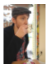
Speaker Verification System

Phil Ashe, Paul Mahoney, Jason Nguyen, Liam Shea
Faculty Advisor: Prof. Maciej Ciesielski

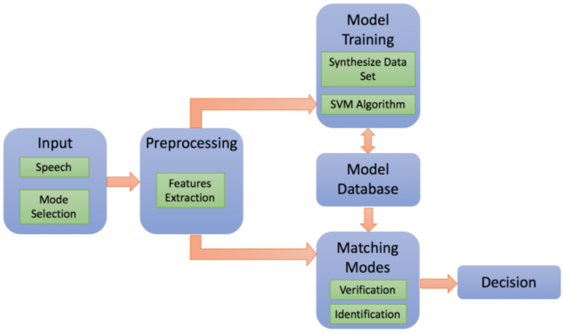


Abstract

Biometrics are a way of identifying an individual by their biological characteristics, among which, fingerprint and retinal scanning, facial recognition, and vocal distinction have become the most utilized identification factors. However, while retinal and facial recognition are considered more secure, they are also perceived as being more intrusive and not suitable for public uses, and fingerprints prone to leave traces behind, which can be a huge security liability. In this project, our team consider a different approach to implementing a gateway security layer by utilizing vocal distinction.

Speaker Identification System (SVS) aims to integrate voice biometrics identification on top of existing technology as a simple, non-intrusive, and easy to operate security measure. The user simply inputs a designated phrase to the system, either to be created as a new entry, or to be verified for further access.

Block Diagram



Specifications

Specification	Goal	Actual
Speaker recognition accuracy	>90%	>70%
Database	SQL database	Unstructured data
Response time	<1 sec	~1 sec
SVM Training time	<1 sec	<5 secs
Training samples needed	<10 recordings	~50 recordings

System Overview

The system operates by analyzing the spectral densities of the user-provided speech input, and the users choose between two modes of operation:

- Training Mode: Train the system to recognize a new speaker
  - Takes multiple recordings (~10-50) of the user speaking the key phrase
  - Pre-process and extract the feature vector from each recording
  - Save extracted feature vectors in internal database
  - Utilize built-in machine learning algorithm (SVM) to form a binary classifying model for that user
- Matching mode: Match a speaker against the user profiles that already exist in system.
  - Exists as two sub-modes: Identification and Verification
  - User's input is collected, pre-processed, and has its features extracted in the same way as in training a new user's profile
  - Identification mode attempts to match the user's input to existing profiles and determines the best match
  - Verification mode verifies whether the user is who they claim to be

Results

With a small speaker set (<5 speakers), the system is able to verify speaker identity ~80% of the time. It works particularly well when this sample set includes both male and female speakers. However, the number of samples needed to build an adequate training model is around 50 samples per user, which is impractical.

With a larger speaker set, our system is not able to reliably verify speaker identity. This is caused by too much variance in our input data. We identified some possible causes of this variance:

- Natural variation in the way a person speaks their phrase (slightly faster or slower, different inflection, etc.)
- Users impatience during training phase, causing them to speak faster and faster
- Silence truncation did not adequately line up signals
- Environmental noise

Acknowledgement

Our team would like to give our most sincere appreciation to Professor Maciej Ciesielski, our advisor, for his patience with our failures along the way and his tireless effort to set us back on the right track; to Professor Paul Siqueira and Professor Aura Ganz, our evaluators, for sharing their valuable time and offering us feedback; to Professor Patrick Kelly, for some advice he offered us; and to Professor Christopher Hollot for his guidance throughout the term, and his effort to make SDP an enjoyable experience.

Department of Electrical and Computer Engineering
ECE 415/ECE 416 – SENIOR DESIGN PROJECT 2017
College of Engineering - University of Massachusetts Amherst

SDP17

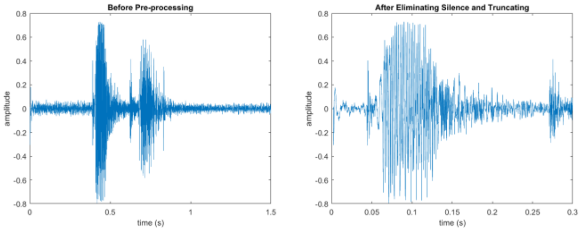
Voice Recording

- Recordings done with Blue Snowball iCE USB microphone
- Recording parameters: 44.1 KHz sample rate, 16 bits per sample, single channel

Audio Pre-processing

Time-domain processing of voice signals to improve accuracy of the frequency-domain feature extraction

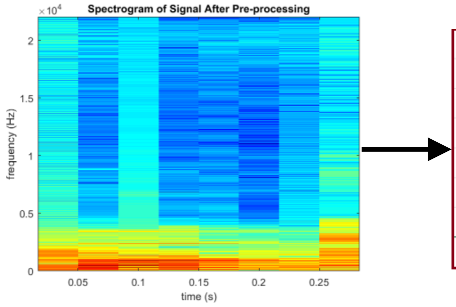
- Silent regions are eliminated by throwing out all samples with amplitude below threshold
- Remaining signal is truncated at 300 ms



Feature Extraction

Transform relatively large audio file into a small set of data points

- Audio signal is first pre-processed in time domain as above
- Spectrogram transform is applied to processed audio signal
- Features extracted by finding frequency bin with highest power content



Feature Vector
344.53
441.43
441.43
893.63
290.7
441.43
430.66
452.2

Support Vector Machine (SVM)

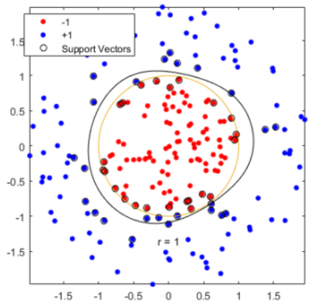
Support Vector Machine (SVM) is a supervised machine learning algorithm used to classify data

- To train an SVM to recognize a user:
  - Read in extracted features for that user. These are part of the "positive" class
  - Read in extracted features for all other users. These are part of the "negative" class
- Each feature vector is plotted as a coordinate in an n-dimensional hyperspace
- The MATLAB SVM algorithm fits a boundary between the positive and negative classes

Feature Matching

To match a user:

- A new unclassified sample is checked against the user's SVM model to see which side of the boundary it falls on
- Below an example of an SVM binary classifier in a 2D sample space



Cost

Development		Production	
Part	**Price**	**Part**	**Price**
Microphone	$50	Microphone	$30
Processing Unit	$50	Processing Unit	$30
Display	$60	Display	$40
Total	$160	Total	$100