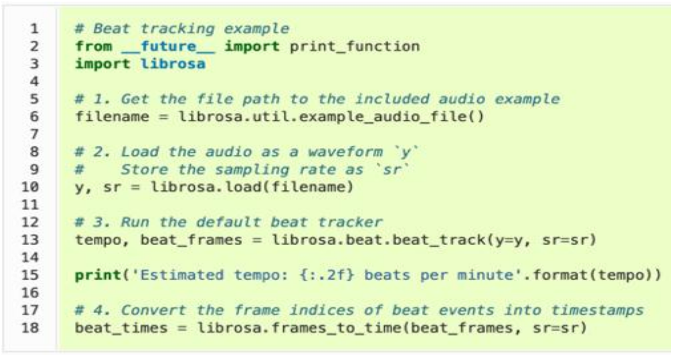
```
[ ]  from thinkdsp import decorate

     cos_sig.plot()
     decorate(xlabel='Time (s)')
```

Here's the sine signal.

Fig. 2

2.5 LibROSA

LibROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. This package provides various submodules that can be used for direct application as audio signal processing. A small code snippet from the libROSA tutorial available at their official website is shown in Fig. 3



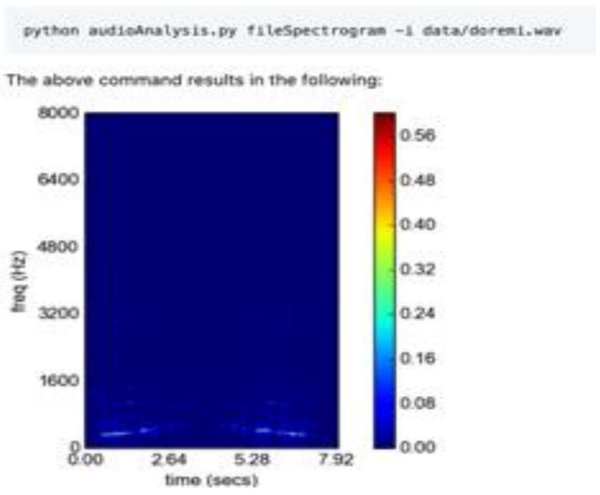
```
# Beat tracking example
from __future__ import print_function
import librosa

# 1. Get the file path to the included audio example
filename = librosa.util.example_audio_file()

# 2. Load the audio as a waveform `y`
#    Store the sampling rate as `sr`
y, sr = librosa.load(filename)

# 3. Run the default beat tracker
tempo, beat_frames = librosa.beat.beat_track(y=y, sr=sr)

print('Estimated tempo: {:.2f} beats per minute'.format(tempo))

# 4. Convert the frame indices of beat events into timestamps
beat_times = librosa.frames_to_time(beat_frames, sr=sr)
```

Fig. 3

2.6 pyAudioAnalysis

pyAudioAnalysis is an open Python library that provides a wide range of audio-related functionalities focusing on feature extraction, classification, segmentation and visualisation issues.

It's dependencies are numpy, Matplotlib, Scipy, Sklearn, hmmlearn, simplejson, eyeD3, pydub and although most python systems will have some of them others will have to be manually installed using the pip commands on the system. It provides the user with audio signal processing functionalities like feature, classification and regression, segmentation, data-visualisation, audio-recording, etc.



```
python audioAnalysis.py fileSpectrogram -i data/doremi.wav
```

The above command results in the following:

Fig. 4

3. CONCLUSION

To conclude, Python is the most popular language for DIgital Audio Signal Processing and it provides a huge amount of tools for the implementation of even very extremely complex problems. Also, due to its special features, solving problems can become more productive and easier as the amount of lines of code decreases very considerably, reducing the development time. The huge open source community of Python is very highly active and most problems are generally already solved on the internet. With this in mind it will be a long time, in my opinion, until python is overthrown from the title of 'most popular language' and the most suitable language for data analysis.

ACKNOWLEDGEMENT

I acknowledge the Head of Department (Computer Engineering) of my college - Prof. Uday Patkar and my Project Guide Prof. Prajakta Pawar for their constant support and guidance.

REFERENCES

[1] Audio signal processing. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Audio_signal_processing

[2] Stack Overflow. (n.d.). Retrieved from https://insights.stackoverflow.com/trends?tags=r,python,javascript,java,c++,c#

[3] Wilmering, Thomas; Moffat, David; Milo, Alessia; Sandler, Mark B. (2020). "A History of Audio Effects". Applied Sciences.

[4] McFee, Brian, et al. "librosa: Audio and music signal analysis in python." Proceedings of the 14th python in science conference. Vol. 8. 2015.

[5] Glover, John C., Victor Lazzarini, and Joseph Timoney. "Python for audio signal processing." (2011).

[6] Giannakopoulos, Theodoros. "pyaudioanalysis: An open-source python library for audio signal analysis." PloS one 10.12 (2015).

[7] Downey, Allen B. Think DSP: digital signal processing in Python. "O'Reilly Media, Inc.", 2016.

[8] De Pra, Yuri, Federico Fontana, and Michele Simonato. "Development of real-time audio applications using python." (2018).