Speaker Recognition

Abraham W. Zewoudie, Post-Doctoral Researcher

Department of Signal Processing and Acoustics

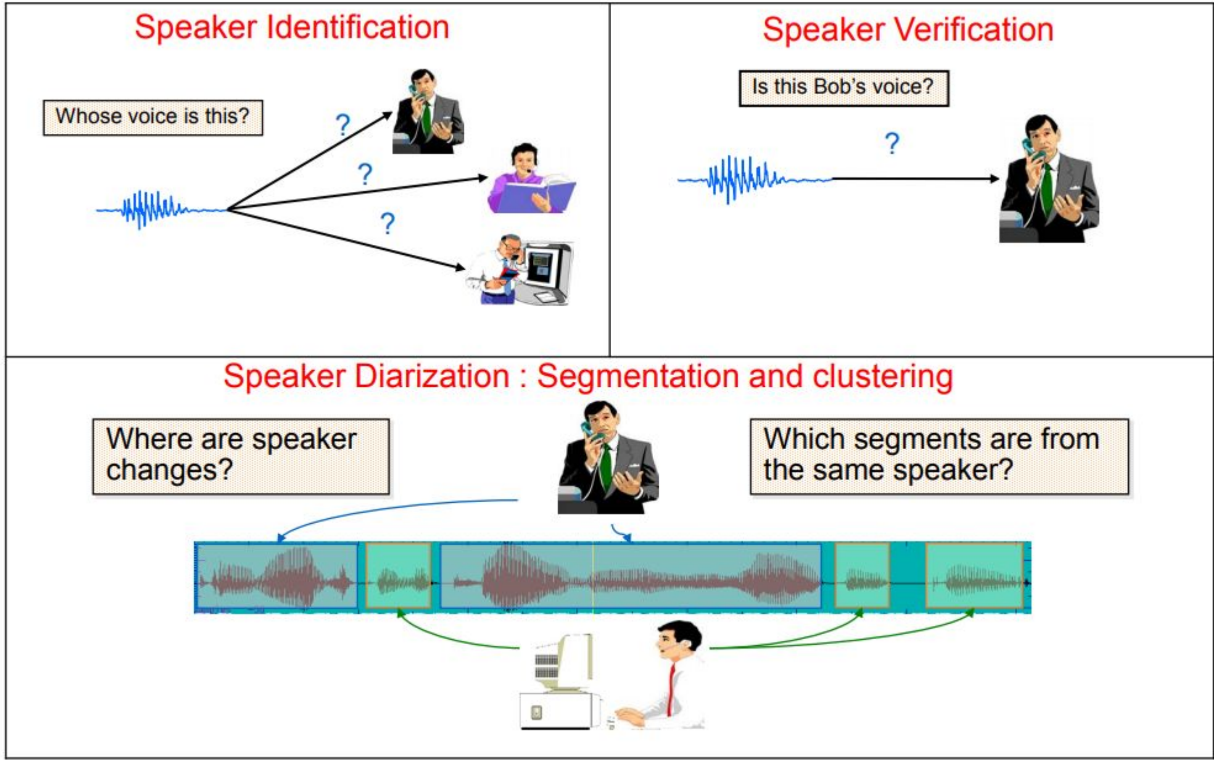
Aalto University

September 30, 2021

Outline

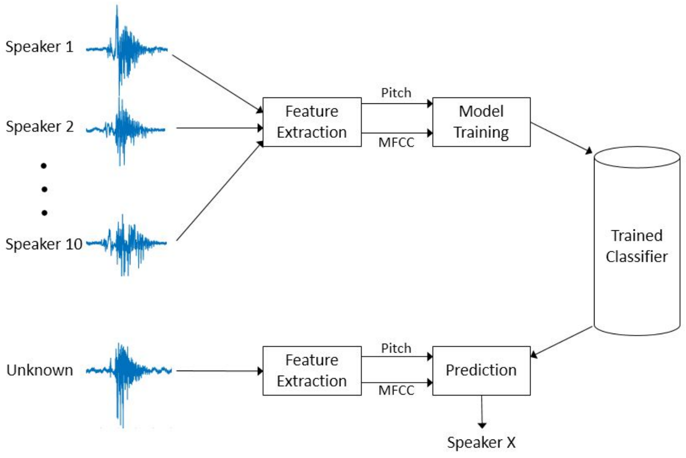
- Overview of Speaker Recognition
- State-of-the-art in Speaker Recognition
  - GMM-UBM
  - i-Vector
  - DNN
- Application Areas
- Performance Evaluations

Speaker Recognition



Steps of Speaker Recognition

1. Feature Extraction: Used features include MFCC, Spectrogram…..
2. Speaker Modeling : The extracted features are used to generate models corresponding to each speaker and stored for comparison during testing. Speaker modeling techniques: GMM, i-Vector and Deep Neural Network (DNN).
3. Classification: Relative scores are computed for each of the speaker models and the one with the highest score is identified to be the test speaker. Scoring methods include Log Likelihood Ratio, Cosine Distance Scoring and Probabilistic Linear Discriminant Analysis (PLDA).

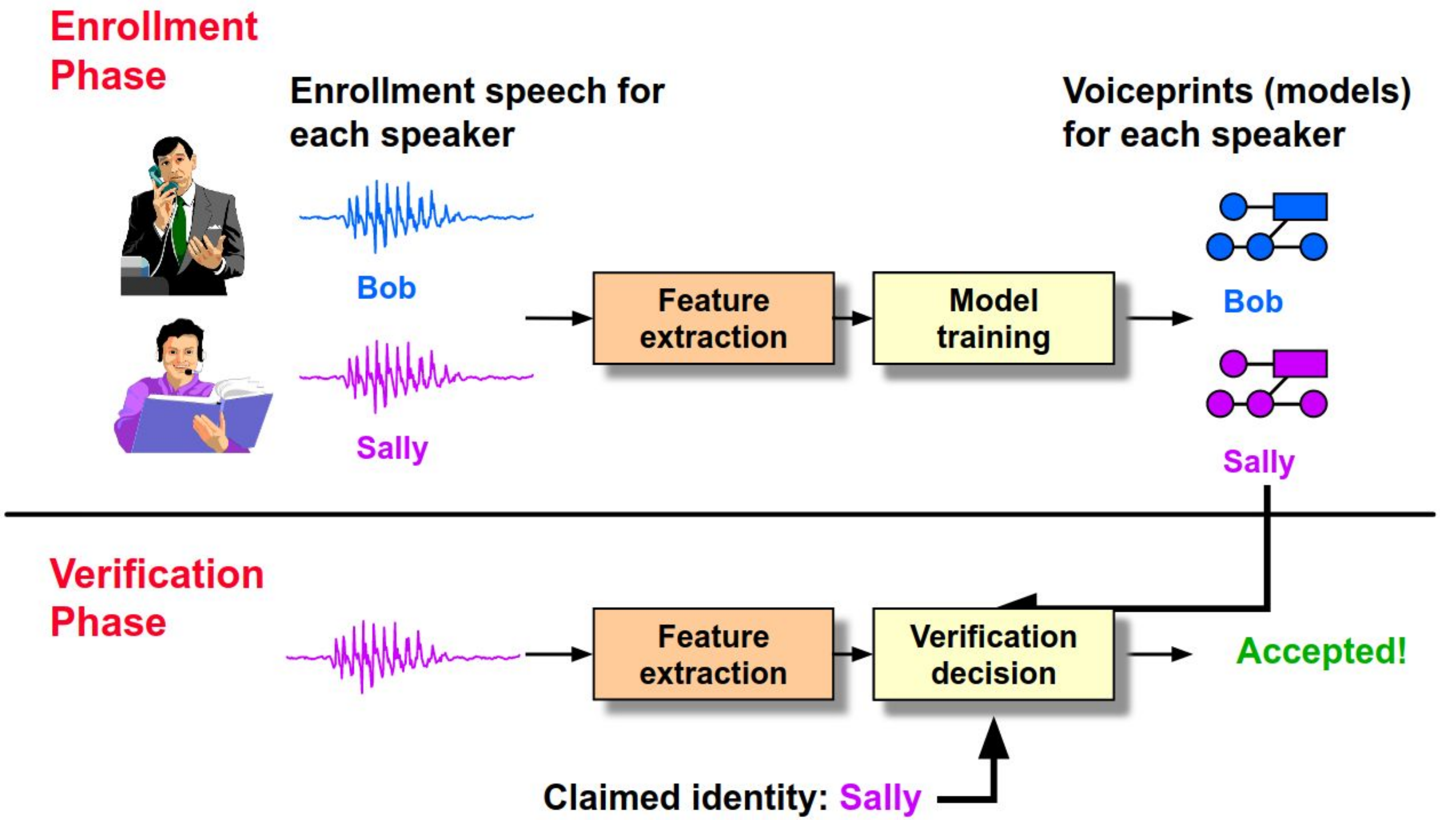


Speech Modalities

- **Text-dependent Speaker Recognition**
  - Recognition system knows text spoken by person
  - Examples: fixed phrase, prompted phrase
  - Used for applications with strong control over user input

- **Text-independent Speaker Recognition**
  - Recognition system does not know text spoken by person
  - Examples: User selected phrase, conversational speech
  - Used for applications with less control over user input
  - More flexible system but also more difficult problem.

Two Phases of Speaker Verification System



Speaker Recognition Challenges

- Speaker verification performance is often degraded in the presence of channel/session variability between enrolment and verification speech signals. Various factors affect channel/session variability:
    - Channel mismatch between enrolment and verification speech signals.
    - Environmental noise and reverberation conditions.
    - The differences in speaker voice (e.g. ageing, health, speaking style and emotional state)
    - Transmission channel (e.g. landline, mobile phone, microphone and voice over Internet protocol (VoIP)).

- Various channel compensation techniques:
    - cepstral mean subtraction (CMS)
    - feature warping
    - cepstral mean variance normalization.

Outline

- Overview of Speaker Recognition
- State-of-the-art in Speaker Recognition
  - GMM-UBM
  - i-Vector
  - DNN
- Application Areas
- Performance Evaluations

GMM-UBM Approach

- A GMM is a weighted sum of M Gaussian densities as given by:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i g(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

where x is a D-dimensional feature vector, i is the index of the i^{th} Gaussian mixture, $g(x|\mu_i, \Sigma_i)$ are Gaussian mixtures.

- As the amount of the enrollment data for each speaker is usually few, it is not so efficient to train a GMM for each speaker from scratch.

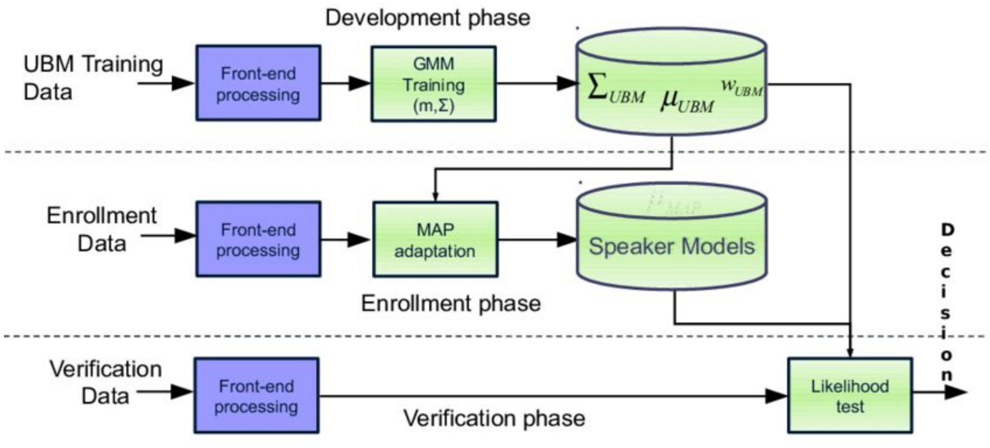
- Therefore, a global GMM, which is referred to as Universal Background Model (UBM), is first trained using a large number of utterances, and then the UBM is adapted to each speaker.

- The adaptation is typically performed using the Maximum a Posteriori (MAP) estimation which includes two steps.

- Maximum a Posteriori (MAP) estimation includes two steps.
    - Sufficient statistics, which are known as Baum-Welch statistics, are calculated given the new feature vectors.
    - The adapted parameters are obtained by the combination of the new statistics for a given speaker and the UBM parameters

Steps of GMM-UBM Approach

- A Universal Background Model (UBM) is first generated using speech samples from all the different speakers.
- MAP (Maximum A Posteriori) estimation is used to obtain models for each of the individual speakers.
- For testing, the feature vectors are extracted from test signal and are compared against all the speaker models in the database
- The model with the highest log likelihood ratio (LLR) is chosen to be the test speaker.



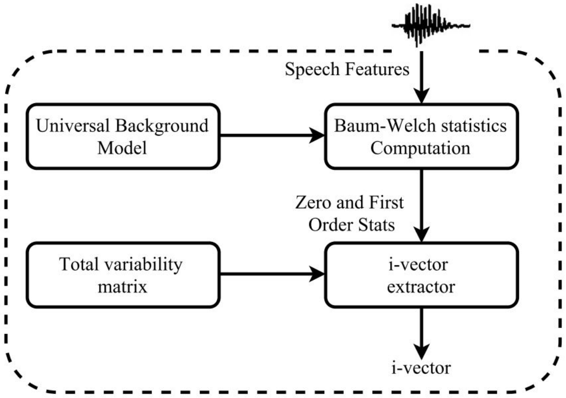
$$LLR = \sum_{t=1}^{T}(log\{p(\vec{x}_t|\lambda_{ref})\} - log\{p(\vec{x}_t|\lambda_{UBM})\})$$

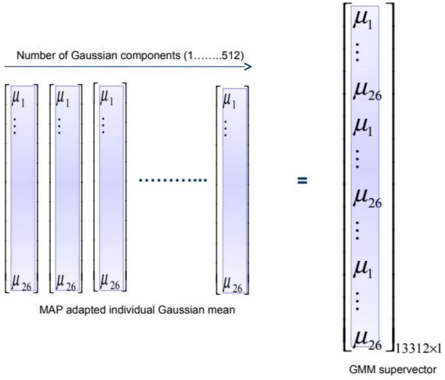
i-Vectors

- A significant contributor to the performance degradation of traditional GMM-UBM speaker verification is the presence of session variability between the training and testing conditions.
- Different approaches have been developed recently to improve the performance of speaker recognition systems. The most popular ones were based on GMM-UBM.
- In i-Vector approach, a given speech recording is represented by a new vector, called total factors as it contains the speaker and channel variabilities simultaneously.
- Speaker recognition based on the *i-vector/x-vector framework* is currently the state-of-the-art in the field.
- Given an utterance, the speaker and channel dependent GMM supervector, M, is defined as follows:

$$M = m + T w$$

m is the speaker and channel independent background UBM super-vector
T is the total variability matrix
w is the extracted i-vector





i-Vector Scoring

- The i-vector based speaker recognition is implemented using two types of classifiers - Cosine distance and Probabilistic Linear Discriminant Analysis (PLDA).

- Both Cosine distance and PLDA make use two different ways for computing the likelihood scores and in either case, the speaker model with the highest score is identified to be the speaker.

- Cosine distance directly compares two inputs and gives out the degree of similarity between them.

- Given two i-vectors w_1 and w_2, PLDA computes the likelihood ratio of the two i-vectors as follows:

$$Score(w_1, w_2) = \frac{p(w_1, w_2|H_1)}{p(w_1|H_2)p(w_2|H_2)}$$

where H_1 indicates that both i-vectors belong to the same speaker and H_2 indicates they belong to two different speakers.

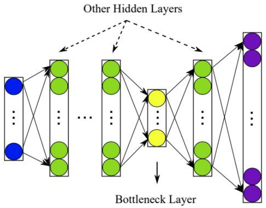
- In PLDA, assuming that the training data consists of J i-vectors where each of these i-vectors belong to speaker I, the j'th i-vector of the I'th speaker is denoted by:

$$w_{ij} = \mu + Fh_i + Gy_{ij} + \Sigma_{ij}$$

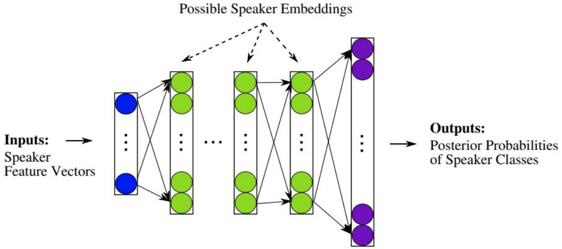
μ is the overall speaker and segment independent mean of the i-vectors in the training dataset
F is the eigen voice matrix (speaker variability)
G is the eigen channel matrix (within-speaker).
Σ_{ij} represents any unexplained data variation.
h_i are the speaker factors and y_{ij} are channel factors.

DNN

- The traditional i-vector approach consists in three main stages: *Baum-Welch statistics computation*, *i-vector extraction*, and PLDA backend.
    - Using i-Vector as input to the network
    - Using bottleneck features as input to the network.



    - Using speaker embeddings (i.e., the speaker characteristics of a speech signal with a single low dimensional vector).



Outline

- Overview of Speaker Recognition
- State-of-the-art in Speaker Recognition
  - GMM-UBM
  - i-Vector
  - DNN
- Application Areas
- Performance Evaluations
- Application Areas

Application Areas

1. **Speaker Recognition for Authentication**
    - It allows users to identify person using their voices.
    - ***Voice sample*** *vs* ***PIN/credit card*** *(lost/stolen) vs* ***PIN or password*** *(forgotten)*

2. **Speaker Recognition for Surveillance**
    - *Security agencies have several means of collecting information.*
    - *One of these is* ***electronic eavesdropping*** *of* ***telephone*** *and* ***radio*** *conversations.*
    - *As these results in high quantities of data, filter mechanism must be applied to find the relevant information.*
    - *One of these filters could be recognition of target speakers that are of interest for the service.*

3. **Forensic Speaker Recognition**
    - *If there is a speech sample that was recorded* ***during a crime****, the suspect's voice can be compared with this to find the similarity of two voices.*

4. **Security**
    - *It is the most obvious application of any biometric authentication applications.*
    - *Examples:* ***Access control****,* ***credit card transactions****,* ***banking access***

Outline

- Overview of Speaker Recognition
- State-of-the-art in Speaker Recognition
  - GMM-UBM
  - i-Vector
  - DNN
- Application Areas
- Performance Evaluations
- Application Areas

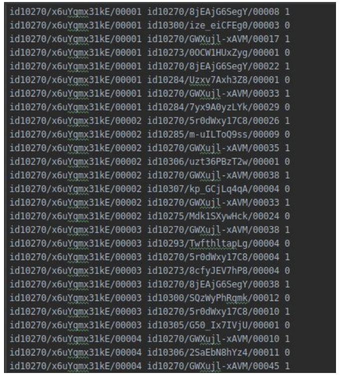
Performance evaluations

- False Acceptance Rate (FAR): It occurs when the speech segments from an impostor speaker are falsely accepted as a target speaker by the system.

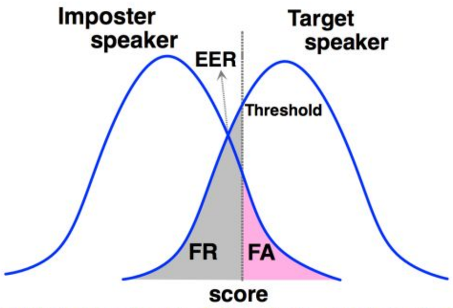
$$\mathrm{FAR} = \frac{\text{Total number of false acceptance errors}}{\text{Total number of imposter speaker attempts}}$$

- False Rejection Rate (FRR): A false rejection occurs when the target speaker is rejected by the verification systems.

$$\mathrm{FRR} = \frac{\text{Total number of false rejection errors}}{\text{Total number of enrolled speaker attempts}}$$

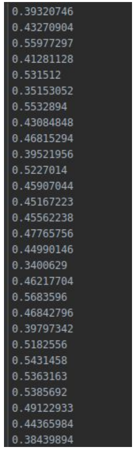


```
id10270/x6uYqmx31kE/00001 id10270/8jEAjG6SegY/00008 1
id10270/x6uYqmx31kE/00001 id10300/ize_eiCFEg0/00003 0
id10270/x6uYqmx31kE/00001 id10270/GWXujl-xAVM/00017 1
id10270/x6uYqmx31kE/00001 id10273/0OCW1HUxZyg/00001 0
id10270/x6uYqmx31kE/00001 id10270/8jEAjG6SegY/00022 1
id10270/x6uYqmx31kE/00001 id10284/Uzxv7Axh3Z8/00001 0
id10270/x6uYqmx31kE/00001 id10270/GWXujl-xAVM/00033 1
id10270/x6uYqmx31kE/00001 id10284/7yx9A0yzLYk/00029 0
id10270/x6uYqmx31kE/00002 id10270/5r0dWxy17C8/00026 1
id10270/x6uYqmx31kE/00002 id10285/m-uILToQ9ss/00009 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00035 1
id10270/x6uYqmx31kE/00002 id10306/uzt36PBzT2w/00001 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00038 1
id10270/x6uYqmx31kE/00002 id10307/kp_GCjLq4qA/00004 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00033 1
id10270/x6uYqmx31kE/00002 id10275/Mdk1SXywHck/00024 0
id10270/x6uYqmx31kE/00003 id10270/GWXujl-xAVM/00038 1
id10270/x6uYqmx31kE/00003 id10293/TwfthltapLg/00004 0
id10270/x6uYqmx31kE/00003 id10270/5r0dWxy17C8/00004 1
id10270/x6uYqmx31kE/00003 id10273/8cfyJEV7hP8/00004 0
id10270/x6uYqmx31kE/00003 id10270/8jEAjG6SegY/00038 1
id10270/x6uYqmx31kE/00003 id10300/SQzWyPhRqmk/00012 0
id10270/x6uYqmx31kE/00003 id10270/5r0dWxy17C8/00010 1
id10270/x6uYqmx31kE/00003 id10305/G50_Ix7IVjU/00001 0
id10270/x6uYqmx31kE/00004 id10270/GWXujl-xAVM/00010 1
id10270/x6uYqmx31kE/00004 id10306/2SaEbN8hYz4/00011 0
id10270/x6uYqmx31kE/00004 id10270/GWXujl-xAVM/00045 1
```



The main goal for speaker verification must be to minimize those errors.

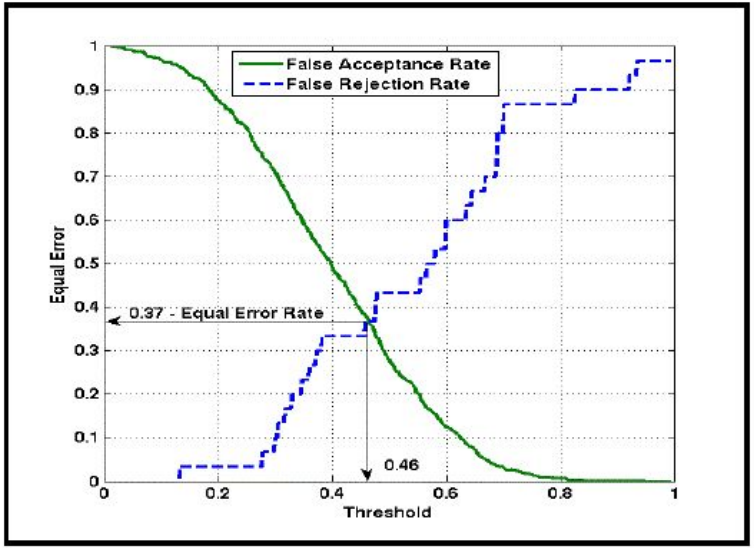
The tradeoff between the errors depend on the application.



```
0.39320746
0.43270904
0.55977297
0.41281128
0.531512
0.35153052
0.5532894
0.43084848
0.46815294
0.39521956
0.5227014
0.45907044
0.45167223
0.45562238
0.47765756
0.44990146
0.3400629
0.46217704
0.5683596
0.46842796
0.39797342
0.5182556
0.5431458
0.5363163
0.5385692
0.49122933
0.44365984
0.38439894
```

Performance evaluations

- The performance metrics of speaker verification systems can be measured using the equal error rate (EER).
- The EER is obtained when the false acceptance rate and false rejection rate are equal.
- The performance of the system improves if the value of ERR is lower because the sum of total error of the false acceptance and the false rejection at the point of ERR decreases



Implementation

1. Read training List

```
id10001/1zcIwhmdeo4/00001 id10001
id10001/1zcIwhmdeo4/00002 id10001
id10001/1zcIwhmdeo4/00003 id10001
id10001/7gWzIy6yIIk/00001 id10001
id10002/0_laIeN-Q44/00001 id10002
id10002/6WO410QOeuo/00001 id10002
id10002/6WO410QOeuo/00002 id10002
id10002/6WO410QOeuo/00003 id10002
..................................
..................................
..................................
id11251/s4R4hvqrhFw/00006 id11251
id11251/s4R4hvqrhFw/00007 id11251
id11251/s4R4hvqrhFw/00008 id11251
id11251/s4R4hvqrhFw/00009 id11251
```

1. Encode target labels with value between 0 and find unique number of classes

[0 0 0 0 1 1 1 1……………………..39 39 39 39]

Number of classes = 40

Implementation

3. Binarize labels in a one-vs-zero fashion

```
[[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

 ...............................................................................
 ...............................................................................
 ...............................................................................

  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]
```

4. Splits the training data into random train and validation subsets (80% vs 20%).

5. Train and save the model.

Implementation

6. We are given a list of trials

```
id10270/x6uYqmx31kE/00001 id10270/8jEAjG6SegY/00008 1
id10270/x6uYqmx31kE/00001 id10300/ize_eiCFEg0/00003 0
id10270/x6uYqmx31kE/00001 id10270/GWXujl-xAVM/00017 1
id10270/x6uYqmx31kE/00001 id10273/0OCW1HUxZyg/00001 0
id10270/x6uYqmx31kE/00001 id10270/8jEAjG6SegY/00022 1
id10270/x6uYqmx31kE/00001 id10284/Uzxv7Axh3Z8/00001 0
id10270/x6uYqmx31kE/00001 id10270/GWXujl-xAVM/00033 1
id10270/x6uYqmx31kE/00001 id10284/7yx9A0yzLYk/00029 0
id10270/x6uYqmx31kE/00002 id10270/5r0dWxy17C8/00026 1
id10270/x6uYqmx31kE/00002 id10285/m-uILToQ9ss/00009 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00035 1
id10270/x6uYqmx31kE/00002 id10306/uzt36PBzT2w/00001 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00038 1
id10270/x6uYqmx31kE/00002 id10307/kp_GCjLq4qA/00004 0
id10270/x6uYqmx31kE/00002 id10270/GWXujl-xAVM/00033 1
id10270/x6uYqmx31kE/00002 id10275/Mdk1SXywHck/00024 0
id10270/x6uYqmx31kE/00003 id10270/GWXujl-xAVM/00038 1
id10270/x6uYqmx31kE/00003 id10293/TwfthltapLg/00004 0
id10270/x6uYqmx31kE/00003 id10270/5r0dWxy17C8/00004 1
id10270/x6uYqmx31kE/00003 id10273/8cfyJEV7hP8/00004 0
id10270/x6uYqmx31kE/00003 id10270/8jEAjG6SegY/00038 1
id10270/x6uYqmx31kE/00003 id10300/SQzWyPhRqmk/00012 0
id10270/x6uYqmx31kE/00003 id10270/5r0dWxy17C8/00010 1
id10270/x6uYqmx31kE/00003 id10305/G50_Ix7IVjU/00001 0
id10270/x6uYqmx31kE/00004 id10270/GWXujl-xAVM/00010 1
id10270/x6uYqmx31kE/00004 id10306/2SaEbN8hYz4/00011 0
id10270/x6uYqmx31kE/00004 id10270/GWXujl-xAVM/00045 1
```

7. We predict using trained model

8. Compute EER using 7 and 8.

```
0.39320746
0.43270904
0.55977297
0.41281128
0.531512
0.35153052
0.5532894
0.43084848
0.46815294
0.39521956
0.5227014
0.45907044
0.45167223
0.45562238
0.47765756
0.44990146
0.3400629
0.46217704
0.5683596
0.46842796
0.39797342
0.5182556
0.5431458
0.5363163
0.5385692
0.49122933
0.44365984
0.38439894
```