TimeSide API as an audio processing web service

Guillaume Pellerin
IRCAM, Paris
pellerin@ircam.fr

Paul Brossier
Aubio
piem@piem.org

Martin Desrumaux
IRCAM, Paris (until 2022)
martin@letik.io

Antoine Grandry
IRCAM, Paris (until 2020)
antoine.grandry@hotmail.fr

ABSTRACT

Audio descriptors can help to analyze, classify and compare sounds by their own characteristics. For large datasets, it is often needed to store the results of the analyses in order to keep everything sustainable and comparable, especially in machine learning usecases. This demo will show how to use the TimeSide REST API as a remote service to process descriptors in order to embed the results in any web application. The new version of the TimeSide player will be presented as a general example of using the dedicated javascript SDK to produce new kinds of applications involving visualization and collaborative annotation.

1. INTRODUCTION

As the number of online audio applications and datasets increase, it becomes crucial for researchers and engineers to be able to prototype and test their own algorithms as fast as possible on various platforms and usecases like computational musicology[13] and streaming services. On the other side, content providers and producers need to enhance user experiences on their platforms with more metadata based on cultural history but also audio feature analyses. Growing those metadata synchronously with the music published on the internet implies that the analysis and storage systems can be easily updated, scaled and deployed.

TimeSide[8] has been developed in this sense to propose an online audio processing service. It provides a REST API as well as a javascript SDK so that web developers can easily embed the service into their own applications without requesting to much local resources on the user side to compute audio features[11][12].

We will show in this demo how to use the SDK and consume the API in various web oriented contexts. As a first example, the new TimeSide Player prototype will be presented.

2. TIMESIDE CORE FRAMEWORK

TimeSide is first a python framework enabling low and high level audio analysis, imaging, transcoding, streaming and labelling. Its high-level API is designed to enable complex processing on very large datasets of any audio or video assets with a plug-in architecture, a secure scalable backend and an extensible dynamic web frontend. Some usecases: scaled audio computing (filtering, machine learning, etc), web audio visualization, audio process prototyping, realtime and on-demand transcoding and streaming over the web, automatic segmentation and labelling synchronized with audio events

Because there are a lot of tools available in the Python ecosystem dedicated to music information retrieval, machine learning and data analysis, we have decided to embed all main ones: Aubio[10], Yaafe[14], Essentia[5], VAMP[9], librosa[3], GStreamer, TensorFlow, Torch, PyTorch, scikit-learn, Jupyter, Pandas and Pytables. They are used to develop native TimeSide plugins though its simple processing API.

3. TIMESIDE REST API

In addition to be usable as a library, TimeSide has furthermore been built into a Django server with a relational PostgreSQL database in order to store music tracks and processing results. Data structure and relations are defined in the models to ensure easy data serialization. The backend is built with Django REST Framework[2] to provide a documented RESTful API[6]. It guarantees interoperability by allowing other servers or multiple frontends to interact with the TimeSide instance. Any application consuming the API is then able to: upload audio track or retrieve them from remote providers, stream original or transcoded sources, run on-demand analysis with customized parameters, deliver and share several types of results: transcoded audio, numerical or graphical outputs of analysis, collect tags and indices on tracks to build annotated audio corpora for further machine learning purpose[15].

4. TIMESIDE JAVASCRIPT SDK

A (SDK) is also available to help . An example of its use - an augmented HTML5 player - is discussed in next section.

In order to build frontends on top of this web API, a Software Development Kit (SDK) is available[7] for the Javascript and Typescript languages. It has been created from the routes of the OpenAPI schema automatically exported from the backend thanks to the OpenAPI Generator[4]. The SDK proposes some examples for clients to reach the server and request some processing.

Web Audio Conference WAC-2022, December 6–8, 2022, Cannes, France.

5. TIMESIDE PLAYER

The previous versions of the player was consuming the data processed and rendered by the backend only as bitmap images. This would avoid to zoom in very little scales[13]. As the REST API gives now the possibility to stream the processing results directly from the database into the browser as vectorized data, we have rebuild the entire interface to display the features through a dynamic multi-track interface based on D3.js[1] and VueJS. An annotation system also allows to describe an audio event between two times on each track so that users can show and share their analyses in a collaborative way. This open a lot of new usecases between sound and music databases and citizen science protocols.

Figure 1: Overview of the Timeside-Player interface picturing timbre and rhythm analyses for the Queen song "Under Pressure". It shows (a) the audio player and various vectorized audio descriptors (Spectral centroid, Spectral kurtosis, Spectral slope, Onsets, Linear Spectrogram and Waveform) as well as the time annotation system.

6. DEMONSTRATION

We will show how to use the player as well as the SDK to interact with the web service requesting YouTube and Deezer track analyses.

7. TECHNICAL REQUIREMENTS

Only a computer, brought by the demonstrator, and a WiFi internet access are needed for this demonstration.

8. ACKNOWLEDGMENTS

This work was supported by the French Research National Agency (ANR) and the WASABI team (contract ANR-16-CE23-0017-01).

9. REFERENCES

[1] D3.js. https://d3js.org.

[2] Django rest framework. https://www.django-rest-framework.org.

[3] Librosa is a python package for music and audio analysis. https://librosa.github.io/librosa/.

[4] Open api generator. https://github.com/OpenAPITools/openapi-generator.

[5] Open-source library and tools for audio and music analysis, description and synthesis. http://essentia.upf.edu/documentation/.

[6] Timeside api documentation. https://timeside.ircam.fr/timeside/api/docs/.

[7] Timeside api sdk / client. https://github.com/Ircam-Web/timeside-sdk-js.

[8] Timeside repository. https://github.com/Parisson/TimeSide.

[9] The Vamp audio analysis plugin system. http://www.vamp-plugins.org.

[10] aubio, a library for audio labelling. https://aubio.org/.

[11] M. Buffa, E. Cabrio, M. Fell, F. Gandon, A. Giboin, R. Hennequin, F. Michel, J. Pauwels, G. Pellerin, M. Tikat, et al. The wasabi dataset: Cultural, lyrics and audio analysis metadata about 2 million popular commercially released songs. In *European Semantic Web Conference*, pages 515–531. Springer, 2021.

[12] M. Buffa, J. Lebrun, J. Pauwels, and G. Pellerin. A 2 million commercial song interactive navigator. In *WAC - 5th WebAudio Conference*, Dec. 2019.

[13] T. Fillon and G. Pellerin. A collaborative web platform for sound archives management and analysis. In *Proceedings of 3rd Web Audio Conference, London*, page 43. Queen Mary University of London, August 2017.

[14] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard. Yaafe, an easy to use and efficient audio feature extraction software. In *Proc. of ISMIR 2010, Utrecht, Netherlands*, pages 441–446. International Society for Music Information Retrieval, 2010.

[15] G. Peeters and K. Fort. Towards a (better) definition of annotated mir corpora. 2012.