Speaker Recognition

Mickael Rouvier

CERI – Avignon Université
mickael.rouvier@univ-avignon.fr

16 septembre 2024

AVIGNON UNIVERSITÉ

Section 1

Présentation sommaire de l'UCE

AVIGNON
UNIVERSITÉ

Plan et objectif du cours

- **Plan du cours**
  - Speaker Recognition
  - Speaker Diarization
  - Automatic Speech Recognition
  - Text To Speech
  - LLM Audio

Organisation

- **Attention** : il y a des choses que je ne dis qu'en cours, donc venez et soyez à l'heure
- Evaluation
  - TP : Coefficient 0.5
  - Examen final : Coefficient 0.5

Section 2

Speaker Recognition

AVIGNON
UNIVERSITÉ

Outline

- **Introduction**
  - Task, Terminology, Framework
- **Neural Embeddings**
  - Speaker embeddings
- **Frame level Architectures**
  - TDNN, ResNet, ECAPA-TDNN
- **Pooling level Mechanisms**
  - Statistics, MHAtt
- **Loss Functions**
  - Cross-Entropy, AM-Softmax, AAM-Softmax
- **Backend**
  - CosineScoring, PLDA, Score-Normalization,Calibration
- **Other Topics in Speaker Recognition**
  - Spoofing Attacks, Adversarial Attacks

Based on the slides of Jesús Villalba titled *Speaker Recognition*.

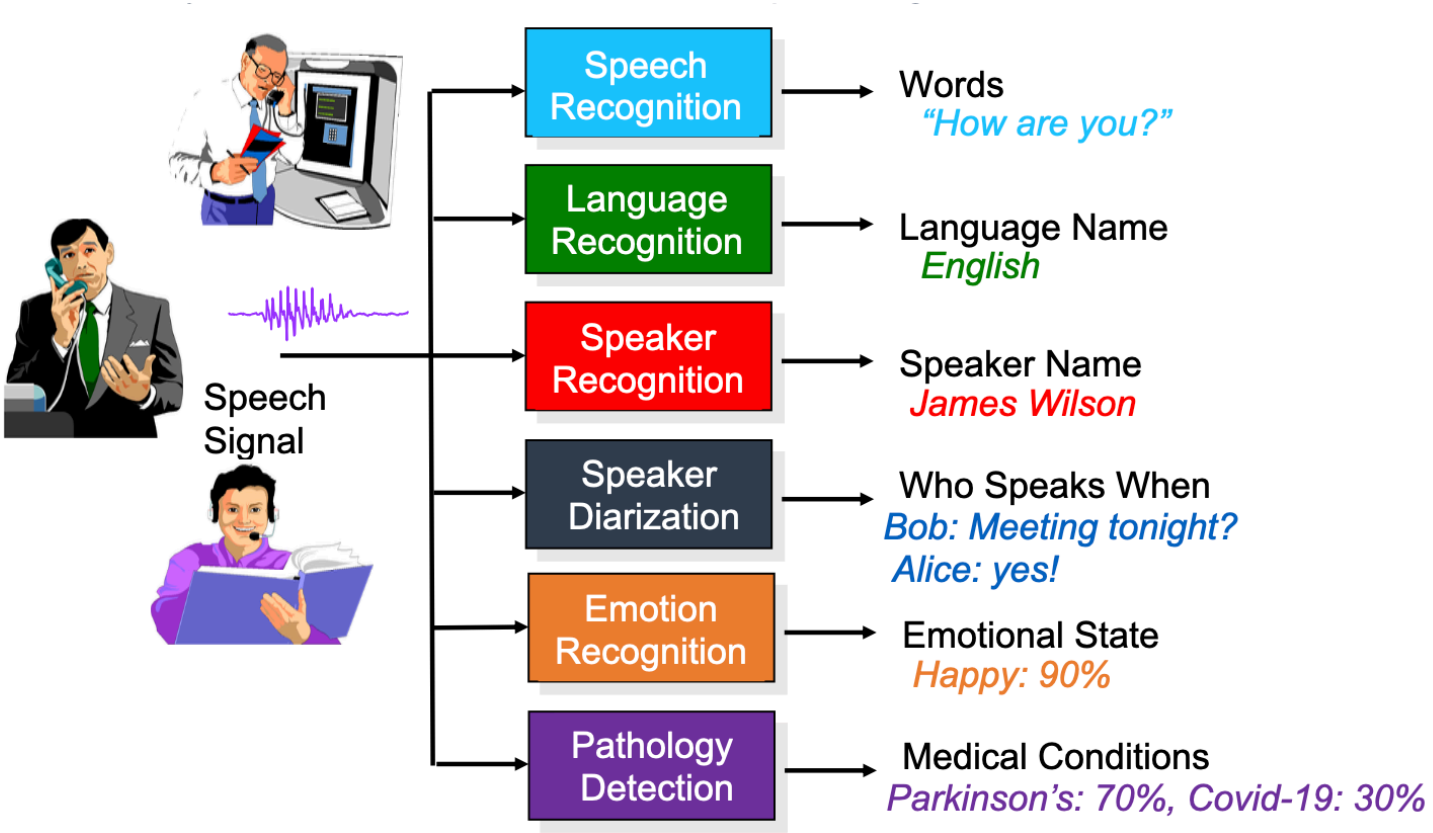
Section 3

Introduction

AVIGNON
UNIVERSITÉ

Extracting Information from Speech

Goal : Extracting Information from Speech



Behaviometrics / Speaker Recognition

- **Behaviometrics :** Behavioral characteristics are related to the pattern of behavior of a person (signature, voice....)
- **Properties of speech are influenced by :**
  - **Anatomy** : Shape and size of voice production organs (vocal tract, larynx, nasal cavity).
  - **Behavioral patterns** : Accent, rhythm, intonation style, pronunciation patterns, vocabulary.
- **Advantages :**
  - **Easy to use** : Speech is a natural way of communication.
  - **Non-intrusive** : Generally well accepted by users.

Speaker Recognition Applications

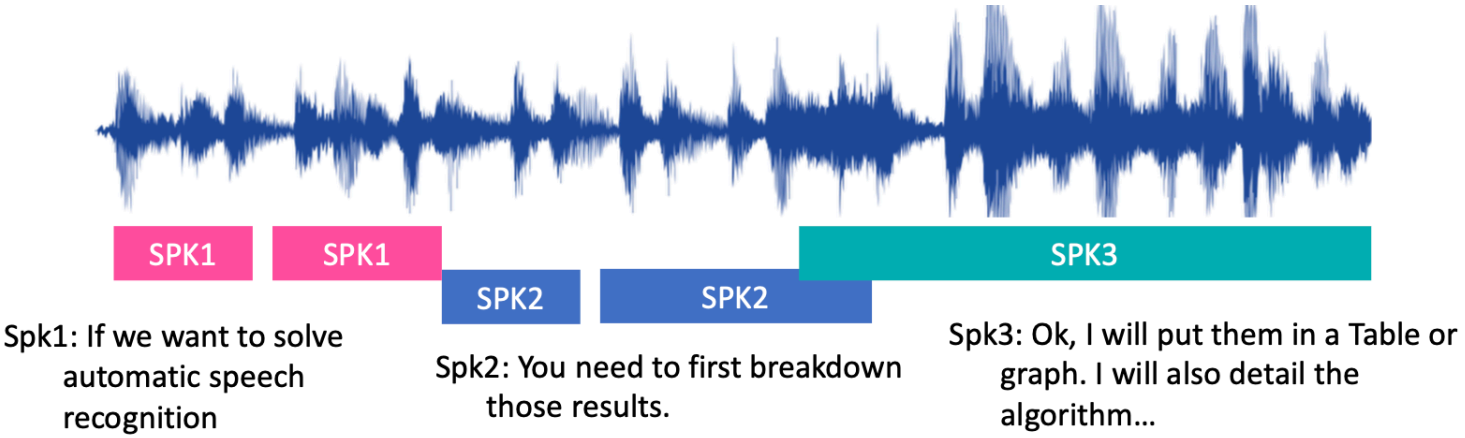
- **Law Enforcement and Forensics**
  - Identifying criminals through analysis of telephone conversations and voice messages.
  - Handling telephone threats by tracing and verifying the speaker's identity.
  - Detecting known fraudsters listed on blacklists.
  - Using voice samples from crime scenes as evidence to convict or exonerate individuals in court.

Speaker Recognition Applications

- **Identity Authentication and Access Control**
  - Access control for high-security physical facilities : military bases, airports, government buildings.
  - Access to computer networks and sensitive web services
  - Password reset processes
  - Enhancing security in telephone and electronic banking systems

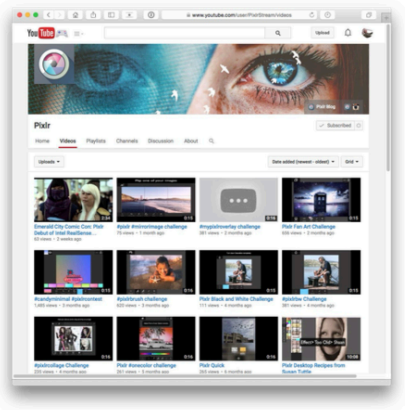
Speaker Recognition Applications

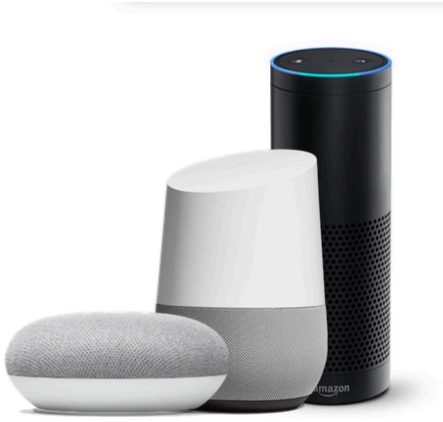
- **Meeting Transcription**
  - Enriching transcripts by adding speaker-specific metadata
  - Identifying speakers to clarify "Who said what ?" in meetings
  - Enhancing the usability of automated meeting summaries for record-keeping and compliance.



Speaker Recognition Applications

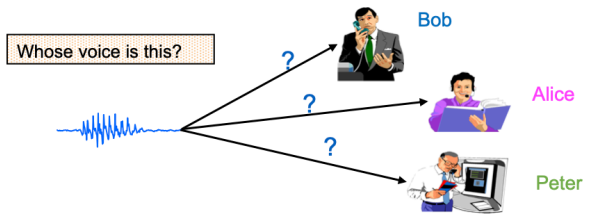
- **Audio-Visual Content Indexing**
  - Platforms : Broadcast Television, Online Videos (e.g., YouTube, Prime Video...)
  - Adding metadata :
    - Who is speaking in this video ?
    - When is he/she speaking ?
  - Improve document indexing and search
- **Customization Features**
  - Voice-Assistant Personalization
    - Tailoring music playlists to user preferences.
    - Enabling voice commands to access personal emails and other private content.
    - Implementing voice-activated parental controls for media consumption.





Speaker Identification

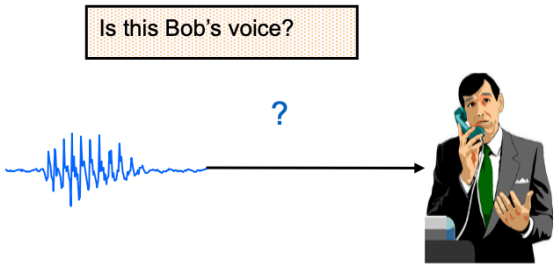
- Determine whether a test speaker matches one of a set of known speakers
- Referred as **closed-set** identification (often used in systems where the possible speakers are already known and cataloged)



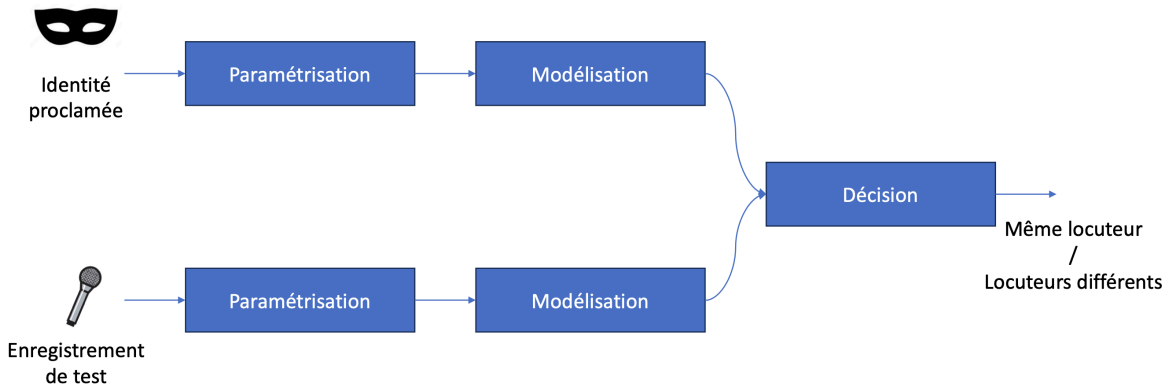
Speaker Verification

- Determine whether a test speaker matches a specific target speaker
- Unknown speech may come from a large set of unknown speakers – referred as open-set verification
- This is most common task in recognition, close to real application

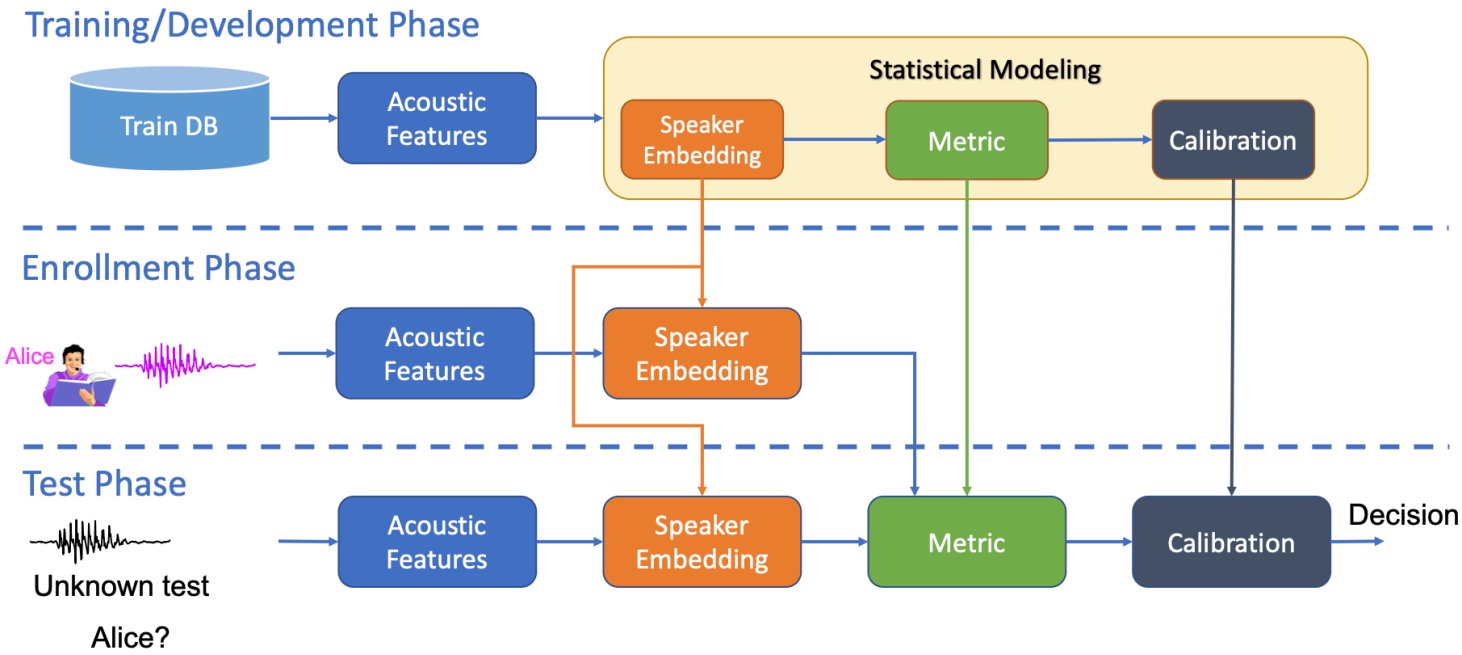
- Tasks :
  - Text-independent : verifying the identity without constraint on the speech content
  - Text-dependent : verifying the identity and the speech content (password)



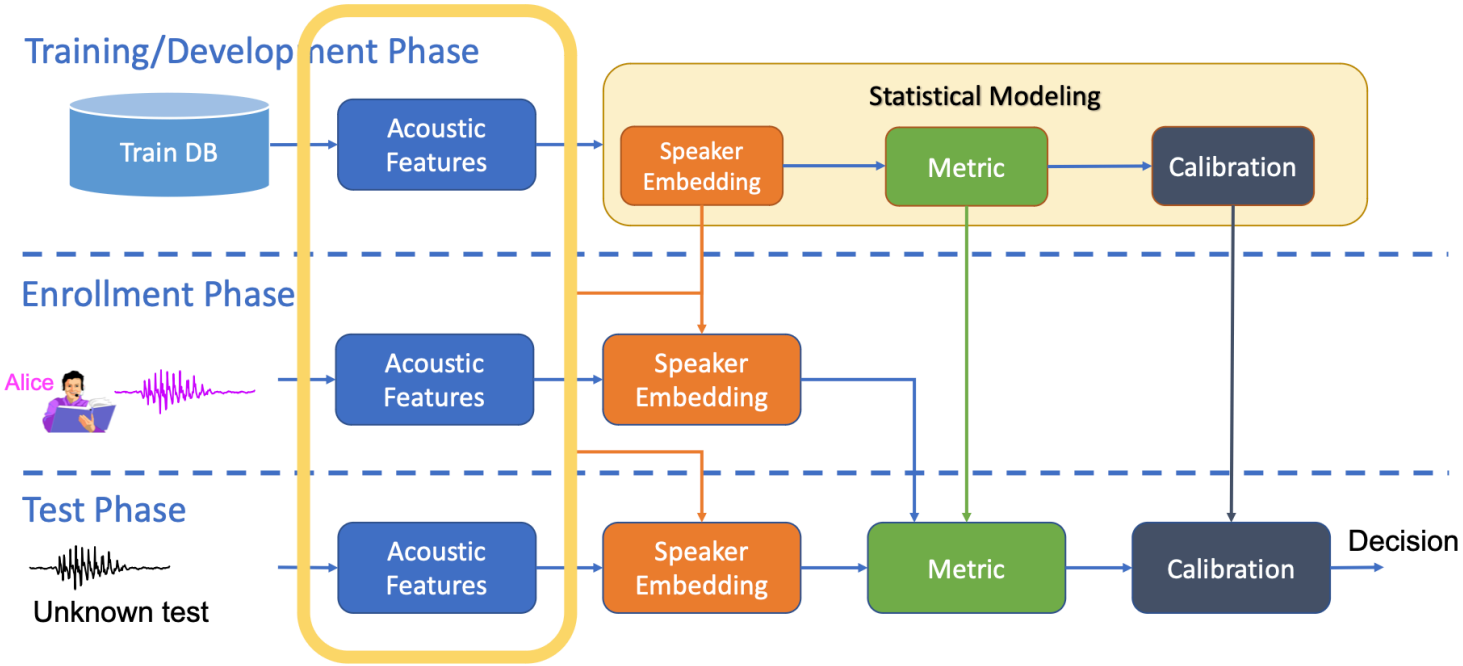
Speaker Verification Architecture



Speaker Verification Pipeline

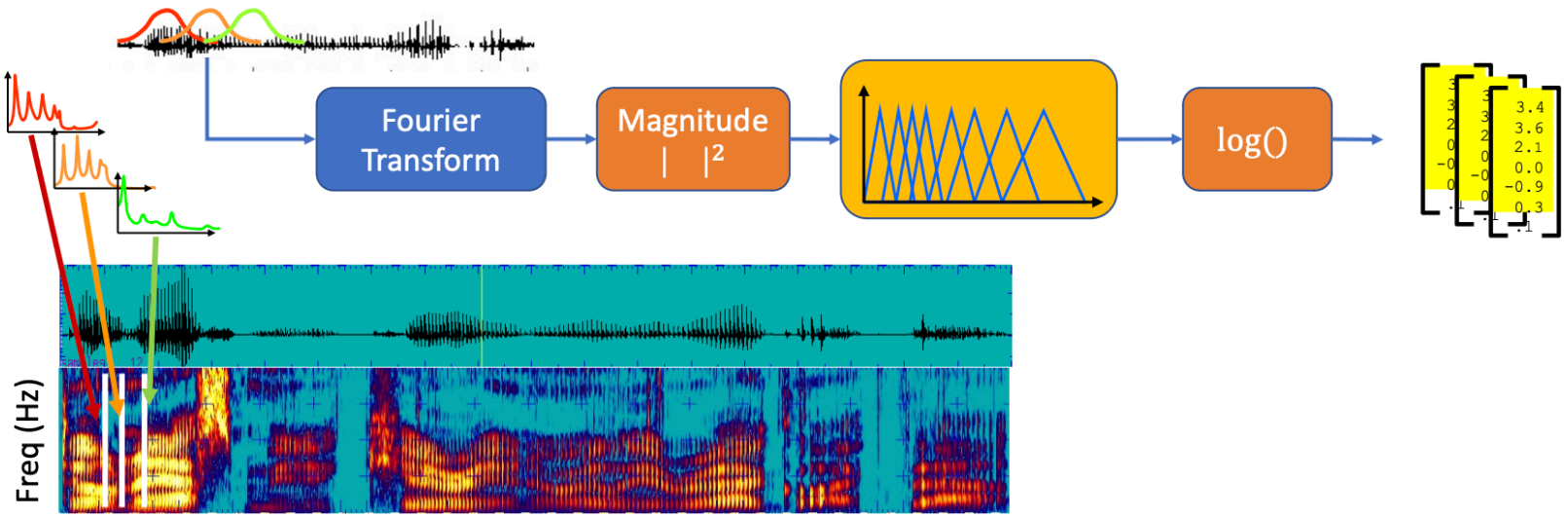


Speaker Verification Pipeline : Acoustic Features

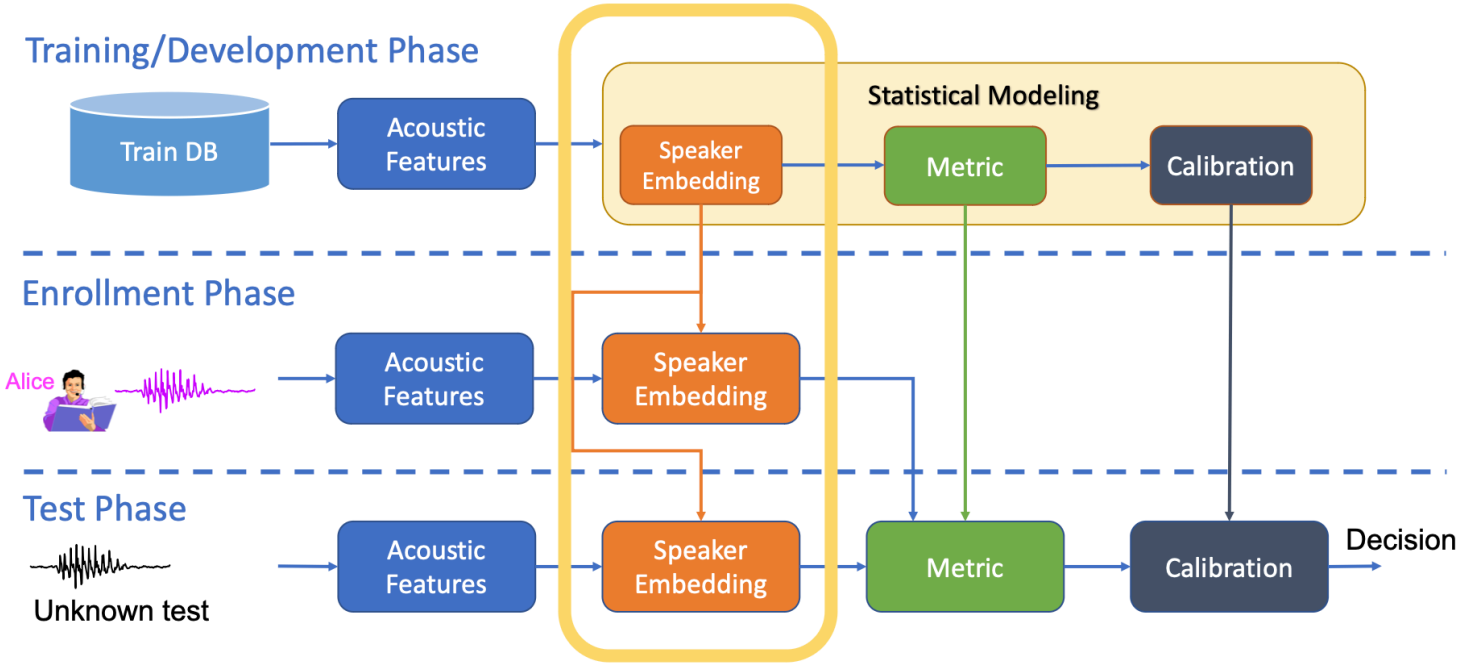


Acoustic Features

- Time sequence of acoustic features is needed to extract the speech information
  - Short-time spectral features are computed using a sliding window
  - Time-frequency representation of the signal
  - Filter bank in log Mel scale (Mel filtered spectrogram)



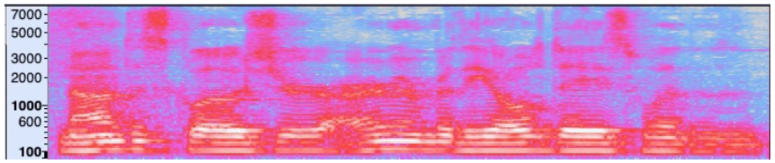
Speaker Verification Pipeline : Speaker Embeddings



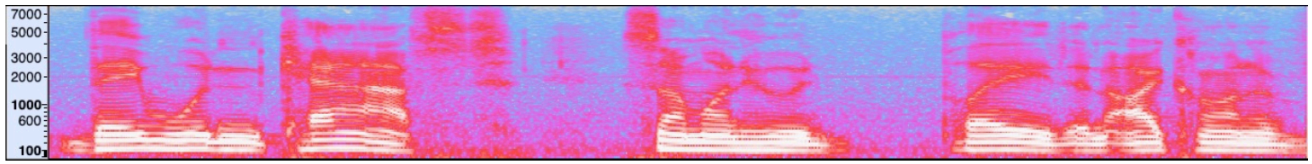
Speaker Embeddings

- **It's difficult to compare Enrollment and Test recordings using Acoustic Features**
- **They have different durations, different number of feature vectors**
  - Cannot calculate something like Euclidean distance between feature matrices
- **The sequence of phonemes differs in each recording**
  - Early systems were text-dependent, requiring the same phrase in both enrollment and test phases

Enroll.:

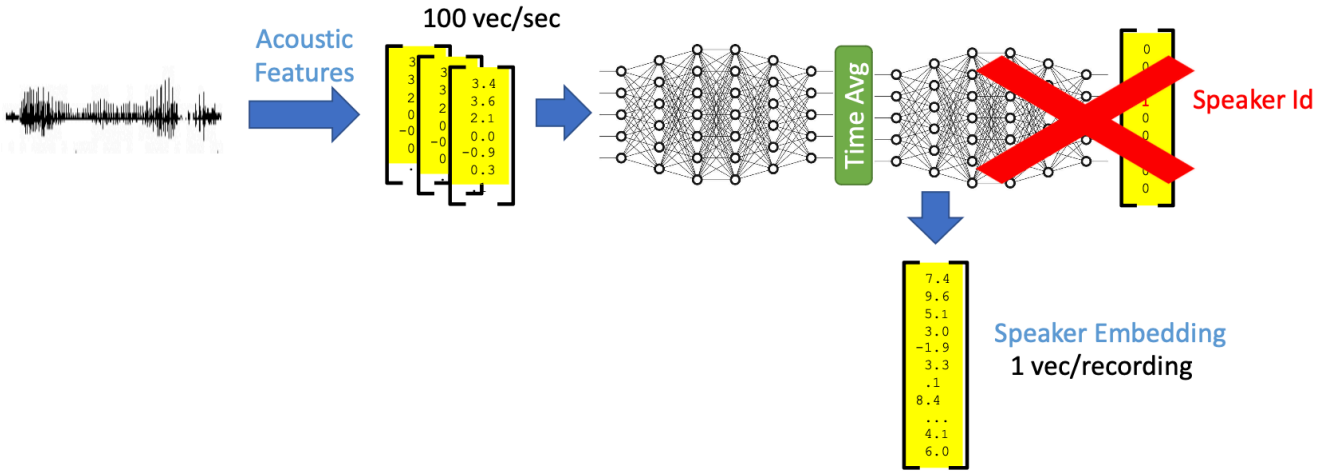


Test:

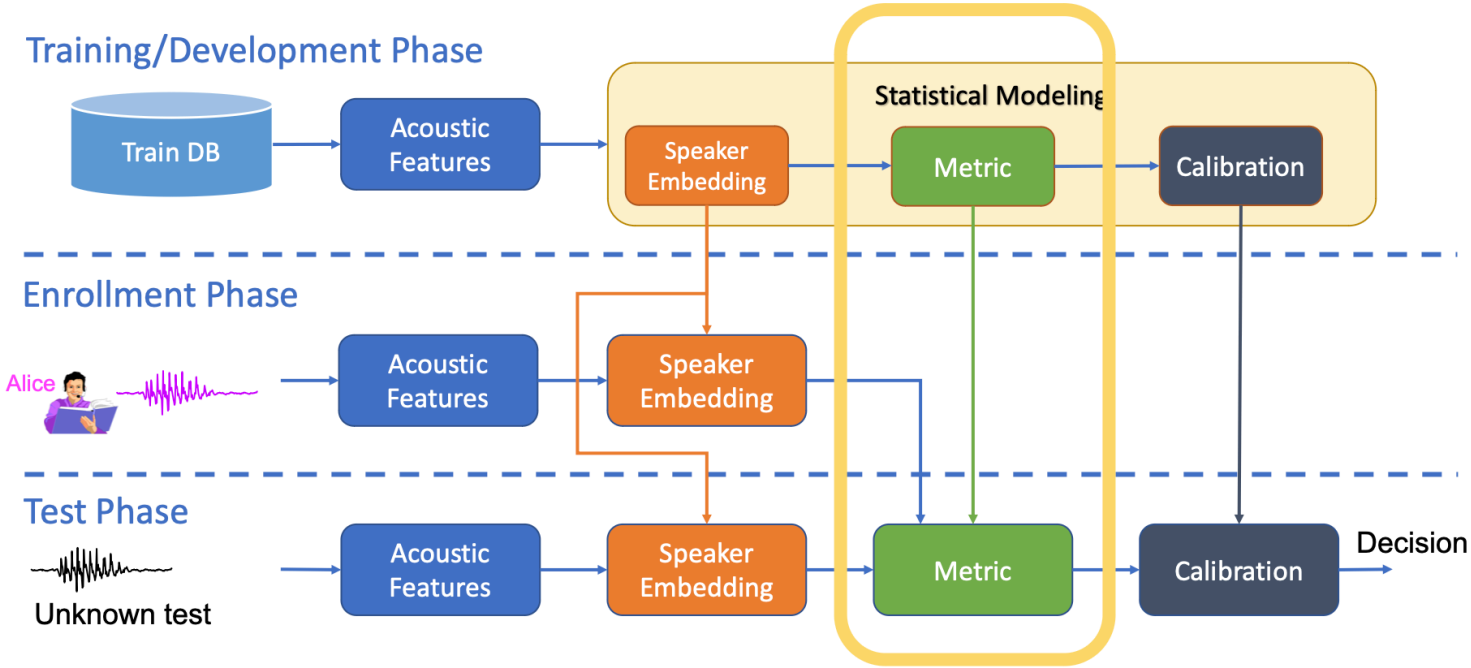


Speaker Embeddings

- **Modern solution : Speaker Embeddings**
  - Transforms variable-length recordings into a single vector called : speaker embedding
  - This embedding retains the speaker's identity information
- **Train the network to classify speakers**
  - Utilizes a large dataset of over thousands of speakers
- **After training**
  - Extract an intermediate layer as the speaker embedding



Speaker Verification Pipeline : Metric

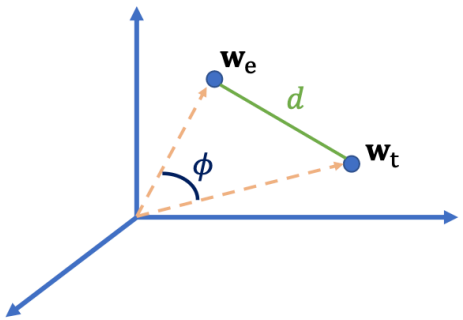


Metric (Back-end)

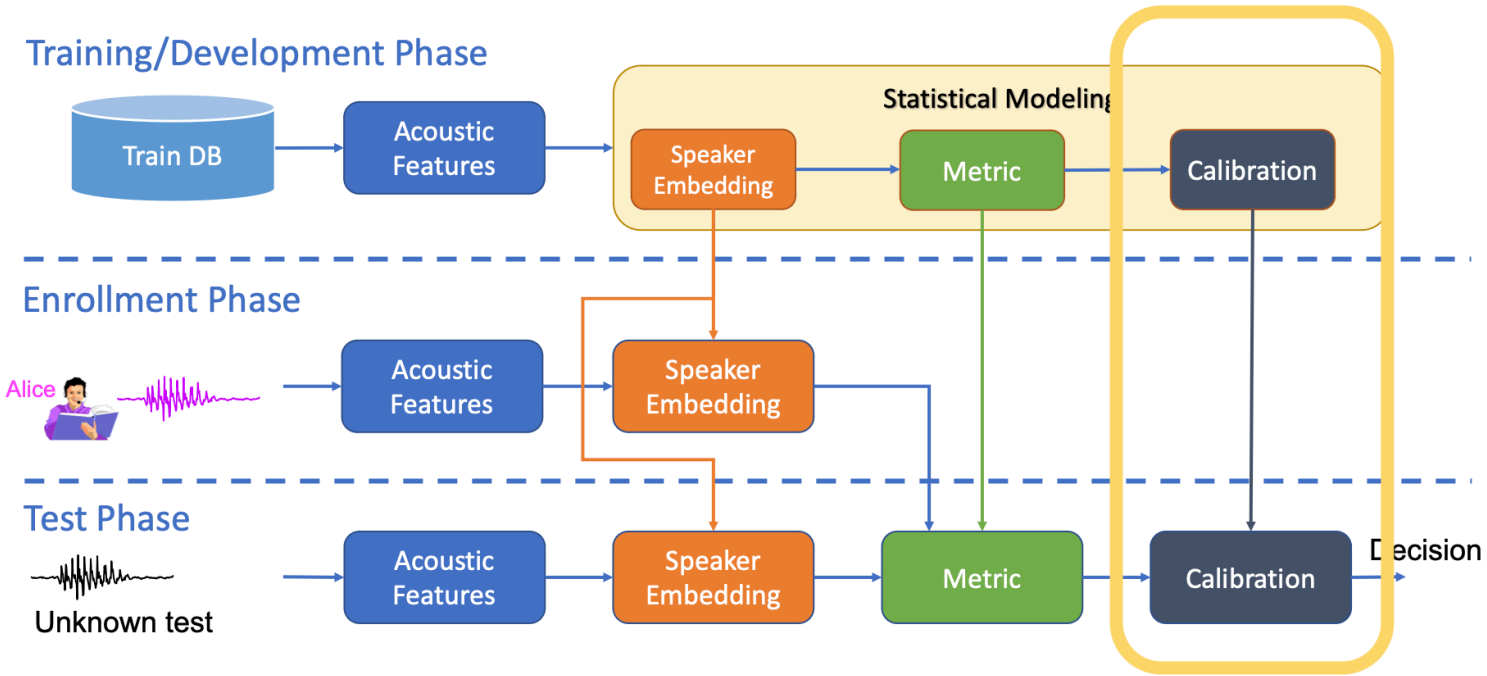
- **We assume that :**
  - w_e : speaker embedding from the enrollment utterance of speaker X
  - w_t : speaker embedding from the test utterance of the person claiming to be speaker X
- **The metric compares the enrollment and test embeddings, w_e and w_t :**
  - Cosine scoring :

$$s = \cos(\theta) = \frac{w_e \cdot w_t}{\|w_e\| \|w_t\|}$$

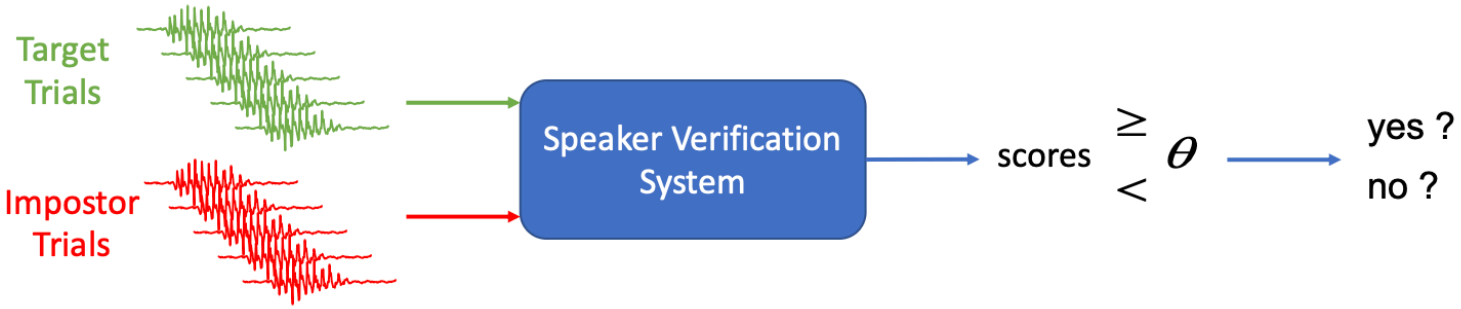
  - PLDA



Speaker Verification Pipeline : Calibration

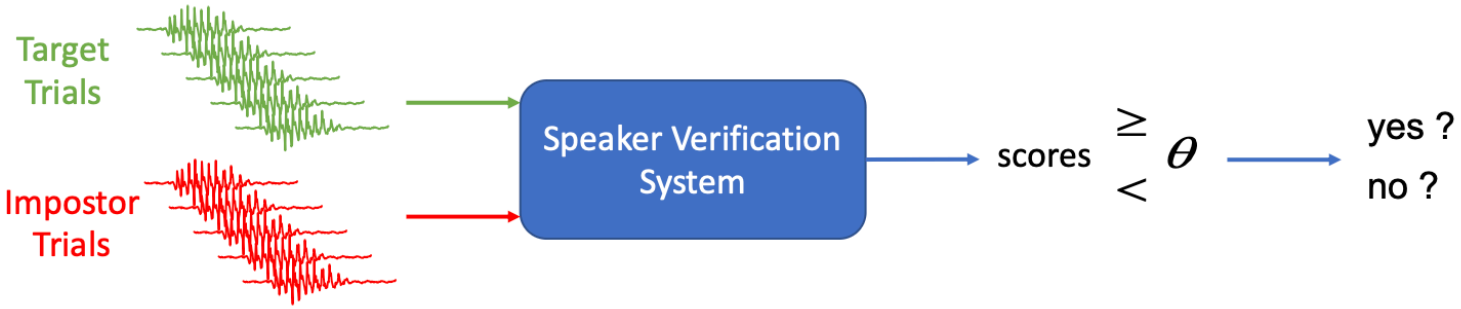


Calibration, the Art of Choosing the Threshold

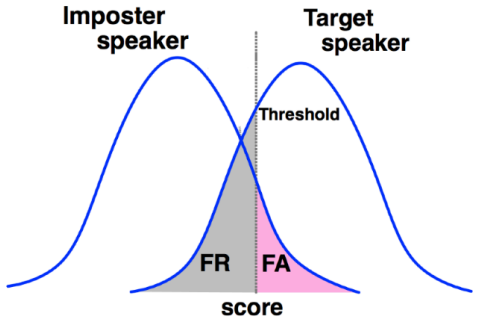


- **How do we choose the decision threshold ?**
  - For high-security applications : Opt for a higher threshold to enhance security.
  - For less critical applications : A lower threshold may suffice, balancing convenience and security.

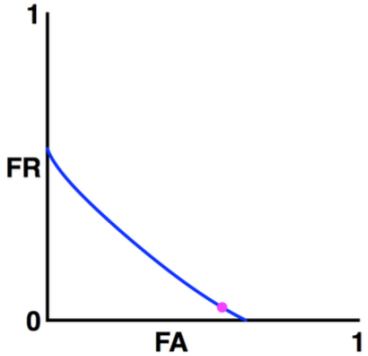
Performance Metrics



- **Put target and impostor trials into the system and count the errors**
- **Types of Errors :**
  - Miss/False Rejection
    - True speaker is classified as an impostor
    - Metric : Miss rate P_{Miss}
  - False alarm
    - Impostor is classified as the true speaker
    - Metric : False alarm rate P_{FA}



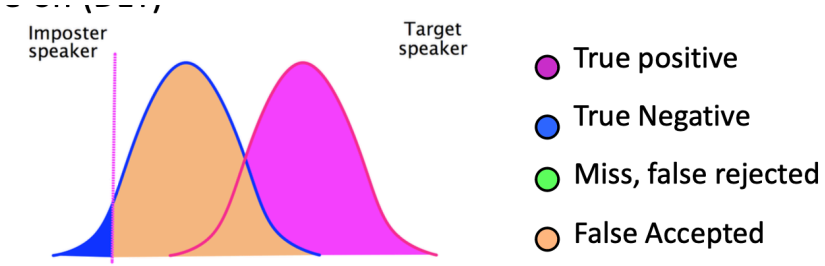
Performance Metrics



- Equal Error Rate (EER)

$$P_{Miss}(\theta) = P_{FA}(\theta)$$

- θ decision threshold



- Detection Cost Function (DCF)

$$C_{DET}(\theta) = P_{Miss}(\theta) + \beta P_{FA}(\theta)$$

- $\beta = \frac{1 - P_{target}}{P_{target}}$
- minimum C_{DET} = $min_\theta C_{DET}(\theta)$

Section 4

Neural Embeddings

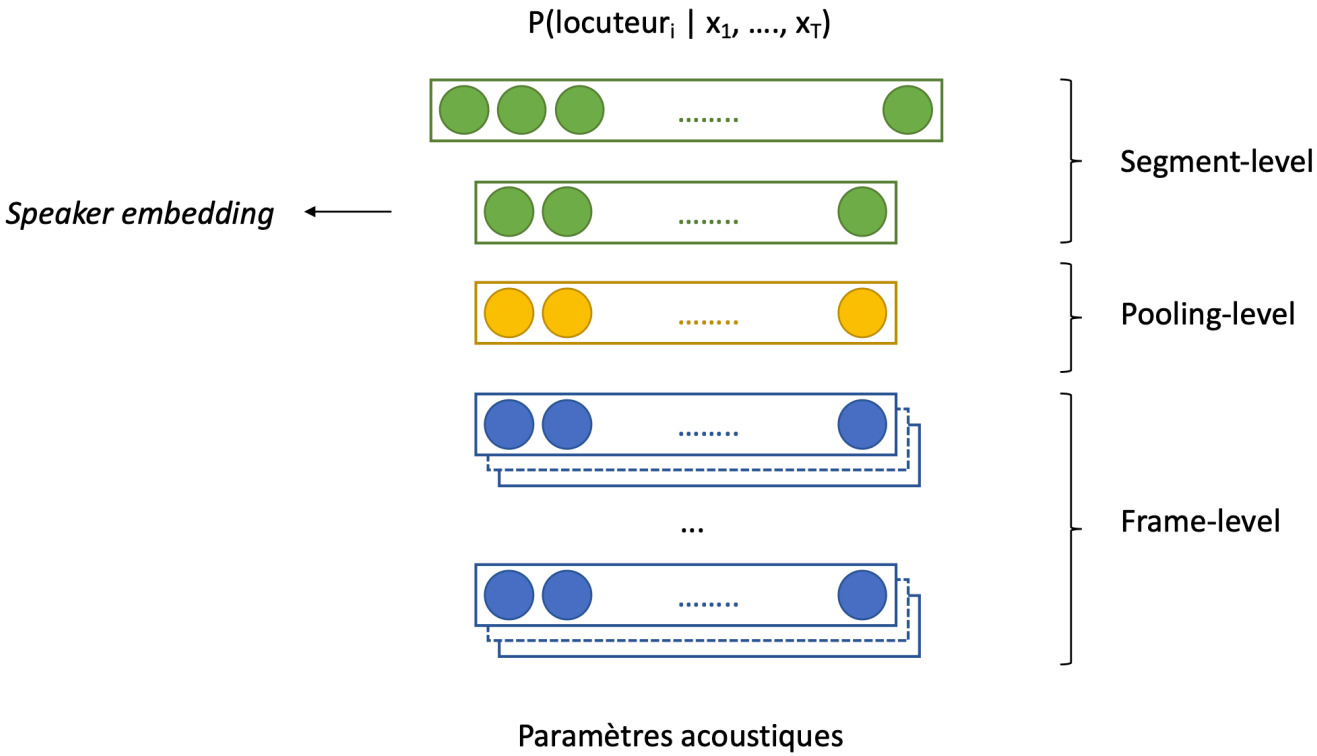
AVIGNON
UNIVERSITÉ

Speaker embeddings

- **Purpose :** The model is specifically trained for speaker identification tasks.
- **Components of the Speaker Embeddings Network :**
  - **Frame Level :** Processes acoustic features extracted from individual frames to capture the immediate properties of the audio signal.
  - **Pooling Level :** Aggregates the frame-level features over time, computing statistical measures like mean and standard deviation.
  - **Segment Level :** Applies the aggregated statistics to generate a durable speaker embedding. This embedding effectively captures and represents the unique identity characteristics of the speaker.
- **Loss Function : Categorical Cross-Entropy Loss**
  - This loss function measures the disparity between predicted speaker identities and the actual identities from the training data.

$$L = -\sum_{i=1}^{M} logP(y_i = t_i|X_i)$$

Speaker embeddings



Section 5

Frame level Architecture

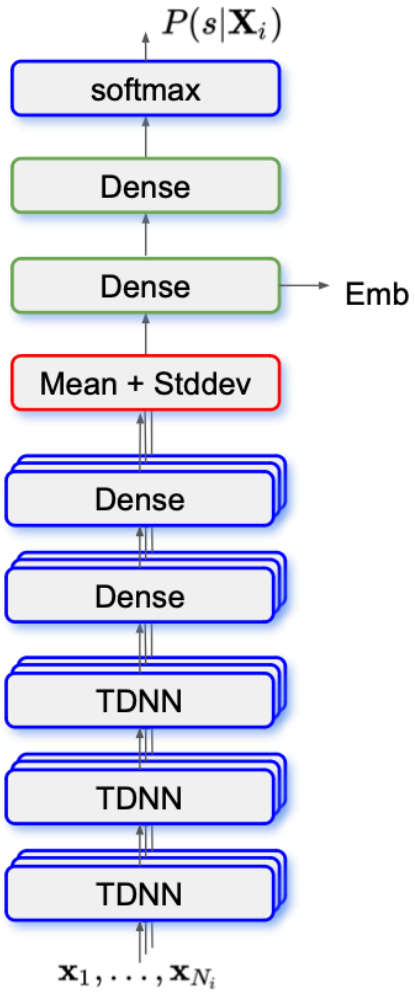
AVIGNON
UNIVERSITÉ

Frame-level Architectures for Speaker Embeddings

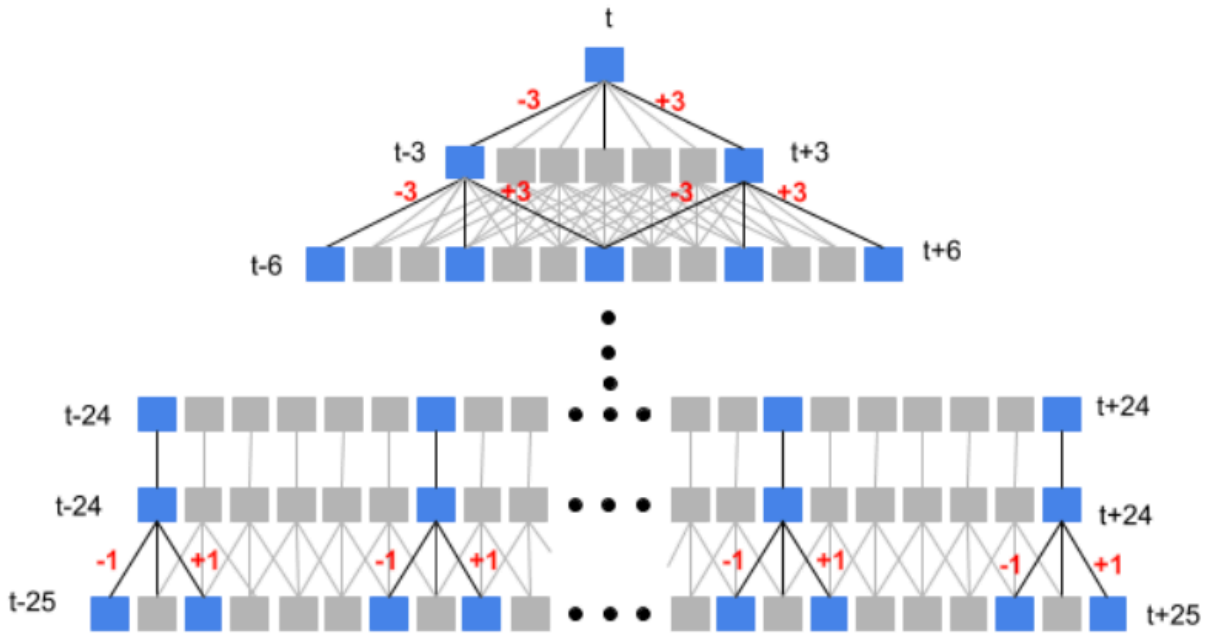
- **Time Delay Neural Network (TDNN) :**
  - A type of feedforward neural network designed to capture temporal dependencies in sequential data by using time-delayed inputs at multiple layers.
- **Residual Networks (ResNet) :**
  - A deep neural network architecture that uses residual connections to mitigate the vanishing gradient problem in very deep networks.
- **ECAPA-TDNN :**
  - An enhanced version of TDNN, integrating channel attention mechanisms to improve speaker embedding extraction.
- **Wav2Vec 2.0 :**
  - A self-supervised learning model that learns speech representations directly from raw audio waveforms.

Time Delay Neural Network

- **Time Delay Neural Network (TDNN)**
  - Designed to recognize temporal patterns in sequential data, such as speech or time series.
- **Aggregates information over progressively larger receptive fields as it delves deeper into the layers.**
  - Each layer captures more contextual information from the input, aiding in understanding complex dependencies.
- **Utilizing dilation within the network architecture causes the receptive field to expand more rapidly.**
  - Dilation involves spacing out the kernel elements, enabling the network to cover larger areas without increasing the size of the kernels.
  - Enhances the ability to process information from various time scales without additional computational costs.

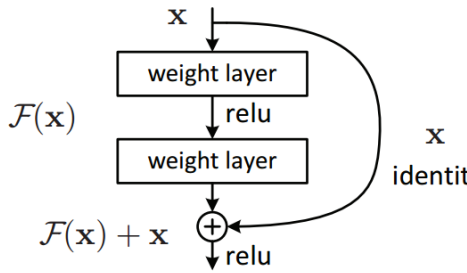


Time Delay Neural Network



Residual Networks (ResNet)

- **ResNet : A Deep Learning Architecture**
  - Designed to enable training of extremely deep neural networks.
  - Helps mitigate the vanishing gradient problem in deep networks.
- **Key Feature : Residual Blocks**
  - Each block includes a shortcut or skip connection that allows activations to be passed directly to deeper layers.
  - Skip connections help to preserve the identity of the input, which stabilizes learning and allows the network to learn identity functions effectively.
- **Benefits of Using ResNet**
  - Enables training of networks with hundreds or even thousands of layers effectively.
  - Achieves higher accuracy with increased depth without a corresponding increase in training difficulty.
  - Widely used in image recognition, achieving state-of-the-art results on benchmarks like ImageNet.



Residual Networks (ResNet)

Table 1 – The proposed ResNet34 architecture. Last row, N is the number of speakers. Batch-norm and ReLU layers are not shown. The dimensions are (Frequency × Time × Channels). The input consists of 60 filter banks from speech segments. During training, we use a fixed segment length of 400.

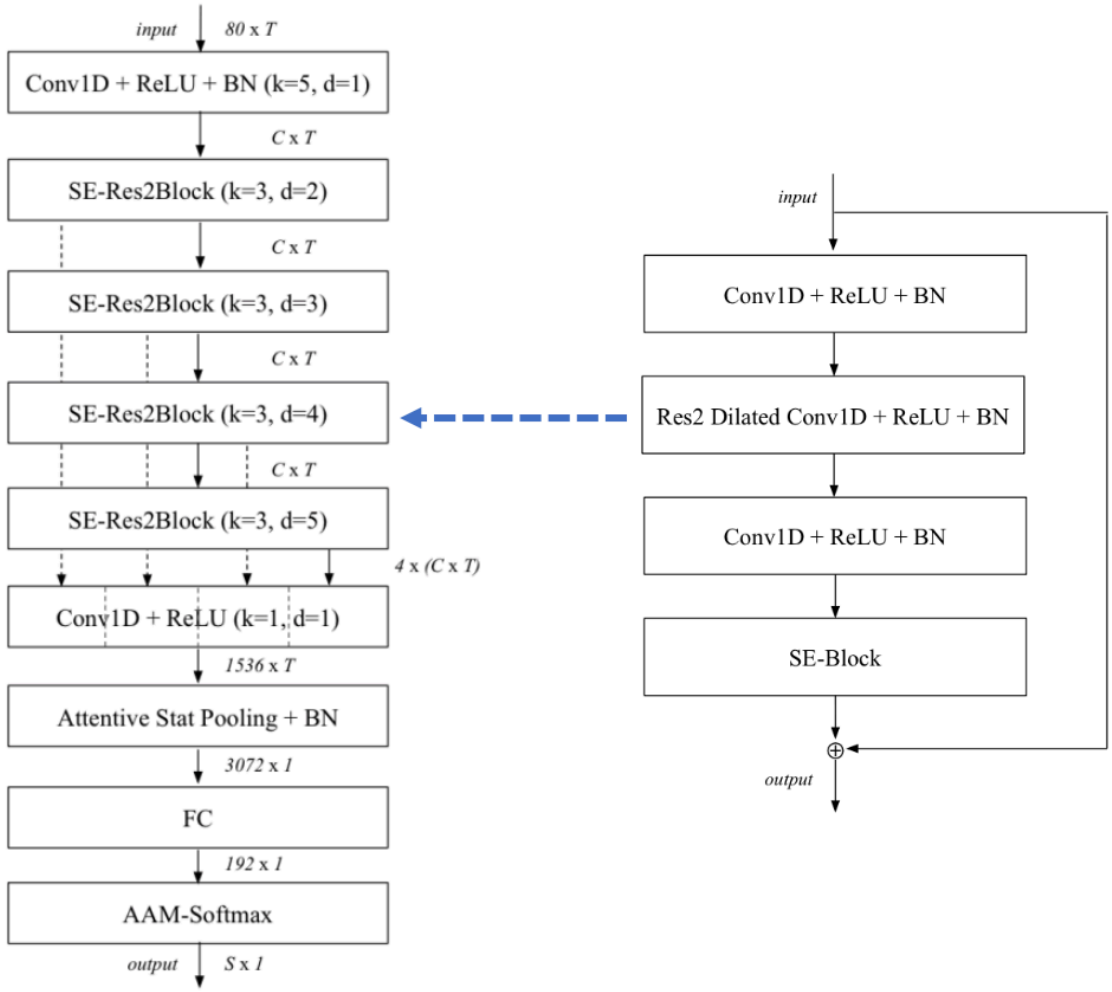
Layer name	**Structure**	**Output**
Input	–	$60 \times 400 \times 1$
Conv2D-1	3×3, Stride 1	$60 \times 400 \times 32$
ResNetBlock-1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$, Stride 1	$60 \times 400 \times 32$
ResNetBlock-2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$, Stride 2	$30 \times 200 \times 64$
ResNetBlock-3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$, Stride 2	$15 \times 100 \times 128$
ResNetBlock-4	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$, Stride 2	$8 \times 50 \times 256$
Pooling	–	8×256
Flatten	–	2048
Dense1	–	256
Dense2 (Softmax)	–	N
Total	–	–

Squeeze-and-Excitation (SE) Blocks

- **Overview of SE Blocks**
  - SE blocks are a network architecture component designed to improve the representational capacity of convolutional networks by focusing on channel-wise feature recalibrations.
  - They adaptively adjust channel responses by explicitly modelling interdependencies among channels.
- **Mechanism of SE Blocks**
  - **Squeeze :** Global average pooling is applied to each channel, compressing global spatial information into a channel descriptor.
  - **Excitation :** A fully connected layer followed by a sigmoid activation function learns to scale each channel's features dynamically.
- **Benefits of Incorporating SE Blocks**
  - Enhances model performance by allowing networks to focus on more informative features.
  - Provides a lightweight integration that can significantly improve the efficiency and accuracy of existing architectures without substantial computational overhead.

ECAPA-TDNN

- ResNet 1d
- Squeeze-Excitation
- Dilated convolutions in the bottleneck layer to increase receptive field



ECAPA-TDNN

- **ECAPA-TDNN : Advanced Architecture for Speaker Recognition**
  - Builds on the principles of traditional TDNN by integrating context-aware features and perceptual augmentation.
  - Designed to capture both temporal dynamics and channel-wise feature dependencies effectively.
- **Core Features of ECAPA-TDNN**
  - Utilizes a densely connected topology within each TDNN layer, improving information flow.
  - Incorporates Squeeze-and-Excitation (SE) blocks to emphasize relevant channel-wise features, enhancing model sensitivity to important cues.
  - Employs multi-layer feature aggregation to ensure rich representational capacity.

Wav2vec 2.0

- **Wav2vec 2.0 : A Self-supervised Framework**
  - Employs a contrastive learning approach to capture rich speech representations from unlabeled audio data.
  - Enhances the ability to discern subtle phonetic differences, crucial for accurate speaker recognition.
- **Key Features in Speaker Verification**
  - Utilizes a pre-trained model to extract features from speech segments without the need for explicit transcription.
  - Leverages contextualized representations that capture temporal and spectral nuances unique to individual speakers.
- **Adaptation for Speaker Verification**
  - Fine-tuning the model on labeled speaker data enhances its specificity to speaker characteristics while retaining robustness against background noise and channel variations.
  - Integration with speaker embeddings, facilitating efficient identification and verification processes.
- **Benefits of Using Wav2vec 2.0**
  - Significantly reduces the reliance on labeled data, decreasing the overall training time and resources.
  - Achieves superior performance in speaker verification tasks, demonstrated through enhanced discrimination in benchmark

Section 6

Pooling level Architectures

AVIGNON
UNIVERSITÉ

Mean and Statistics Pooling

- **Global Average Pooling**
  - Mean of Encoder representations along the time dimension
  - For 1D Encoders : (B, C, T) -> (B, C)
  - For 2D Encoders : (B, C, F, T) -> (B, CxF, T) -> (B, CxF)
- **Global Statistics Pooling**
  - Concatenation of :
    - Mean along the time dimension
    - Standard Deviation along the time dimension
  - For 1D Encoders : (B, C, T) -> (B, 2xC)
  - For 2D Encoders : (B, C, F, T) -> (B, CxF, T) -> (B, 2xCxF)

Attentive Statistics Pooling

- **Statistics Pooling with different weight for each frame**
  - More weight is assigned to most important frames :

$$w_t = \frac{\exp\left(a_t\right)}{\sum_{t=1}^{T} \exp\left(a_t\right)}$$

  - Weight mean :

$$\mu_{att} = \sum_{t=1}^{T} w_t x_t$$

  - Weighted Standard Deviation :

$$\sigma_{att} = \sqrt{\sum_{t=1}^{T} w_t (x_t - \mu_{att})^2}$$

Multi-Head Attention

- **Attentive Statistics Pooling with H heads :**
  - Each heads focuses on different subsets or types of frames.
  - Computes a different set of weights per head

$$w_t^h = \frac{\exp\left(a_t^h\right)}{\sum_{t=1}^{T} \exp\left(a_t^h\right)}, \quad h = 1, 2, \ldots, H$$

  - The weights of each head sum to 1 over the time dimension.
  - Compute weighted statistics (mean and std. dev) for each head :

$$\mu_{att}^h = \sum_{t=1}^{T} w_t^h x_t, \quad \sigma_{att}^h = \sqrt{\sum_{t=1}^{T} w_t^h (x_t - \mu_{att}^h)^2}$$

  - Concatenate the mean and std. dev from all heads.
  - Final output feature map for multi-head attention :

$$\text{Feature Map} : (B, C, T) \rightarrow (B, 2 \times C \times H)$$

Section 7

Loss Functions

AVIGNON
UNIVERSITÉ

Multi-class Cross-Entropy Loss

- **Categorical Cross-Entropy :** This is the loss function used for training speaker embedding models in classification tasks.
- **Loss Function Formula :**

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

where :

  - N : Total number of samples.
  - C : Total number of classes.
  - $y_{i,c}$: Binary indicator (0 or 1) showing whether class label c is the correct class for sample i.
  - $\hat{y}_{i,c}$: Predicted probability of sample i belonging to class c (i.e., output of the model's softmax function).

- **Explanation :**
  - The loss penalizes the model when the predicted probability ($\hat{y}_{i,c}$) for the true class is low.
  - The model minimizes this loss by increasing the predicted probability for the correct class for each sample.

Additive Angular Margin Softmax (ArcFace)

- **ArcFace :** A loss function that introduces an angular margin to improve discriminative power in embedding space.
- **Loss Function Formula :**

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cdot(\cos(\theta_{y_i}+m))}}{e^{s\cdot(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{C}e^{s\cdot\cos(\theta_j)}}$$

where :

- N : Number of samples.
- C : Number of classes.
- θ_{y_i} : Angle between the embedding of sample i and the weight vector of the correct class y_i.
- s : Scaling factor (usually a hyperparameter to control the magnitude).
- m : Additive angular margin to enhance separation between classes.

Additive Angular Margin Softmax (ArcFace)

- **ArcFace :** A loss function that introduces an angular margin to improve discriminative power in embedding space.
- **Explanation :**
  - ArcFace improves class separability by enforcing a margin between the angles of embeddings for the true class and other classes.
  - The additive margin (m) makes it harder for the model to classify samples correctly unless their embeddings are closer to the correct class center.
  - The softmax operation with angular margin encourages embeddings of the same class to form tighter clusters and pushes apart different class embeddings in the angular space.

Section 8

Back-end

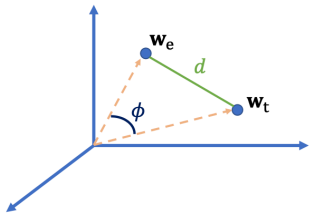
AVIGNON
UNIVERSITÉ

Metric

- **We assume that :**
  - w_e : speaker embedding from the enrollment utterance of speaker X
  - w_t : speaker embedding from the test utterance of the person claiming to be speaker X
- **The metric compares the enrollment and test embeddings, w_e and w_t :**
- Cosine scoring :

$$s = \cos(\theta) = \frac{w_e \cdot w_t}{\|w_e\| \|w_t\|}$$

- Simplest Metric
- Works well in many tasks : VoxCeleb, Eval Data is in-domain...



Linear Discriminant Analysis (LDA)

- **Goal of LDA :**
  - LDA aims to project high-dimensional data into a lower-dimensional space.
  - The projection optimizes class separation by :
    - **Maximizing the distance between class means** (between-class variance).
    - **Minimizing the spread of data points within each class** (within-class variance).

Linear Discriminant Analysis (LDA)

- **Covariance Matrices :**
  - *Between-class scatter matrix* (S_B) :

$$S_B = \sum_{i=1}^{k} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

  - *Within-class scatter matrix* (S_W) :

$$S_W = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$$

- **LDA Optimization Objective :**

$$\text{Maximize } J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}$$

where W is the projection matrix that maximizes the ratio of between-class to within-class variance.

Probabilistic Linear Discriminant Analysis (PLDA)

- **Goal of PLDA :**
  - PLDA extends Linear Discriminant Analysis by introducing a probabilistic model.
  - It models both between-class and within-class variability probabilistically.
- **Generative Model :**
  - PLDA assumes the data x can be decomposed into a sum of two components :

$$x = \mu + Vy + Uz + \epsilon$$

where :

- μ : Global mean of the data.
- Vy : Between-class component, where y is the latent variable representing the identity or class.
- Uz : Within-class component, where z represents within-class variations (e.g., session variability in speaker verification).
- ϵ : Residual noise, assumed to follow a Gaussian distribution.

Probabilistic Linear Discriminant Analysis (PLDA)

- **Likelihood Function :**
  - PLDA models the likelihood of observing a sample x given the identity y and within-class variability z :

$$p(x|y,z) = \mathcal{N}(x|\mu + Vy + Uz, \Sigma)$$

    where $\mathcal{N}$ is a Gaussian distribution with mean $\mu + Vy + Uz$ and covariance matrix Σ.
- **Comparison with LDA :**
  - In LDA, class separation is based on maximizing inter-class distance and minimizing intra-class variance.
  - In PLDA, the probabilistic formulation explicitly models the variation within and between classes using latent variables.

Back-end Scoring Takeaways

- **Cosine scoring is effective for simpler tasks :**
  - Works well when both training and test data come from the **same domain**.
  - Suitable for tasks where there is minimal domain variability.
- **PLDA outperforms cosine scoring when data is out-of-domain :**
  - PLDA is robust when the embedding training data comes from a **different domain** than the test data.
  - Incorporating some **in-domain data** during PLDA training can further improve performance.

Section 9

Other Topics in Speaker Recognition

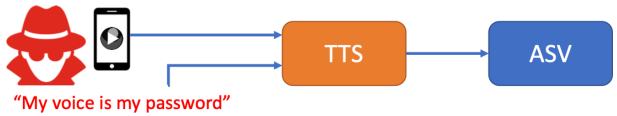
AVIGNON
UNIVERSITÉ

Spoofing Attacks

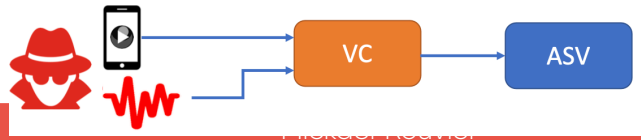
- Impostor tries to Impersonate a legitimate user
- Acquire Victim's Voice Sample

- **Replay Attack :** The attacker replays a pre-recorded voice sample of the victim to bypass the system's security.

- **Text-to-Speech (TTS) :** The attacker generates synthetic speech by converting written text into a voice that mimics the victim.



- **Voice Conversion (VC) :** The attacker converts their own voice to sound like the victim's voice using voice conversion technologies.



Spoofing Detection Methods

- **Physical Access Spoofing :**
  - Involves playing spoofed audio over an **air channel** (e.g., via loudspeaker).
  - Focuses on detecting artifacts related to **loudspeakers** and **far-field audio transmission**, such as distortions caused by environmental factors.
- **Logical Access Spoofing :**
  - Involves injecting spoofed audio directly into the Automatic Speaker Verification (ASV) system **digitally**, bypassing the air channel.
  - Focuses on detecting artifacts from **Text-To-Speech (TTS)** and **Voice Conversion (VC)** vocoders, such as synthetic signal imperfections.

Adversarial Attacks

- **Adversarial Attacks :**
  - These attacks introduce a small, carefully crafted perturbation to an input signal.
  - The perturbation is **imperceptible to humans**, meaning that the altered signal appears unchanged to the human ear or eye.