Gender recognition by voice

Dvir Gev & Aviad Gilboa

209530583 316080514

Project link

Introduction

- The purpose of this project is to develop and implement an AI system that can accurately predict the gender of a speaker based on a given voice recording.
- We have selected this open-source git as our starting point for our work.
- Through the process of developing and optimizing this model, we have gained an understanding of various concepts in the field of machine learning, including improving and optimizing the model to achieve better results

Introduction

Accurate prediction of the gender of a speaker can be valuable in a variety of situations:

- Voice assistants (Siri, Alexa, Google Assist,...)
- Virtual worlds (Metaverses, video games, ...)
- Services on the phone (automats and humans)
- Defense services (e.g. women's services at the club)
- And it can also be a basis for other uses…

Dataset

- The dataset we using in the project is the commonvoice-Mozilla dataset.
- In our approach to coding, we took the following details into account:

Adventages of The dataset

- Large amount of data
- The length of each file is avg 4 second. Learning from short time with the real-time speaker.
- diversity of English speakers
- diversity of aged speakers
- equal number of any gender voice

Disadvantages of The dataset

- Only English speaker – other languages have a different accent and volume
- No children’s voices (less relevant)
- No recording interruption, like in the real world.

Current Approach

Our original Git project wanted to write a tutorial to teach about deep learning, using Tensorflow framework.

He focused on the following:

- Preparing the Dataset
    - He only filtered the labeled samples in the genre field.
    - He balanced the dataset so that the number of female samples is equal to male samples.
    - He used the Mel Spectrogram extraction technique to get a vector of the length 128 from each voice sample.
- Building the Model
    - He used a deep feed-forward neural network with 5 hidden layers
- Training the Model
    - He trains the model using the previously loaded dataset.
- Testing the Model
    - He reached 91% accuracy on samples that the model has never seen before.

Improvements - Accuracy

- The approach for training the model using a neuron network.

Current approach:

(1) Convert all the dataset to Wav audio files using ffmpeg.

(2) Adding features to our NumPy vector using a function that take details from the voice file and returns the vector file.

(3) For short the time the model training with the new vector, That be like 5 times bigger we add early stopping for the train model

(4) Change the value of the Dropout from 0.3 to 0.55 in the training progress to avoid a minimal the overfitting

(5) Adding PCA algorithm to inspect the un-relevant features for our predicted mission.

(6) Calculate the optimal Incline for deciding the index of the last important features we will use for the predictions and training.

(7) Saving the PCA model for use with this information in our testing file mission

Improvements - Performance

- noisy files – why do we not insert
- Reliable speaker – we consider that the speaker not try to lie about his gender.
- The adding features – make parameters worthier, but it save time (early stop the improvement of the model).
- Overfitting
- PCA
- Predict our model on new files.

Results

Baseline	
Accuracy	91.16%
Precision	90.4%
Recall	91.62%
F1-Score	91.01%
Time	116.37 Sc

Add features to the train and predict vector without PCA	
Accuracy	92.41%
Precision	88.8%
Recall	97.8%
F1-Score	92.64%
Time	84.83 Sc

Add features to the train and predict vector with PCA	
Accuracy	97.74%
Precision	98.59%
Recall	96.76%
F1-Score	97.66%
Time	58.75 Sc

Code Questions

- PCA.py :
  1. plotPCA: from the internet with adding our parameters.
  2. find incline – from Lior and Gal project.
- convertmp3ToWav.py: by Aviad
- fromSali.py
- makeNewDataBase.py: by Aviad & Dvir
- more_test.py: by Dvir
- test.py: from the basic project and changes by us.
- utils.py: from the basic project and changes by us.
- train.py: from the basic project and changes by us.
- preparation.py: from the basic project and changes by us.

Discussion – Algorithms

- PCA
- Calculate the incline of the last feature in the PCA
- Neuron network
- Early stoping

Discussion – Open Questions & Future Work