essentia.JS

A JavaScript library for Music and Audio Analysis on the web

Albin Correya[1] **Dmitry Bogdanov**[1] **Luis Joglar-Ongay**[1,2] **Xavier Serra**[1]
Music Technology Group, Universitat Pompeu Fabra[1]
SonoSuite[2]

ISMIR - 2020, Paper 260

https://essentia.upf.edu/essentiajs

Motivation

- JavaScript is a language that can be used on both client and server-side.
- Modern web browsers and web audio technologies became powerful.

Web Audio API
Audio Worklet

- But, we still lack extensive and cross-platform tools/libraries for music and audio analysis on the Web.
- Possible cross-compilation solutions of native MIR libraries are not easy to use for even experienced software developers.
- Writing a library from scratch in JS is labor-intensive and may lack reproducibility required for MIR applications.

ESSENTIA

Essentia is an open-source **C++/Python** library for audio signal processing, developed at the MTG-UPF and licensed under **Affero GPLv3**.

Functionalities
- DSP blocks
- Audio features
  - Spectral features
  - Rhythm and tempo
  - Tonality and melody
  - Loudness
  - Fingerprinting
- Inference with **TensorFlow** models

Design criteria
- **C++** with **Python** bindings
- **Large-scale** deployment
- **Real-time** processing
- **Cross-platform**
  - (Linux, MacOS, Win, iOS, Android, JS)

Actively maintained by a team of developers and researchers

Essentia.js Design Choices and Functionalities

- Easy installation
  - NPM, CDN etc
- User friendly API & utility tools
  - Easy-to-use
- Modularity & Extensibility
  - Easy-to-customize and build
- Web standards compliance
  - Web Audio API, Audio Worklet, ES6 etc.
- Lightweight and lesser dependencies
  - As small as 2.5 MB ~
- Reproducibility across different platforms
  - Native and Web (C++, Python and JS)
- Extensive documentation and examples

Essentia.js Architecture

Web-friendly **MIR JavaScript Library** for both web browsers and Node.js apps.

Features
- Easy-to-install
- Easy-to-use
- Easy-to-extend
- Easy-to-deploy
- Web standards compliance
- Lightweight
- Reproducibility & Cross-platform

Getting Started

npm

```
npm install essentia.js
```

HTML <script> tag CDN

```
<script src="https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia-wasm.web.js"></script>
<script src="https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia.js-core.js"></script>
<script src="https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia.js-plot.js"></script>
<script src="https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia.js-extractor.js"></script>
```

ES6 style imports

```
import Essentia from 'https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia.js-core.es.js';
// import essentia-wasm-module
import { EssentiaWASM } from 'https://cdn.jsdelivr.net/npm/essentia.js@0.1.0/dist/essentia-wasm.module.js';
```

Boilerplate examples

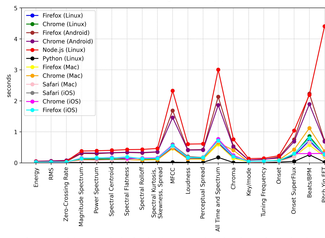
Applications

- Demos for many MIR tasks.
  - Interactive educational materials, listening tests, etc. (to mention a few)
- Machine learning applications using Tensorflow.js / ONNX.js + Essentia.js.
- Web, mobile, IoT apps for music education, identification, analysis etc.
- Creative applications for Creative coding, Games, Graphics, New media etc (to mention a few).

And, much more …

Initial Benchmarks

Mean execution times (s) for Essentia algorithms by platform for a 5 seconds audio



https://mtg.github.io/essentia.js-benchmarks/

Future Work

- Performance improvements
- Pre-trained ML models for audio analysis and classification for web browsers.
- Usage example for various platform and domains
- Support building diverse and inclusive user community

https://essentia.upf.edu/essentiajs

upf. Universitat Pompeu Fabra Barcelona
MTG Music Technology Group
ESSENTIA