2020

Audio Data Capture/Tagging Tools for Machine Learning Apps

Alvaro Eugenio Ardila Perez
Grand Valley State University

Audio Data Capture/Tagging Tools for Machine Learning Apps

Alvaro Eugenio Ardila Perez

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

December 2020

GRAND VALLEY
STATE UNIVERSITY

The signatures of the individuals below indicate that they have read and approved the project of Alvaro Eugenio Ardila Perez in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

Jonathan Engelsma, Project Advisor Date

Robert Adams, Graduate Program Director Date

Paul Leidig, Unit head Date

Contents

Abstract 4
Introduction 5
Problem Statement 6
Background and Related Work 7
Program Requirements 8
Implementation 9
Conclusions and Further Work 11
Bibliography 12

Abstract

Machine learning is based on two things, data and statistics, by feeding data into a computer and applying statistics the application can learn any type of pattern behind the administered data. Based on this we wanted to create an application that allows people to control and improve their dental hygiene by listening to the user brushing their teeth. However, during the study of the project, it was identified that for this specific objective there was not enough data for the development and execution of this project. Therefore, we decided to create a tool to gather, label and categorize audio files so they can be used in any kind of machine learning project.

This tool was designed not only to tag any type of video or audio, allowing the user to extract their data, but also to guarantee the privacy of the information by keeping all the input files locally and sending only the extracted audio to a protected service. Furthermore, this tool was developed to be compatible with both mobile devices and desktop computers. It is important to note that this tool can also be used in many other areas that involve audio detection as it can easily be expanded to suit the researcher's need.

Introduction

When I started developing mobile applications in 2012, my goal was simple, to create an application that allows the user to perform a task, for example a note-taking application, a calculator, or an application to keep track of their expenses.

With the computational change that we are currently experiencing and the introduction of machine learning in our pockets where our phones can identify who we are just by looking at us. I think we have a huge area to explore that is currently underutilized and that is audio data. With today's technology, our phones can identify just about any sound we teach them.

At the beginning of the project, our goal was to create an application to aid the user's oral care and improve overall dental health. However, while working on it, we noticed that the publicly available datasets for audio learning were almost non-existent and those available were not reliable enough to build the application we wanted.

For this reason and to find a better way to collect audio datasets, we were faced with the decision to use a third-party tool or to create our own tool to tag the data, extract it, and separate it from reliable sources such as videos. We decided to create our own tool for two reasons we wanted to always guarantee privacy and to be able to use the tool from anywhere.

The objectives of this project were:

1. Collect information by creating a tool that allows the collection of audio data sets applying the principles of Machine Learning.
2. Provide a flexible tool capable of running on any device, extracting the audio from any type of video as a source of information for updating and providing feedback on the database.
3. Guarantee the privacy of the information collected and allow it to be used safely from anywhere.

Problem Statement

Currently, most projects based on software development implement Machine Learning as part of their execution within the information processing. However, there is no knowledge of a tool that allows audio learning through the extraction of audio from different types of videos to be stored in datasets. This project aims to create such a tool which will allow audio data capture in a safe, flexible and efficient way while always maintaining privacy.

Background and Related Work

In order to achieve the development of the project, part of its research consisted in finding tools that could allow the capture of labeled audio data. As a result of the above, the existence of some third-party libraries was evidenced, through which the labeling of audio data sets was possible. Some of these tools are:

- EchoML (ritazh, 2020): a web-based tool to visualize audio waveforms and label the features for extraction.
- audio-annotator (CrowdCurio, 2020): web-based tool to annotate developed by CrowdCurio where you can visualize the audio data into both the spectrogram and the waveform.
- audio-labeler (hipstas, 2020): web-based tool to annotate developed by hipstas.
- praat (praat, 2020): desktop-based application to both label and extract features from an audio file.
- peaks.js (BBC, 2020): web-based application developed by BBC R&D to allow users to make accurate clippings of audio content in the browser, using a backend API that serves the waveform data.

At the time of research, the only library that supported video feed was peaks.js, however, none of them allowed to extract the labeled audio into different files to be sent to the server. All of them relied on sending the entire file into the server and then splitting with the annotated values from the website. This was a drawback of all of them as we did not want to handle video files as privacy can be breached.

Program Requirements

Based on the needs described above, to carry out the execution of the project, it must be considered that the creation of the tool must contain a minimum of initial user requirements that allow the capture of audio data and labeling of the information under the conditions raised as necessary and thus the subsequent collection of information. The user requirements described below are a sample of the characteristics to be implemented within the process of developing the project in question to achieve the proposed objective:

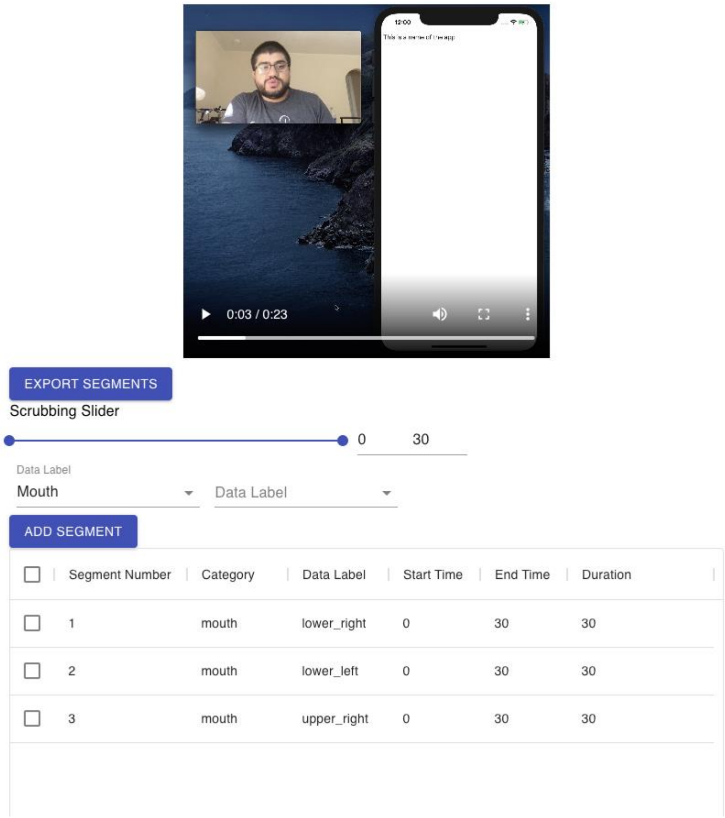
1. The application shall run on both mobile and desktop platforms.
2. The application shall be able to work with the most common video formats.
3. The application shall run locally to the user and only send the extracted audio.
4. The application shall be flexible to update the categories or labels without requiring a deployment.

Thus, the previous characteristics will allow the development of the project base applying the principles of Machine Learning, in order that the tool from its initial structure, shall be flexible and efficient and can fulfill the purpose of extracting audio data from the most common video formats. The foregoing also implies that the tool can be updated or modified according to the needs that arise both from the user and from the development.

Implementation

To do the implementation we decided to split the work into 3 main phases, first we had to create a way for the user to select the segment of a file that corresponds to a specific tag, the second phase was adding a way for the user to use any kind of video they have collected and the final phase where we had to split the audio and send it to the server.

To be able to provide the service on both mobile and desktop we decided to use React.js to create a web application to tag the data from files, with this web application we solved the first phase of implementation with a relatively good overall code base for the next phases.



0 30

Data Label: Mouth

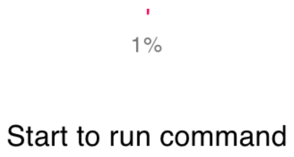
Data Label

☐	Segment Number	Category	Data Label	Start Time	End Time	Duration
☐	1	mouth	lower_right	0	30	30
☐	2	mouth	lower_left	0	30	30
☐	3	mouth	upper_right	0	30	30

Figure 1 User interface for labeling.

While working on the first phase we found a library to handle video transcoding and audio extraction called ffmpeg.wasm, the library is a pure port implementation on web assembly based on the desktop suite of tools ffmpeg. This library allowed us to do the second phase where at the beginning of the web application we loaded the video and transcode it into a more suitable format.

Figure 2 Initial step to transcode the selected file.

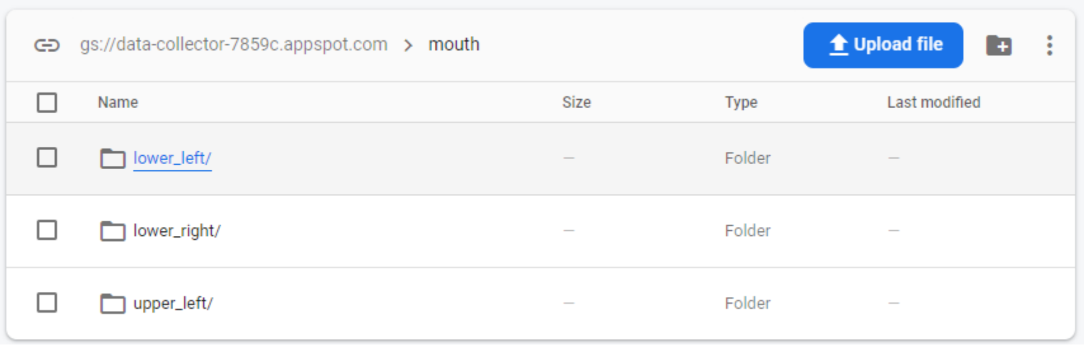


1%

Start to run command

Figure 3 Transcoding library running the conversion.

And finally for phase 3 we used ffmpeg.wasm in conjunction with firebase to do both the file management and tag configuration from the server.



□	Name	Size	Type	Last modified
□	lower_left/	–	Folder	–
□	lower_right/	–	Folder	–
□	upper_left/	–	Folder	–

Figure 4 Firebase console

Conclusions and Further Work

From the analysis of the project and during its implementation, it can be concluded that, in the way this tool was designed, it will be able to fulfill the purpose of collecting audio datasets specifically aimed at oral care where the next step is to try to help users understand their oral hygiene.

It is important to note that while the development of this project has a specific end goal, it must be said that this tool can also be used in many other areas that involve audio detection because it can be easily expanded to suit the researcher's need.

Currently the tool is being used by a small group of people in ACI to do some preliminary model building aimed towards oral care. There was a discussion to use this tool in other projects such as personal care projects and even location awareness projects.

Bibliography

BBC. (2020). *peaks.js*. Retrieved from peaks.js: https://github.com/bbc/peaks.js

CrowdCurio. (2020). *audio-annotator*. Retrieved from audio-annotator: https://github.com/CrowdCurio/audio-annotator

Facebook. (2020). *Reactjs*. Retrieved from Reactjs: https://reactjs.org/

ffmpegwasm. (2020). *ffmpeg.wasm*. Retrieved from ffmpeg.wasm: https://github.com/ffmpegwasm/ffmpeg.wasm

hipstas. (2020). *audio-labeler*. Retrieved from audio-labeler: https://github.com/hipstas/audio-labeler

praat. (2020). *praat*. Retrieved from praat: https://github.com/praat/praat

ritazh. (2020). *EchoML*. Retrieved from EchoML: https://github.com/ritazh/EchoML