Text Independent Speaker Recognition Application

EE 769 Course Project

By -
Huzefa Chasmai 15D170013
Mohit Patil 150050017

Application Description

There were two target applications that we had planned to work on.

1) *Voice Based Application Unlocker*

2) *Voice Biometric Attendance Marking*

We implement a basic bootstrapping application that can be extended to above.

Problem Description:

Speaker Recognition:

It comprises of two problems:

- *Speaker Identification*:

  Given an utterance identify whose utterance amongst a set of speakers.

- *Speaker Verification*:

  How likely it is that the utterance is of the given person

Implementation Details

Speaker Identification:

- Hidden Markov Models : We model different HMM’s for different speakers and given a new sample the model having the highest log likelihood
- Data Set : We used the LibriSpeech ASR Corpus for training our HMM models
- Features : Used MFCC’s and its deltas and double deltas as features (39 dim)
- Used the hmmlearn library for the HMM’s and the librosa library for most of the sound processing

Speaker Verification:

- Found the minimum bounds of the log likelihood for a particular model and checked whether given input is in the bound.

Some Results :

Tuning of hyperparameters : Used 20 % of the data as validation to tune the parameters.

Hyperparameters : (num of states in hmm, num of speakers in training set, num of audio files per speaker, num of iterations of the hmm training EM algorithm)

Validation accuracy results :

(50, 40, 10, 10) : 92.67% ;

(40, 10, 10, 10) : 91.41% ;

(50, 40, 40, 10) : 98.99% ;

(50, 40, 20, 10) : 91.92%

Learning and Experiments

- While the HMM's performed well for speaker identification it didn't perform so well for speaker verification
- Even in the train samples for the HMM, the probability of acceptance by the HMM was very less
- The probability of acceptance depends highly on the utterance length
- It was very difficult to set a threshold of acceptance probability for speaker verification
- Storing and creating a new model for every speaker we want to consider is tough
- Having knowledge of all speakers beforehand is difficult

Platform uses for building application

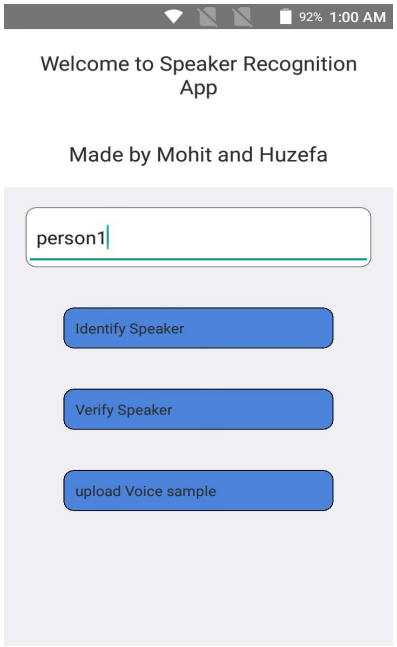
Frontend

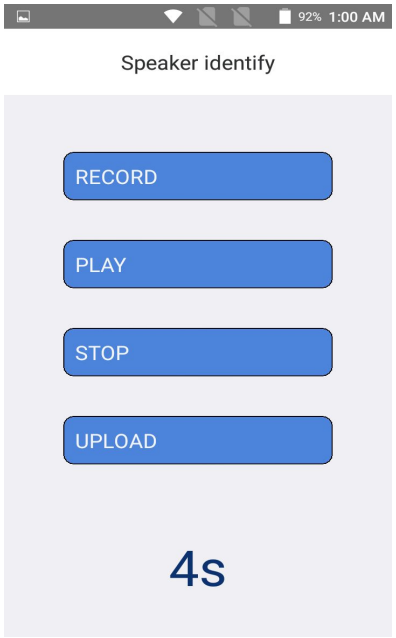
- We made an Android as well as an IOs based application using react-native framework

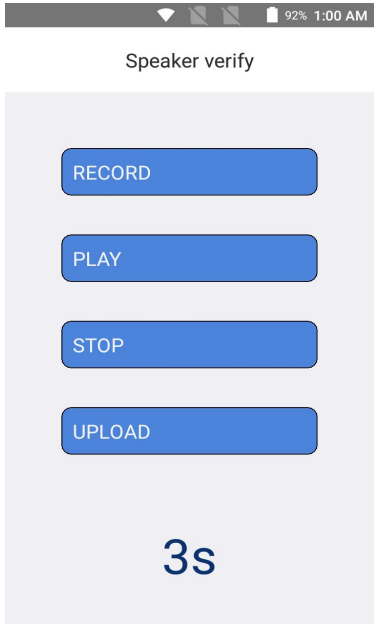
Backend

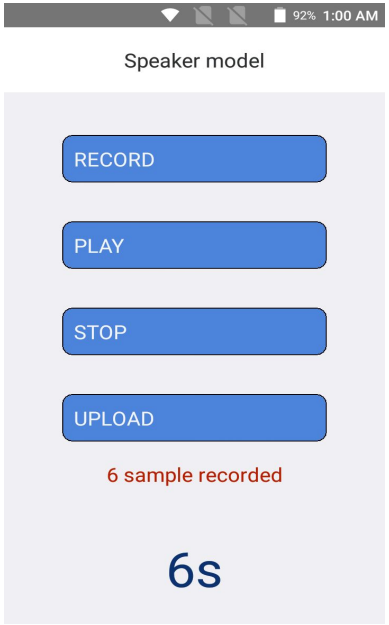
- We made server of the application in python using Django framework

Demo :


92% 1:00 AM
Welcome to Speaker Recognition App
Made by Mohit and Huzefa
person1
Identify Speaker
Verify Speaker
upload Voice sample


92% 1:00 AM
Speaker identify
RECORD
PLAY
STOP
UPLOAD
4s


92% 1:00 AM
Speaker verify
RECORD
PLAY
STOP
UPLOAD
3s


92% 1:00 AM
Speaker model
RECORD
PLAY
STOP
UPLOAD
6 sample recorded
6s

Thank You