Murad Ismayilov
Nahian Rahman
Claire Anna Wirtchafter

4/10/2023

MAIS 202 - Final Project

Deliverable 1

Choice of dataset:

We chose to use Google Research's audio set (https://research.google.com/audioset/). It is a large data collection with over 2-million videos. These videos are also divided into subcategories of sounds such as music, speech, vehicles, and a variety of objects.

In case we need to add more data to improve the accuracy of the model, we can use some datasets from Kaggle, such as this dataset for signal processing or musical chord classification dataset

Methodology:

1. Data preprocessing: the google dataset of various videos with classified sounds contains accurate and easy-to-extract data. In order to extract audio from the videos we will just use the moviepy python library. Then, we will use FFT (Fast fourier transform) in order to extract features from the audio files and pass them to our model (possible to do in the form of a spectrogram).

2. Machine learning model: we want to classify and extract components of an audio file. We can build a CNN model using Tensorflow (or PyTorch). It is a very popular and easy choice due to the straightforward tensorflow api. CNNs are less computationally intensive than RNNs and they can more easily extract features. The only significant con of a CNN is lack of temporal understanding. RNNs do take into account the temporal dimension and can process audios of varying length. That being said, RNNs are very computationally expensive, learn slower, and are not as good at identifying multiple

audio components simultaneously. Therefore an RNN model would be better suited for command recognition, but in our case, CNN is superior.
An alternative would be to use Hugging Face and to fine-tune a pre-made model.

3. As our model aims to classify different types of sounds, we can start with a confusion matrix as it provides a detailed breakdown of how the model's predictions compare to the actual labels. Using this, we can assess True Positives (correctly identified sounds), True Negatives (correctly identified absence of sounds), False Positives (incorrectly identified sounds) and False Negatives (missed sounds).

   Then, precision-recall can be used to measure the proportion of true positive predictions out of all positive predictions. Maximizing precision will ensure that there are fewer False Positives, so we can avoid incorrectly removing sounds from the audio track. Maximizing recall will ensure that there are fewer False Negatives, so sounds that are present in the audio will not be missed.

   Accuracy should be used to measure the proportion of correctly predicted instances (both true positives and true negatives) out of the total instances.
   Logistic loss is also important in training a classification model as it gives a probabilistic estimate or confidence score for each class that is predicted.

Application:
The user inputs an audio track (pre-recorded/can record using a microphone). The model then detects the different types of sound present and displays them to the user.
Time permitting, our model will also allow the user to select sounds to remove from the audio track. The edited audio will then be played back to the user.