Human Computer Interaction – Project I
Speaker Identification Tool
Ozan Çağlayan 2011-2012

1. What is Speaker Identification?

Speaker Identification (or Speaker Recognition) is the process of validating a user's *claimed* identity using characteristics extraced from their voices.

Speaker Recognition: *Recognizing **who** is speaking*
Speech Processing: *Recognizing **what** is being said*

2. Phases

2.1. Enrollment

During *enrollment*, the speaker's voice is recorded and typically a number of features are extracted to form a *voice print*, t*emplate*, or *model*.

2.2 Verification

In the *verification* phase, a speech sample or "utterance" is compared against a previously created voice print.

3. Variants of speaker recognition

3.1 Text-Dependent

If the text must be the same for enrollment and verification, the system is called *Text-Dependent.* (Ex: A common pass phrase, passwords, PINs, knowledge-based information, etc).

3.2 Text-Independent

The text during enrollment and verification may be different. In fact, the enrollment may happen without the user's knowledge (ex. Forensic Applications).

4. Technology

Frequency estimation, hidden Markov models, Gaussian mixture models (GMM), pattern matching algorithms, neural networks, matrix representation, vector quantization, decision trees, etc.

5. Tools

In this section you will find brief explanations about the tools & technologies used to develop the *Speaker Identification* tool.

5.1. MARF

MARF stands for *Modular Audio Recognition Framework.* It contains a collection of algorithms for Sound, Speech and Natural Language Processing arranged into an uniform framework to facilitate addition of new algorithms for pre-processing, feature extraction, classification, parsing, etc. implemented in *Java.*

The main goal behind MARF is to build a general open-source framework to allow developers in the audio-recognition industry to choose and apply various methods, contrast and compare them, and use them in their applications.

Please visit *http://marf.sourceforge.net* for further details.

5.2. Python & Qt4

The User Interface and interfacing with MARF is accomplished using Python programming language and PyQt4 (Python bindings for Nokia's Qt Toolkit).

Please visit http://www.python.org and http://qt.nokia.com for further details.

5.3. Sound Recording

The sound recording for sample acquisition is made using *PyAudio*, a Python interface to the popular *PortAudio* C library.

Please visit http://people.csail.mit.edu/hubert/pyaudio/ for further details.

5.4. Source Code

The source code is kept under *GitHub,* a public and free GIT repository for social coding.

Please visit https://github.com/ozancaglayan/SPID for source codes.

6. How does it work?

6.1. Speaker Database

Speakers are stored in a comma-separated (CSV) file called *speakers.txt.* The format is as the following:

```
<id:int>,<name:string>,<training-samples:list>,<testing-samples:list>
```

An example *speakers.txt* consisting of 7 speakers is below:

```
1,Serge,serguei1.wav|serge-family-cardholder.wav|serge-fed-adv-shipyard.wav|
serge-fed-sci-station.wav|serge-fed-starbase.wav|serge-fed-trade-station.wav|
serge-good-standing.wav|serge-insured-person.wav|serge-master-card.wav|serge-
mysterious-disappearance.wav|serge-primary-cardholder.wav|serge-pulse-
cannon.wav|serge-spouse.wav|sergueiandhismom.wav,serge-label.wav
2,Ian,ian1.wav|ian2.wav|ian3.wav|ian4.wav|ian5.wav|ian6.wav|ian7.wav|ian8.wav|
ian9.wav|ian10.wav|ian11.wav|ian12.wav|ian13.wav|ian14.wav,ian15.wav
3,Steve,steve1.wav|steve-train1.wav|steve-train2.wav|steve-train3.wav|steve-
train4.wav|steve-train5.wav|steve-train6.wav|steve-train7.wav|steve-train8.wav|
steve-train9.wav|steve-train10.wav|steve-train11.wav|steve-
train12.wav,steve2.wav|steve-test1.wav|steve-test2.wav
4,Jimmy,jim10.wav|jim11.wav|jim12.wav|jim13.wav|jim14.wav|jim15.wav|jim1.wav|
jim2.wav|jim3.wav|jim4.wav|jim5.wav|jim7.wav|jim8.wav|jim9.wav|jimmy1.wav|
jimmy2.wav,jim6.wav
5,Dr. Suen,suen1.wav|suen2.wav,suen2.wav
6,Margarita Mokhova,rita1.wav|rita2.wav|rita3.wav|rita4.wav|rita5.wav|rita7.wav|
rita8.wav|rita9.wav|rita10.wav|rita11.wav|rita12.wav|rita13.wav|rita14.wav|
rita15.wav,rita6.wav
7,Ozan,ozan1.wav|ozan2.wav|ozan3.wav,testing-20111212_232323.wav
```

6.1. Samples

- Training samples are stored in the `training-samples` directory,
- Testing samples are stored in the `testing-samples` directory.

Samples are standard WAVE files with the following properties:

- Audio Format: PCM SignedInt (WAV)
- Sampling Rate: 8000 Hz
- Sample Size: 16-bit
- Channels: 1 (Mono)

In the PCM encoding an analog signal is represented as a sequence of amplitude values. Range of amplitude is given by the *sample size* which represents the number of bits that a PCM value consists of. In our case the range is [0-65535] but since we are using signed integers the range is actually [-32768, 32767].

6.2. Preprocessing

The preprocessing layer implements various filters in order to prepare the speech signal for further feature extraction.

All input is *normalized* before the preprocessing layer to maintain an average amplitude level between recordings in order to ensure that features will be comparable.

Only the method that is used in this tool will be explained briefly for the sake of simplicity.

Offered preprocessing methods in MARF are: *Raw-meat, Normalization, Noise removal, Silence removal,* ***Endpointing,*** *low-pass filter, high-pass filter, band-pass filter, high-frequency boost.*

6.2.1. Endpointing

By the *end-points* we mean the local minimums and maximums in amplitude changes. A variation of that is whether to consider the sample edges and continuous data points (of the same value) as end-points.

See http://www.clear.rice.edu/elec301/Projects99/wrcocee/endpt.htm for the details of end-point detection.

6.3. Feature Extraction

Feature extraction layer implements various feature extraction algorithms in order to characterize the sound samples.

Offered feature extraction methods in MARF are: *FFT (with Hamming window),* ***LPC (Linear Predictive Coding)****, F0 (The Fundamental Frequency), Min/Max, Feature Extraction Aggregation, Random Feature Extraction.*

6.3.1. LPC (Linear Predictive Coding)

LPC evaluates windowed sections of input speech waveforms and determines a set of coefficients approximating the amplitude vs. frequency function.

The LPC coefficients evaluated at each windowed iteration, yields a vector of coefficient of size *p.* These coefficients are averaged across the whole signal to give a mean coefficient vector representing the *utterance.* Thus, a *p* sized vector is used for training and testing. A *p* value of around 20 was observed to be accurate and computationally feasible.

6.4. Classification

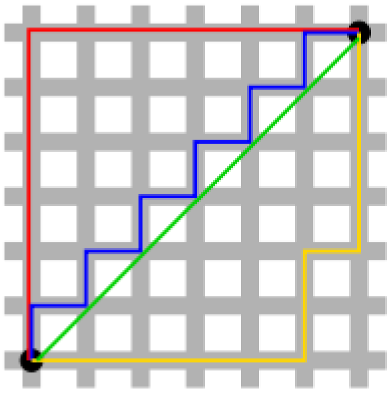
The classification layer is responsable to decide whether 2 feature vectors are matching each other based on some threshold value or other technical tools.

Offered classification methods in MARF are: ***Chebyshev Distance***, *Euclidean Distance, Minkowski Distance, Mahalanobis Distance, Diff Distance, Artificial Neural Network, Random Classification.*

6.4.1. Chebyshev Distance

Chebyshev distance is used along with other classifiers for comparison. Chebyshev distance is also known as a city-block or Manhattan distance.

$$d = \sum_{i=1}^{n} |x_i - y_i|$$



All three pictured lines (red, blue and yellow) have the same length (12) for the same route. In Euclidian geometry, the green line has length 6×√2 ≈ 8.48.

7. Experimentation

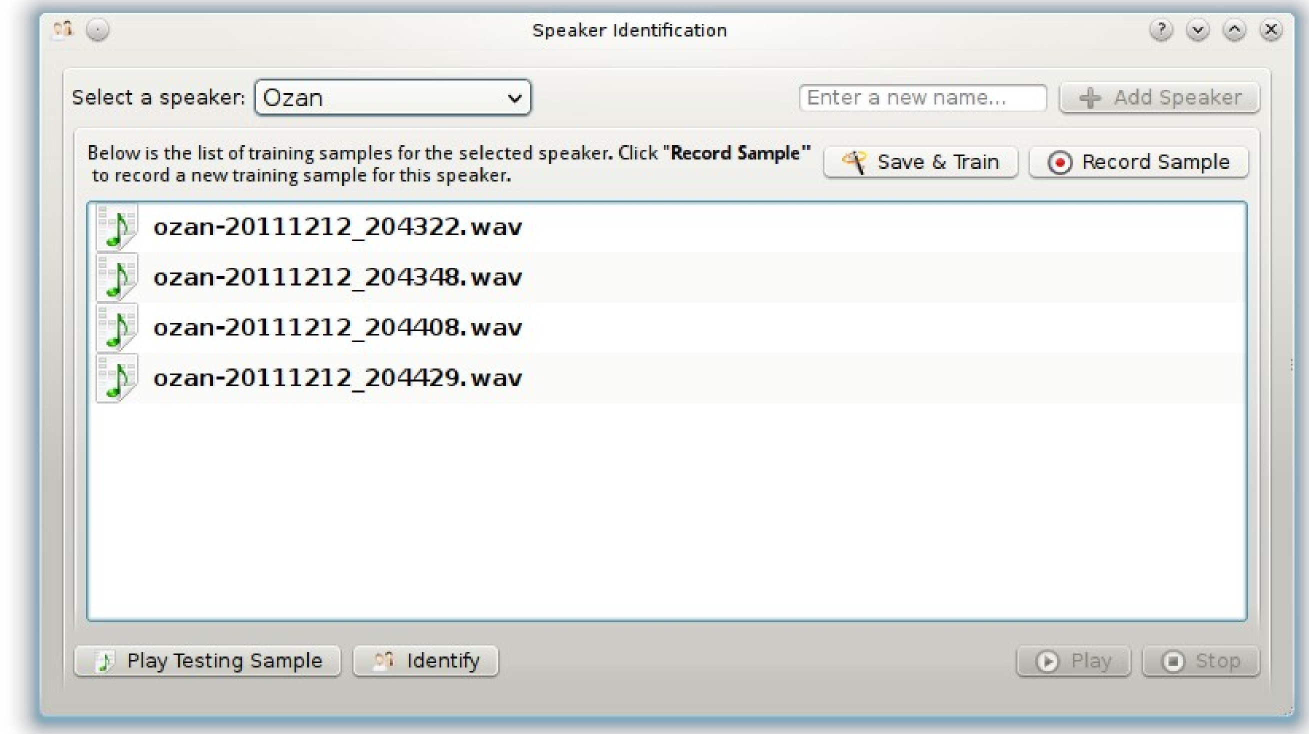
The results from the *MARF Manual* shows that the best results are achieved using:

- ***Endpointing*** in Preprocessing layer,
- ***LPC*** in Feature Extraction layer,
- ***Chebyshev Distance*** in Classification layer.

Guess	Run #	Configuration	GOOD	BAD	Recognition Rate,%
1st	1	-endp -lpc -cheb	24	5	82.76
1st	2	-raw -aggr -eucl	22	7	75.86
1st	3	-norm -aggr -diff	22	7	75.86
1st	4	-norm -aggr -cheb	22	7	75.86
1st	5	-raw -aggr -mah	22	7	75.86
1st	6	-raw -fft -mah	22	7	75.86

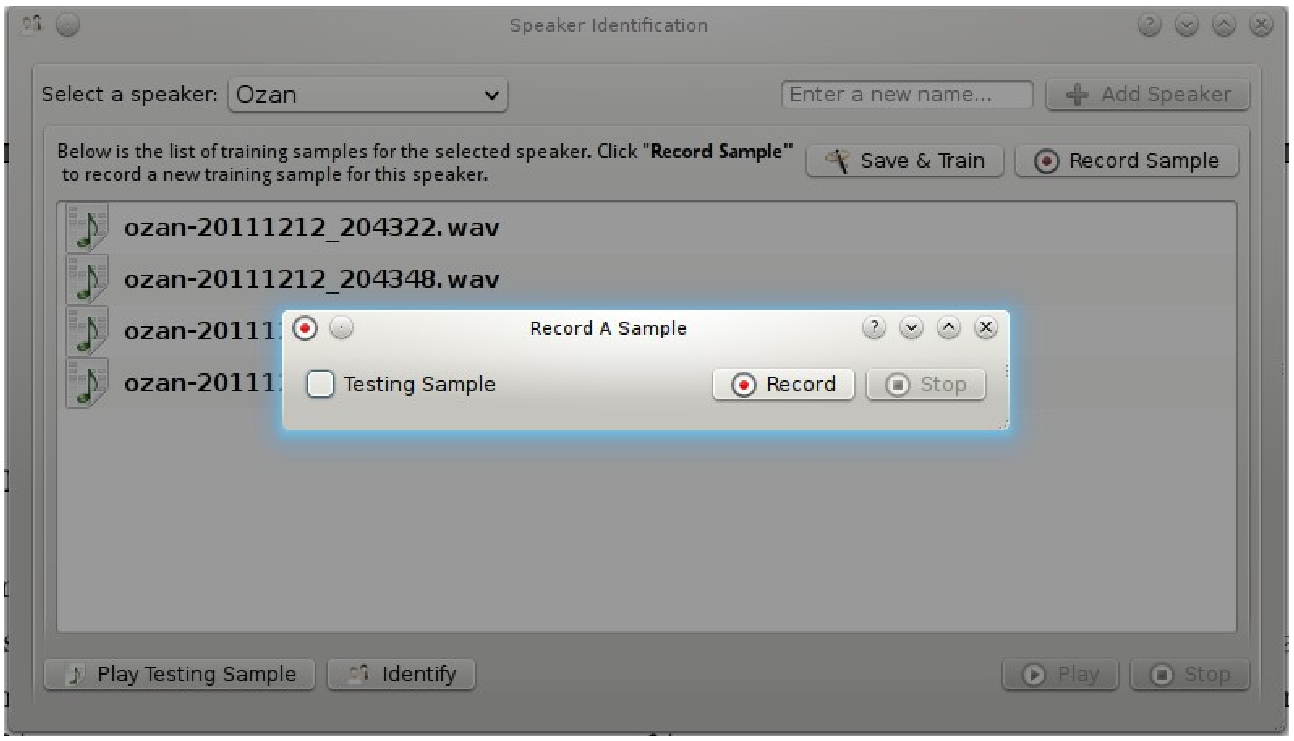
That is why I choosed those 3 parameters during the development of this tool. But you can always change the switches easily by editing the `Marf.py` Python source file.

8. Screenshots



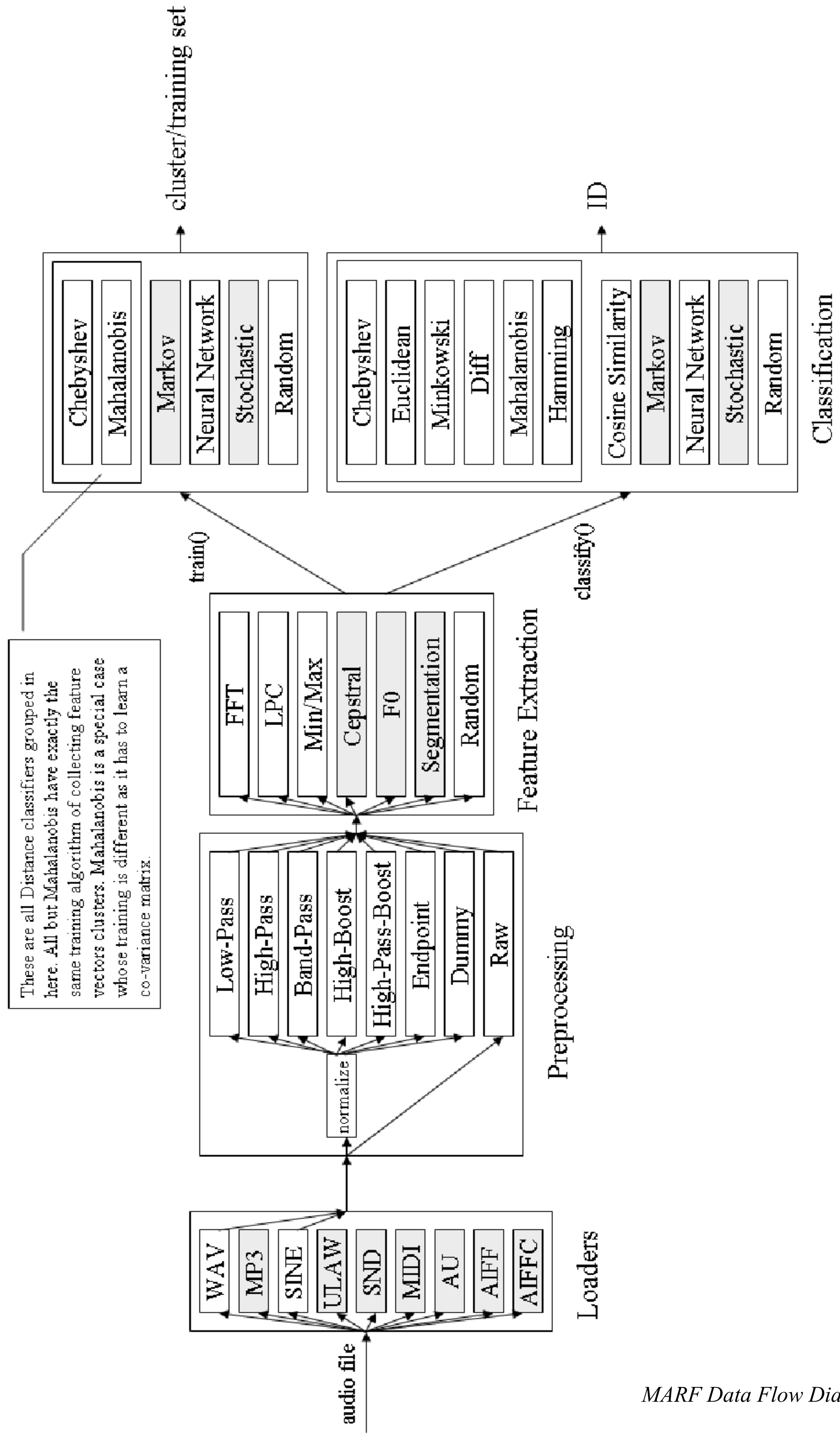
Below is the list of training samples for the selected speaker. Click "**Record Sample**" to record a new training sample for this speaker.

ozan-20111212_204322.wav
ozan-20111212_204348.wav
ozan-20111212_204408.wav
ozan-20111212_204429.wav



Below is the list of training samples for the selected speaker. Click "**Record Sample**" to record a new training sample for this speaker.

ozan-20111212_204322.wav
ozan-20111212_204348.wav
ozan-20111
ozan-20111



These are all Distance classifiers grouped in here. All but Mahalanobis have exactly the same training algorithm of collecting feature vectors clusters. Mahalanobis is a special case whose training is different as it has to learn a co-variance matrix.

MARF Data Flow Diagram