Machine Learning-Based Automatic Audio Censoring System

Rituraj Ingale[1], Sairaj Mote[2], Prasad Kirkade[3], Prof. V.B. Baviskar[4].

[123] Student, Department of Computer Engineering, SKNSITS, Lonavala, Maharashtra, India.
[4]Assistant Professor, Department of Computer Engineering, SKNSITS, Lonavala, Maharashtra, India.

Abstract - In this paper, we are going to create an online tool that will automatically censor explicit content from an audio file (or specified words) using Natural Language Processing and Audio Processing. The implementation of the system relies on two components: First, audio is recorded using any audio recording device like microphone, smartphone or any recording device that can produce output in the system specified audio format, second, audio processing is done on cloud computing platform for speed and ease of use. In the initial phase, we are going to use Google Speech To Text API for converting audio to time-stamped trans-scripture that then can be used to identify and replace explicit words from the audio file. When the processing is done on the cloud the output file will then can be downloaded by the user.

The challenges that we will be solving are, creating a deep learning model that identifies explicit words without converting them to text and creating a PWA that can run across all major form factors.

Keywords - Natural Language Processing, Google Speech To Text, Machine learning, Cloud Computing, Audio Processing.

I. INTRODUCTION

Due to the recent increase in media consumption, more and more people are being exposed to content that may not be suitable for them. Also, there are professional or individual creators that spend a lot of time censoring their content for online websites like YoutTube. If there is some explicit content in that audio that may get demonetized or banned entirely. The proposed application will process such audio either a file at a time or in real-time and search for explicit words. Then it will replace those words using a default beep sound or any desired audio clip. It uses Natural Language Processing for semantic analysis of the audio and filtering the content based on it. All this process will be done using cloud computing so that no matter what device is used to censor the audio file, it won't matter if it has the processing power required for audio processing. It will be a Progressive Web App which means it can be installed as an app on android devices and desktop computers without any complications and it will be highly salable and accessible. This will increase the productivity of individual creators and audio censoring professionals.

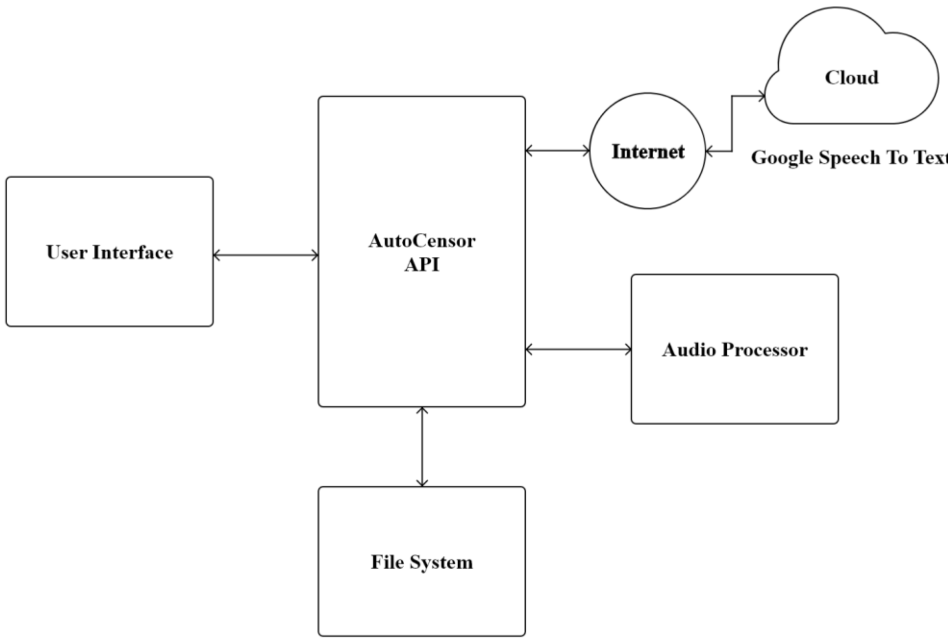
II. LITERATURE SURVEY

Weijiang Feng et.al[1] proposed research on the High-Performance Audio Matching which would learn features using Convolutional Deep Belief Network. The research's goal was to implement a technique to learn specific features of a given audio dataset using CDBN. The audio feature extraction in the traditional CENS based algorithm was easily implemented using this new Convolutional Deep Belief Network. It matched the audio with the existing language.

Alessio Ferrari et.al[2] illustrated how the NLP model that was formerly purely research-based can also be commercially implemented. The most important example was using Natural Language Processing to extract customer requirements. In Software and many other project-based industries, Requirement Analysis is an important aspect for the Planning and Model Designing of the project.

Pranav Kaushik et.al[3] illustrated a survey on how efficiently Natural Language Processing is implemented using Statistical Learning, Deep Learning, and Reinforcement Learning. They concluded on the discovery that NLP is best implemented using Deep Learning and Neural Network models. Reinforcement Learning is better to implement reward oriented environment systems.

III. PROPOSED SYSTEM APPROACH



The proposed system is a simple Web App. It is an interactive User Interface that allows the user to upload the required audio file. Then this audio is processed to make censored content. This is a fast-paced application that lets the user get its audio marked with the required time frames of the unwanted words.

It consists of a Central API server that is always ready to accept an input file. This is a Flask Library based Python Scripted Server that accepts file path as a request from the client and processes the audio from that file path.

It receives the required file from the File System. The Audio file is required to analyze and deduce the transcript from the file. This file is converted into a series of frames. These data frames are sent to the Google Cloud API.

Google Speech-To-Text API is a cloud-based API that accepts the data frames of audio and supply them to some NLP model. These models are generally RNN based Deep Learning Models that are used to classify the audio into a most probable string on language. This API returns the audio transcripts with their time stamps.

The received transcript is checked for the list of unwanted words. If there exists such a word in the transcript it is immediately sent to process audio. The audio processor is required to process the word and adds the Beep sound to the given timeframe.

IV. METHODOLOGY

1. **Application Program Interface (API):**
   A Central API is created that acts as the control center of the whole Web Application. It is a server script based on Python Library Flask. This helps in creating an API based request-response communication structure.

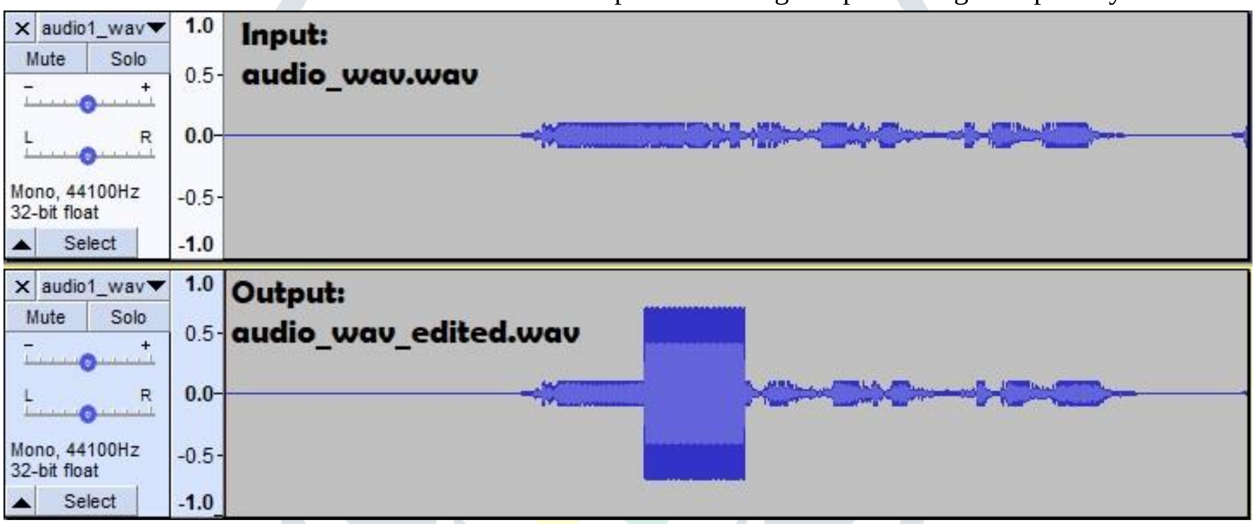
```
import flask
app = Flask(__name__)
```

```
if __name__ = "__main__":
        app.run()
        pass
```

This is the simplest and most efficient method of building an API based architecture.

2. **Web Application Implementation:**
   A Web Application implementation is achieved using HTML, CSS, and JavaScript. HTML and CSS are the basic backbones of the whole Front End User Interface. JavaScript acts as the logic and the brain behind Front End. It helps the webpage connect with the server asynchronously.
   JavaScript uses the technique of Asynchronous Java and XML, AJAX to communicate with the Central API. JavaScript sends and receives in only text format.

3. **Audio Processing (WAVE):**
   import wave
   A WAVE file is an audio file saved in the.wav format. These files are easiest to understand and implement the format of audio. These are low compression high processing capability format.



4. **Speech Recognition (Google Speech-To-Text):**
   Google Speech-To-Text is a Google Cloud API. It is a Deep Learning-Based model used for NLP and Speech Recognition. It is an implementation of Speech Recognition made by Google. Its configuration helps us with different types of user requirements. Like enable_word_time_offset helps to get Time Offset values on the transcript. This helps in the point of the start and end of a particular word.

V. CONCLUSION

We are going to propose a system that will improve the workflow of audio editing professionals as well as improve parental control due to its efficiency in filtering age-inappropriate audio content. We are going to train a deep learning model which will be far more efficient than other available NLP models and it is a progressive web application lends itself for greater platform flexibility than other existing applications without requiring powerful hardware on the device.

ACKNOWLEDGMENT

This work is supported as a blind assistive system for any state in India. Authors are thankful to the Faculty of Engineering and Technology (FET), Savitribai Phule Pune University, Pune for providing the facility to carry out the research work.

REFERENCES

[1] Weijiang Feng, Naiyang Guan, Zhigang Luo "High-performance Audio Matching with Features Learned by Convolutional Deep Belief Network" 2016.

[2] Alessio Ferrari "Natural Language Requirements Processing: from Research to Practice" in ACM/IEEE 40th International Conference on Software Engineering: Companion Proceedings 2018.

[3] Pranav Kaushik, Akanksha Rai Sharma "Literature Survey of statistical, deep and reinforcement learning in Natural Language Processing" in ICCCA 2017.

[4] Hitoshi ISAHARA "Resource-based Natural Language Processing" National Institute of Information and Communications Technology 2007.

[5] Bill Z. Manaris" Natural Language Processing Tools and Environments: The Field in Perspective" Computer Science Department University of Southwestern Louisiana.

[6] Mladen Stanojevic, Sanja Vranes "A Natural Language Processing for Semantic Web Services" IEEE 2005.