UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

MASTER TEHNOLOGII MULTIMEDIA

SECURE AUTOMATIC SPEAKER VERIFICATION SYSTEM

Martina Alexandra SPAN, II TM

Coordonator: Ph.D. Eng. Mircea GIURGIU

2022

str. Memorandumului nr. 28, 400114 Cluj-Napoca, România
tel. +40-264-401200, fax +40-264-592055, secretariat tel. +40-264-202209, fax +40-264-202280
www.utcluj.ro

***Abstract*— This paper follows the development of a voice recognition application for secured systems. The task is to determine the identity of a person by his/her voice. This is known as Automatic Speaker Recognition and falls in the category of biometric security systems, as it is related to human characteristics or individuality.**
The application is implemented in Python and is composed of two distinct phases, the training phase and testing phase. They can each be assigned to a task, namely the enroll and recognition task.

***Keywords*— Speaker Recognition, Voice Authentication, Feature Extraction, MFCCs, GMM**

I. INTRODUCTION

Voice recognition has become a popular solution in the past few years, expanding its use in a multitude of applications, such as security, biometric authentication, speech recognition and others.

Voice recognition aims to recognize the person speaking, rather than the words themselves. It is mainly classified into speaker verification and speaker identification.

In speaker identification, the voice of a person to be identified is compared to a set of known speakers. The unknown speaker is identified as the speaker whose model best matched the input utterance.

In speaker verification, the voice of a speaker is either accepted or rejected as the voice of a particular person, the target speaker.

Speaker identification can be text-dependent or text-independent. The text-dependent scenario includes the use of a set of words in both enrollment and recognition phase. In text-independent no prior information is considered and the verification of the speaker it's done without constraints on the speech content. This paper focuses on the development of the latter such system.

II. FUNDAMENTALS

Feature extraction is the process of analysis of speech signals. The signals are transformed into feature vectors from where useful relevant information of the speech signal is retained and redundant and irrelevant information are rejected.

Mel-frequency cepstral coefficients (MFCCs) method is one of the most popular strategies for feature extraction in both audio and speech signal.

We strive to make the correct identification of the speaker using the Gaussian mixture model (GMM). The features extracted are fed to GMM-based approaches that have the purpose to create speaker models for identification.

A. MFCCs

MFCC is a technique designed to extract features from an audio signal, mapping the signal onto a non-linear Mel-Scale that mimics human hearing and provides the MFCC feature vectors.

Mel-Frequency suggests the use of the Mel scale. **Ceps**trum, comes from a word play of the word **Spec**trum. And the 'Coefficients' are the values we get out of the features.

Mel scale is established on the human perception of sound. Due to the fact that human perception is non-linear, the distances on this scale increase with frequency. Formula to calculate the estimated mels for a given frequency f in Hz is:

$$mel(f) = 2595 * \log10(1+f/700)$$

The human ear is responsive to both the static and dynamic characteristics of a signal and the MFCC mainly focuses on the static characteristics.

B. *GMM*

Gaussian Mixture Model is widely used in speaker recognition and speaker verification. It is used to model the acoustic features of the individual by capturing the distribution of their speech characteristics.

The Gaussian distribution is defined by its mean and covariance matrix, representing the center and the shape of the distribution.

The features previously extracted as MFCCs are modeled using GMM. In the verification phase, the audio signal is compared with each previously enrolled GMM and a similarity score it's computed. Other notable existing techniques are neural networks, i-vectors and deep learning models.

III. SPEAKER IDENTIFICATION

As established previously in the speaker identification approach no identity is claimed from the speaker. The automatic system it's in charge of determining their identity from a predefined close-set of known speakers.

The identity of the user it's then verified by matching the winner speaker found by the automatic system with the username provided.

A. *Dataset*

In the implementation of this project the 'Speaker Recognition Audio Dataset' provided by Kaggle is used. This dataset contains a number of 50 speakers, each speaker has at least 30 audio wav files, one minute each.

The protocol for speaker verification can be classified in three phases: development, enrollment and evaluation.

B. *Voice Authentication System*

The voice authentication system is composed of two distinct phases, a training phase and a test phase. These phases can be looked at as two tasks, the enroll and recognition task. The main steps of the speaker identification system:

Training phase:

- Data formatting and management
- Extracting MFCC features from the training data
- Training GMMs for each speaker

Testing phase :

- Extracting MFCC features from the testing data
- Scoring the extracted MFCCs against the GMMs
- Recognizing the speaker's based on the scores

IV. IMPLEMENTATION

The audio files we work with have the WAV format.

Pre-processing - the signal is sampled at a frequency of 16KHz for feature extraction, which is the standard data format of automatic speech recognition (ASR). Sampling is the technique used to convert these digital signals into a discrete numeric form. Sampling is done at a certain frequency and it generates numeric signals.

Framing - the input signal it's divided into short frames of 30ms (it can be between

20-30ms), with 15ms overlapping frames for our analysis. Represented in the code as the wst and fpt parameters.

Windowing - refers to splitting the audio signal into temporal segments needed to smooth the effect of using a finite-sized segment for the subsequent feature extraction by tapering each frame at the beginning and end edges. Typically the number of samples (N) in FFT must be an integer power of 2.

FFT window size samples = sr * wst (window size seconds)

Most popular method to extract spectral features for speech is by the use of perceptually based Mel spaced *filterbank* processing of the Fourier Transformed signal.

1. *Fast Fourier Transform*

FFT is a mathematical concept that can decompose the signal into a sinusoidal signal and bring out the individual frequencies. It is a commonly used algorithm for DFT (Discret FT), applied on each frame to calculate the frequency spectrum, called STFT(Short-Time Fourier-Transform).

2. *Mel Filter Bank*

Each filter in the filter bank is a triangular filter having a magnitude of 1 at the center frequency and decreasing linearly towards 0 till it reaches the center frequencies of the two adjacent filters.

Number of filters in the filterbank is represented as ‘nbands’ in the code and it’s set at 20 (default 40). MFCCs are computed using filter-banks (designed to mimic the perception of speech) by applying a DCT retaining a number of the resulting coefficients while the rest are discarded.

3. *Logarithm*

Humans do not hear sound in a linear scale so the next step is to compute the logarithm of the magnitude spectrum. This is obtained by converting DFT values into one value.

4. *Discrete Cosine Transform*

MFCC uses Discrete Cosine Transform (DCT) to calculate coefficients. DCT is a linear transformation used to eliminate the redundancy and duplicity of audio information.

Due to overlapped filter banks the filter energies are correlated. DCT is computed to decorrelate the filter bank energies. The cepstral values from second to fourteenth is taken and 13 Coefficient will represent information regarding vocal tract features.

5. *Deltas and Double Deltas*

The first derivative MFCC features can be extracted as Delta (differential) - changes in the cepstral features with time.

By taking the derivative of Delta features, Double Delta (acceleration) coefficients are extracted. Includes even longer temporal context.

MFCC

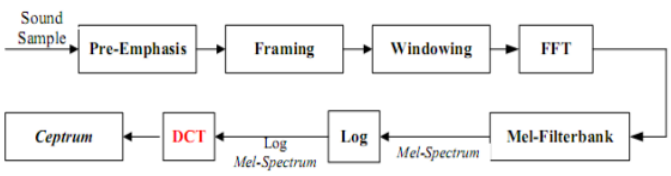

Fig. 1 Block diagram of MFCC

As we talked about each step in particular the extraction of the MFCCs is as follows:

- Take the Fourier transform of (a windowed excerpt of) a signal.
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows (filter-banks).
- Take the logs of the powers at each of the mel frequencies.
- Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
- The MFCCs are the amplitudes of the resulting spectrum.

GMM

After we extract the features vectors from the training data set, we stove to model a denser representation of the acoustic signal for each speaker.

The GMM is a probabilistic model that represents a probability distribution as a weighted sum of Gaussian component distributions (density). The distribution of feature vectors extracted from a person's speech is modeled by a Gaussian mixture density.

We use the GMM implementation from the scikit-learn library.

```
gmm = mixture.GaussianMixture(
n_components=10, max_iter=100,
covariance_type='diag', n_init=3)
```

The parameters consist of a mean, the diagonal of the covariance matrix and a weight.

- n_components: specifies the number of Gaussian components in the GMM (Options: 'spherical', 'tied', 'diag', 'full')
- max_iter: Iteration of the process (default 100)
- covariance_type: These experiments were performed on GMM with diagonal covariance matrix
- n_init: Number of initializations to perform. the best results is kept (default 1)

The EM (Expectation Maximization) algorithm is used to estimate GMM parameters. The aim of Maximum Likelihood Estimation is to find the model parameters which maximize the likelihood of GMM given the training data.

The model for each speaker is saved under the name of the speaker in the models folders from where all the models are retrieved and compared one by one with the features extracted from the audio file in the testing phase.

For a reference number of M speakers, the objective is to find the speaker model which has the maximum posterior probability for the input feature vector sequence.

```
scores                                  =
np.array(gmm.score(feature_vector))
```

Compute the per-sample average log-likelihood of the given data X. Where x is the list of n_features-dimensional data points. Each row corresponds to a single data point.

V. ACCURACY, PRECISION, RECALL

To evaluate the performance of the speaker identification system we use a Confusion Matrix. This is a useful machine learning method that allows us to measure recall, precision and accuracy. The matrix scheme is presented in the next figure.

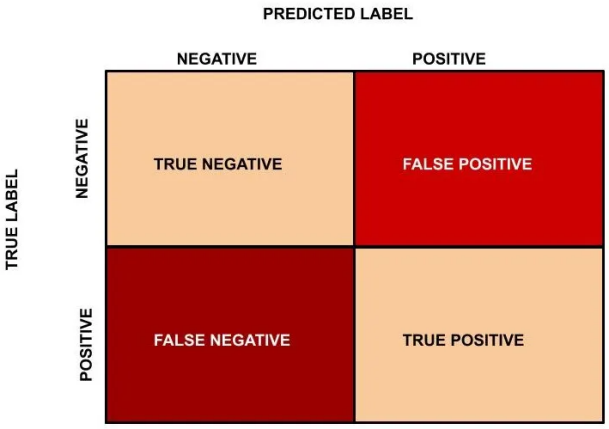

Fig. 2 Confusion Matrix

Where:
TP - True Positive
TN - True Negative
FP - False Positive
FN - False Negativa

The accuracy of the model is obtained by applying the following formulas:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In the next table we can see the number of speakers and the number of audio files, each of them one minute long, used both in training and testing the model.

In the case the model it's trained for only five speakers the accuracy of the model is of 1, which represents the maximum. Even a singular audio file, given its length, performs well under such circumstances.

No speakers	Train	Test	Accuracy	Precision	Recall	F1 score
5	1	10	1	1	1	1
5	10	10	1	1	1	1
10	1	10	1	1	1	1
10	10	10	0.84	0.80	0.84	0.80
20	5	10	0.93	0.91	0.93	0.91
20	10	10	0.89	0.88	0.89	0.86

We can observe that at some point the incresing number of audio files provided for training decressease the performance of the model.

The Convusion Matrix of the test done on a number of 20 users with 5 audio files for training and 10 for testing available in Fig. 3

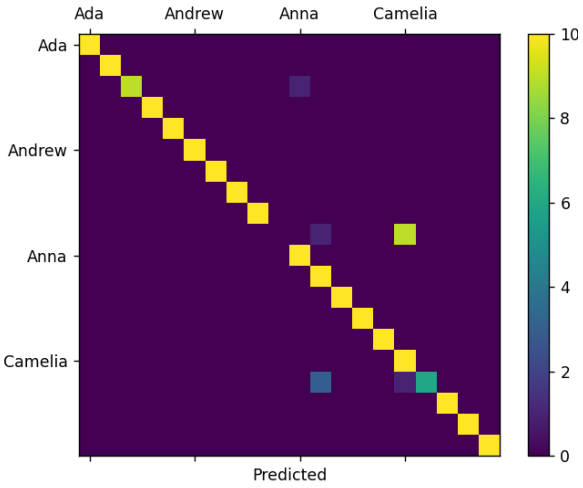

Fig. 3 CM 20 users

According to the experiments, the performance of the method is very good. To improve the speaker recognition it is important to maximize the use of the speaker data (training data), that translated into maximizing the size (no. of components) of the model.

Due to the complexity of the model (models dimensions - too large) and the length of the audio files we are confronted with the problem of the reduced performance of the speaker recognition system.

VI. CONCLUSION

This article focuses on the performance of a text-independent speaker identification system using MFCC and GMM.

The study used Python for implementation and the data used was withdrawn from the 'Speaker Recognition Audio Dataset' provided by Kaggle. The data set consists of speech signals sampled at a frequency of 16KHz and of one minute duration.

The system is composed of two distinct phases - the train and test phase. We are also able to see how the models perform using the Confusion Matrix from where we look over the accuracy, precision, recall and F1-score of our system.

BIBLIOGRAPHY

[1] https://towardsdatascience.com/a-step-by-step-guide-to-speech-recognition-and-audio-signal-processing-in-python-136e37236c24
[2] http://www.enggjournals.com/ijet/docs/IJET17-09-03-513.pdf
[3] https://www.creative-mathematics.cunbm.utcluj.ro/wp-content/uploads/2007_vol_16/creative_2007_16_124_129.pdf
[4] https://courses.csail.mit.edu/6.857/2016/files/31.pdf
[5] https://www.researchgate.net/publication/324031666_Voice_Biometric_A_Technology_for_Voice_Based_Authentication
[6] https://www.destinationcrm.com/Articles/Editorial/Magazine-Features/Voice-Biometrics-Are-Not-100-Percent-Foolproof-but-Steadily-Improving-152171.aspx
[7] https://arxiv.org/pdf/2012.00931.pdf
[8] https://medium.com/analytics-vidhya/speaker-identification-using-machine-learning-3080ee202920
[9] https://www.researchgate.net/publication/222682226_An_Overview_of_Text-Independent_Speaker_Recognition_from_Features_to_Supervectors
[10] https://www.researchgate.net/publication/284149919_Speaker_Identification/fulltext/568822a308aebccc4e155a45/Speaker-Identification.pdf
[11] https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd
[12] https://www.mdpi.com/2224-2708/10/4/72/pdf