Intelligent Consumer Technologies

Prof. Paolo Napoletano
a.a. 2024/2025

Signal, image, and natural language processing in Consumer Technologies

Speech Processing

Topics: Speech Signal, Recognition tasks in Speech Processing, Wake-word and keyword detection, Voice Activity Detection, Speaker Recognition

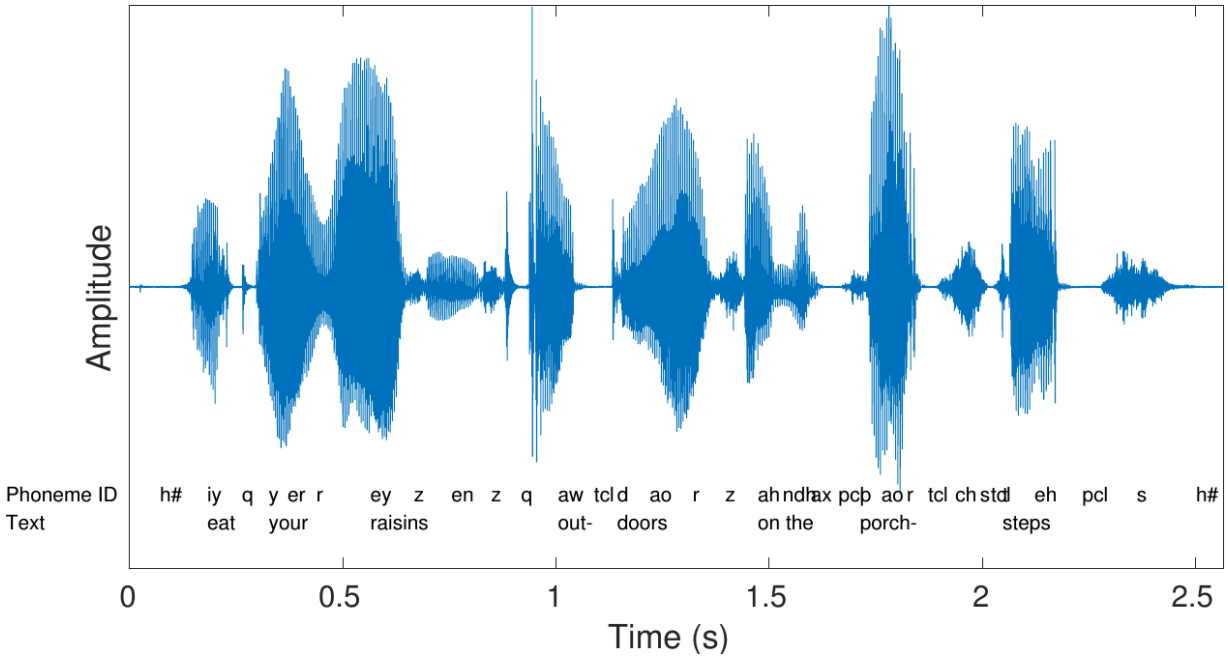
Learning Objectives

- Being able to define all recognition tasks in Speech processing
- Being able to draw a pipeline of Speech Processing for Smart Speakers
- Being able to define a simple algorithm for VAD detection
- Being able to define a simple algorithm for Wake-word detection
- Being able to define a simple algorithm for Speaker Recognition
- Being able to define a simple algorithm for Speaker Diarization

Audio Signal → Speech Signal

definitions

Speech signals are sound signals, defined as pressure variations travelling through the air. These variations in pressure can be described as waves and correspondingly they are often called sound waves.

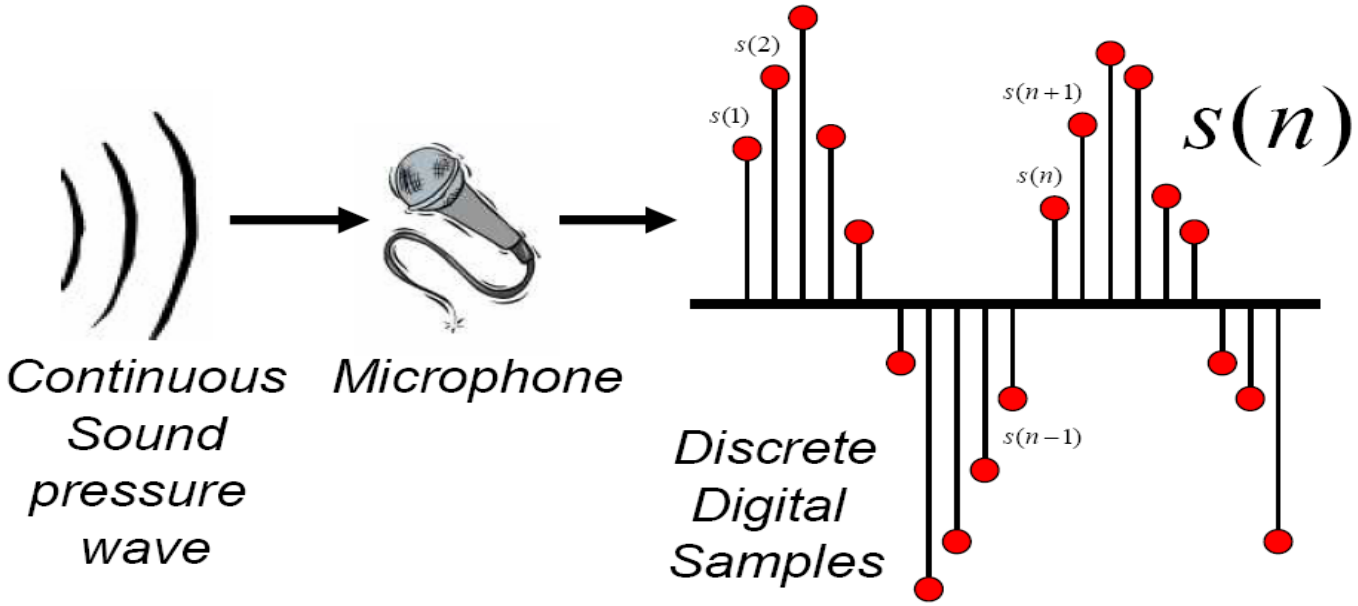


* See **additional materials** on https://speechprocessingbook.aalto.fi

Audio Signal

definitions

Represent continuous signal into discrete form.



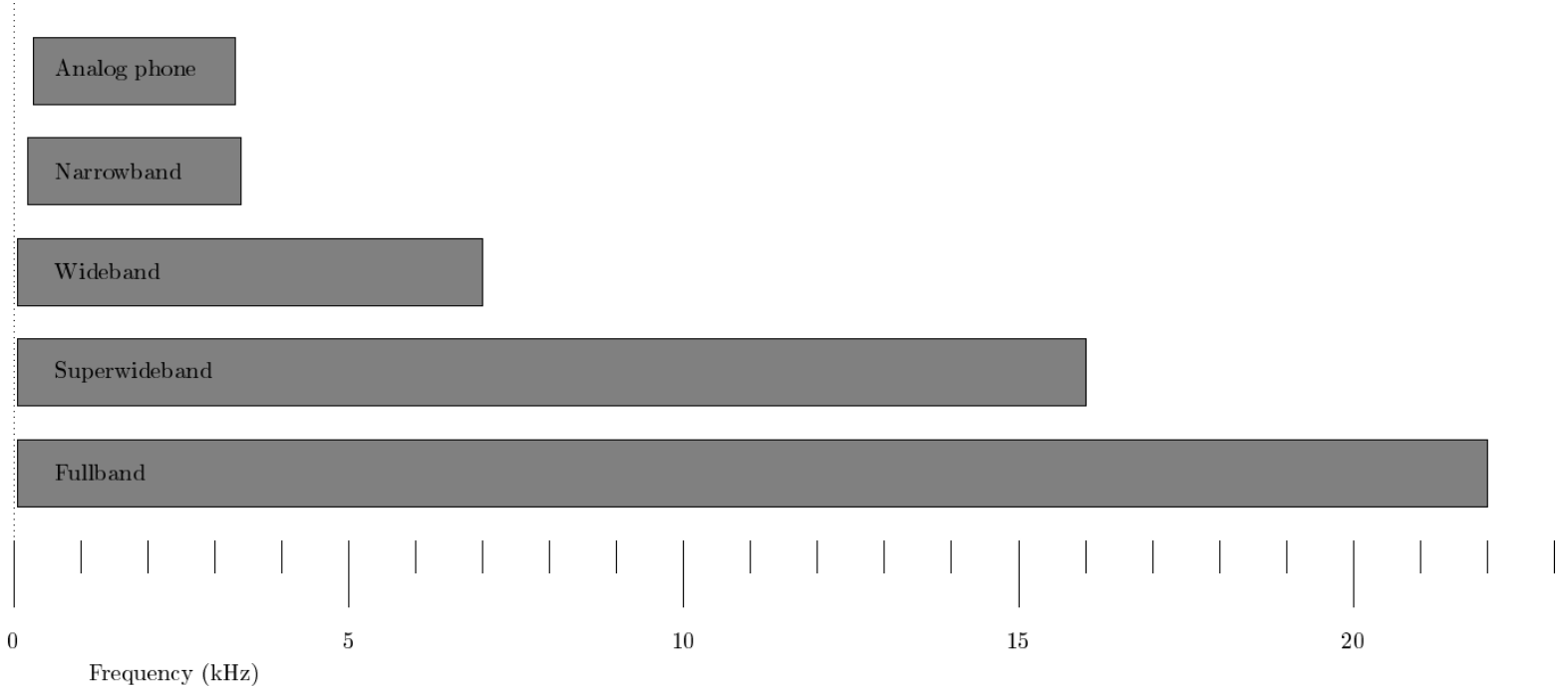
* Part of these slides are taken from Dan Jurafsky, Stanford University

Audio Signal → Speech Signal

definitions

We assume that the acoustic speech signals have been captured by a microphone and converted to a digital form.

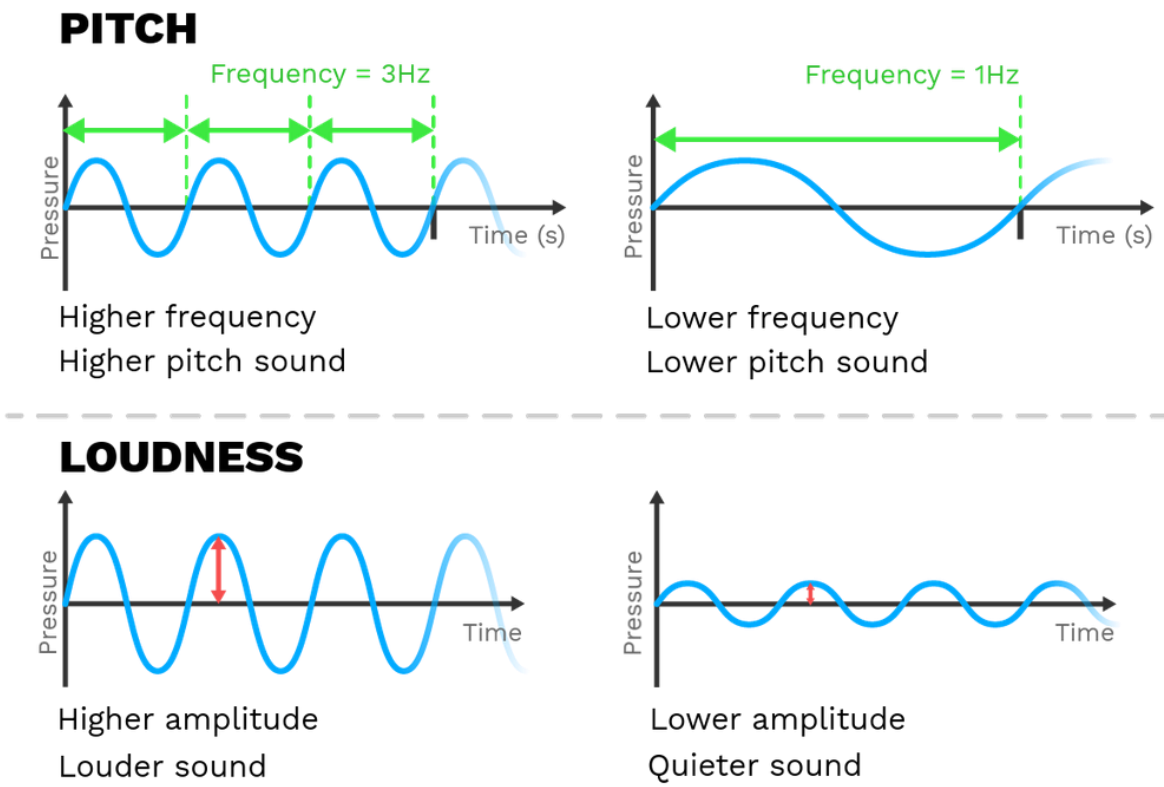
The Nyquist frequency, which is half the sampling rate and defines the upper end of the largest bandwidth



* See **additional materials** on https://speechprocessingbook.aalto.fi

Psychological Dimensions

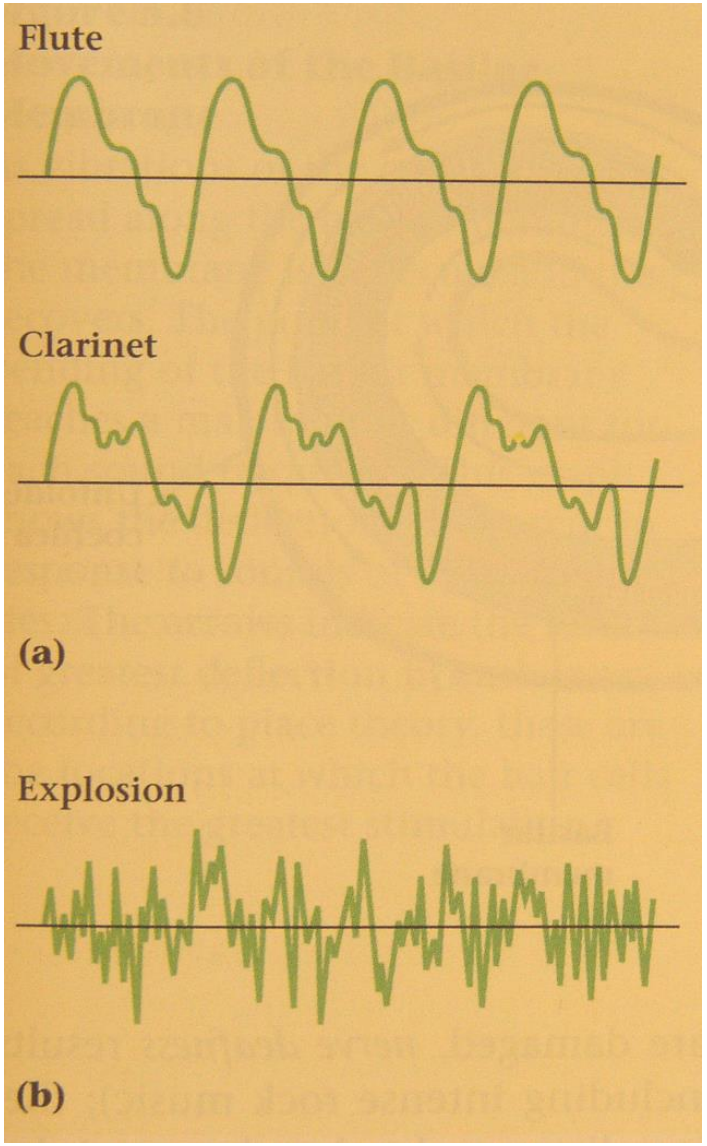
- **Pitch**
  - higher frequencies perceived as higher pitch
  - hear sounds in 20 Hz to 20,000 Hz range
- **Loudness**
  - higher amplitude results in louder sounds
  - measured in decibels (db), 0 db represents hearing threshold



Psychological Dimensions

- **Timbre** (tam-bre)
  - complex patterns added to the lowest, or *fundamental*, frequency of a sound, referred to as *spectra*
  - spectra enable us to distinguish musical instruments
  - characteristic of sound that allows us to differentiate between sounds with the same pitch and loudness but come from different sources.

- Multiples of fundamental frequency give music
- Multiples of unrelated frequencies give noise



Sound Intensity

- ***Intensity*** (I) of a wave is the rate at which sound energy flows through a unit area (A) perpendicular to the direction of travel

$$I = \frac{1}{A}\frac{\Delta E}{\Delta t} = \frac{P}{A}$$

P measured in watts (W), *A* measured in m^2

- *Threshold of hearing* is at 10^{-12} W/m^2
- *Threshold of pain* is at 1 W/m^2

Decibel Scale

Describes intensity relative to threshold of hearing based on multiples of 10

$$dB = 10 \log \frac{I}{I_0}$$

I_0 is reference level
$= 10^{-12}$ W/m^2

Sound	Decibels
Rustling leaves	**10**
Whisper	**30**
Ambient office noise	**45**
Conversation	**60**
Auto traffic	**80**
Concert	**120**
Jet motor	***140***
Spacecraft launch	***180***

Audio Perception

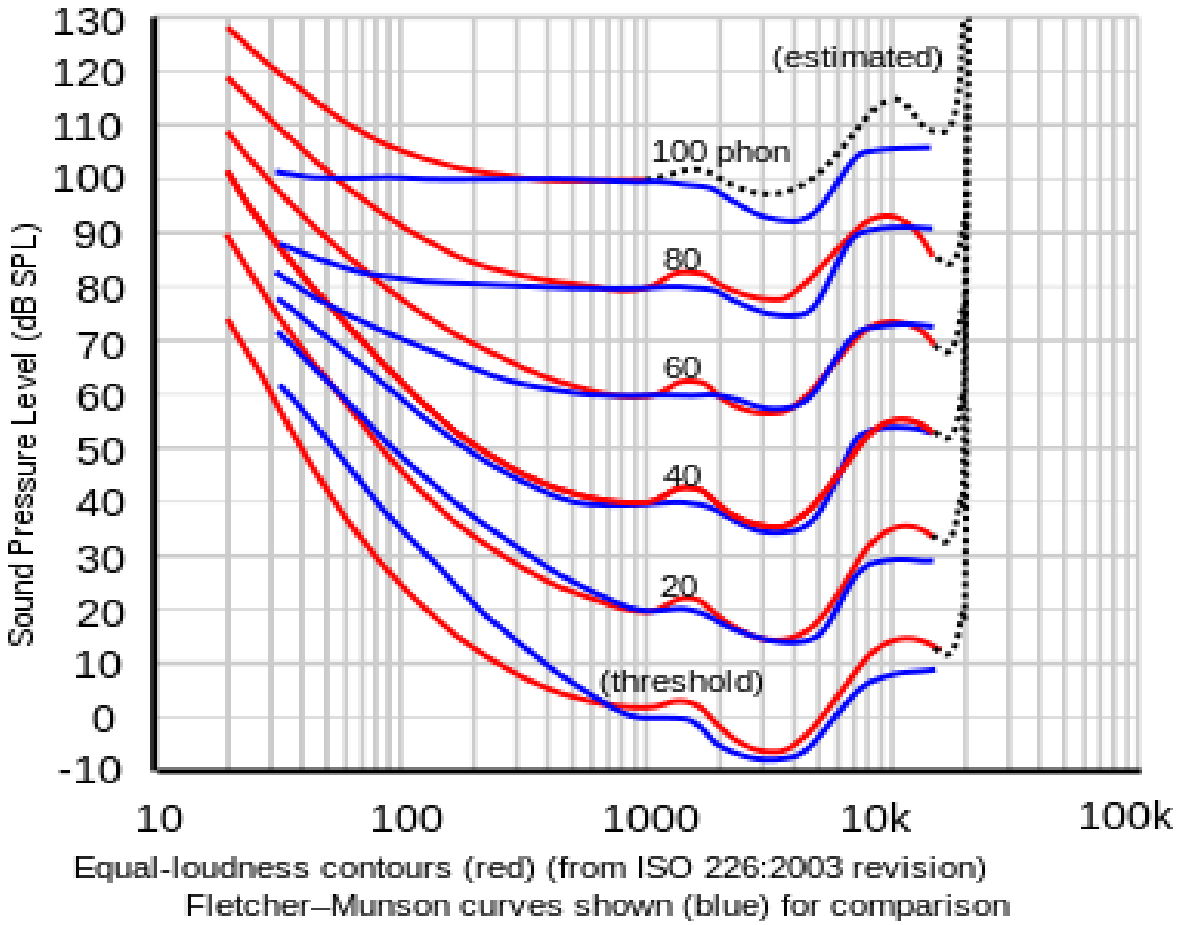
- More sensitive to **loudness** at **mid frequencies** than at other frequencies
  - intermediate frequencies at [500hz, 5000hz]

- Perceived loudness of a sound changes based on the frequency of that sound
  - basilar membrane reacts more to intermediate frequencies than other frequencies

* Parts of these slides are taken from Klara Nahrstedt

Audio Perception

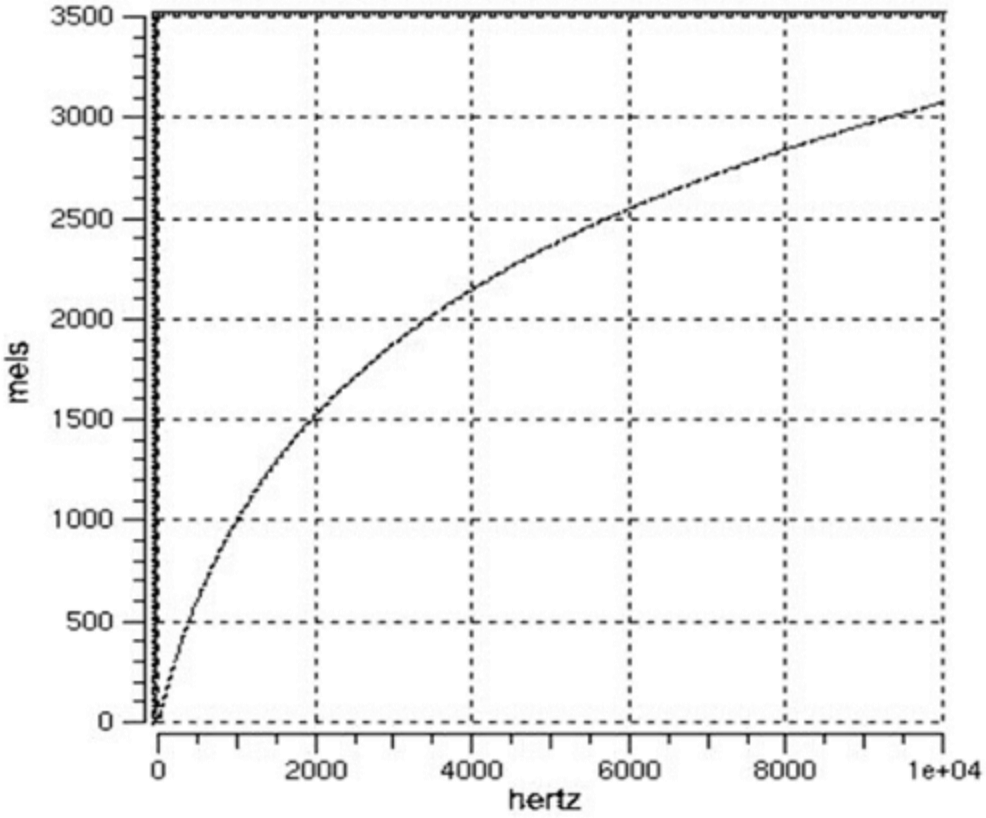
Each contour represents an equal perceived sound



* Parts of these slides are taken from Klara Nahrstedt

Mel scale

- Human hearing is not equally sensitive to all frequency bands
- Less sensitive at higher frequencies, roughly > 1000 Hz
- I.e. human perception of frequency is non-linear.
- Proposed by Stevens, Volkman and Newman in 1937 as a perceptual scale of pitches
- A 1000 Hz tone, 40 dB above the listener's threshold = 1000 mels.



Recognition tasks in Speech processing

definitions

- **Speech recognition**, which refers to converting an acoustic waveform of spoken speech to the corresponding text (speech-to-text).
- **Speaker recognition and speaker verification**, which refer to, respectively, identifying the speaker (who is speaking?) and verifying whether the speaker is who he claims to be (is it really you?).
- **Speech synthesis**, which entails the creation of a natural sounding speech signal from text input (text-to-speech).
- **Speech enhancement**, refers to improving a recorded speech signal, for example with the objective of removing background noise (noise attenuation) or the effect of room acoustics.

* See **additional materials** on https://speechprocessingbook.aalto.fi

Recognition tasks in Speech processing

definitions

- **Wake-word** and **keyword detection**, refers to the task where the purpose is to find single characterizing words from continuous speech.
- **Voice activity detection (VAD)**, refers to the task of determining whether a signal contains speech or not (is someone speaking?).
- **Speech diarisation** is the process of segmenting a multi-speaker conversation into continuous single-speaker segments.
- **Paralinguistic analysis tasks**, refers generally to the extraction of non-linguistic and non-speaker identity related information from speech signals, such as speaker emotions, health, attitude, sleepiness etc.

* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection

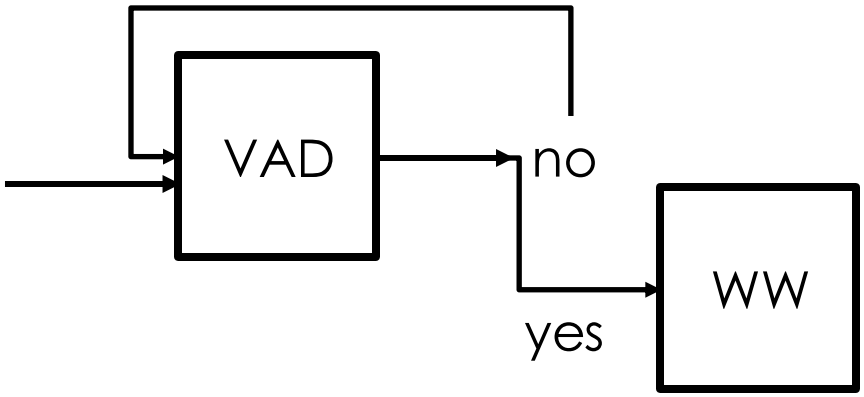
Voice Activity Detection (VAD)

details

- Most of the time, **speech devices** are just sitting there **waiting** for instructions when nobody is speaking. Speech recognition algorithms moreover use a lot of computational resources such that it would be wasteful to have them analyse sounds when there is no speech present.

- **Voice activity detection** (VAD) takes care of the first part, to detect whether speech is present or not, such that all activities are in a sleep mode when speech is not present.

```
while(true){

        audio_segment = getAudio();

        if(VAD(audio_segment){

                out= ww_detection(audio_segment);

        }

}
```

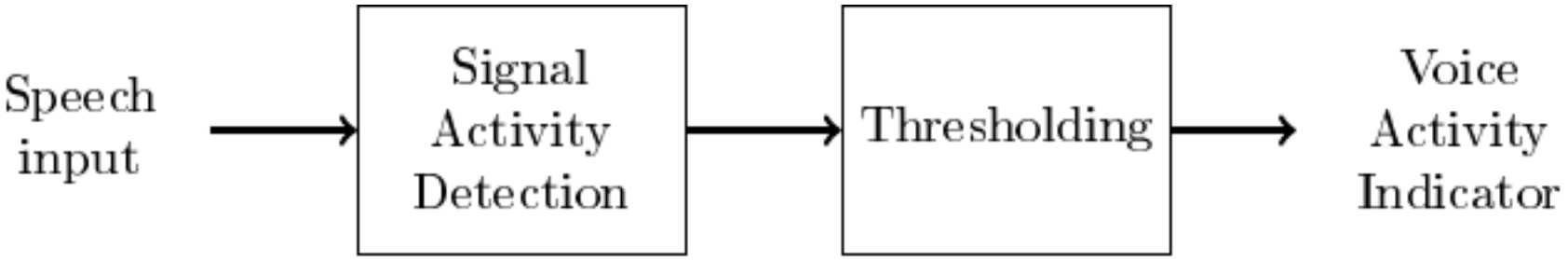


* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection (VAD)

Low-noise VAD = Trivial case

- Let us consider a case where a speaker is speaking in a (otherwise) **silent environment**.
  - When there is no speech, there is silence.
  - (Any) Signal activity indicates voice activity.
- Signal activity can be measured by, for example, estimating **signal energy per frame** ⇒ the energy thresholding algorithm



* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection (VAD)

Signal Energy

By **signal energy**, we usually mean the variance of the signal, which is the average squared deviation from the mean:

$$Energy(x) = var(x) = E[(x - \mu)^2]$$

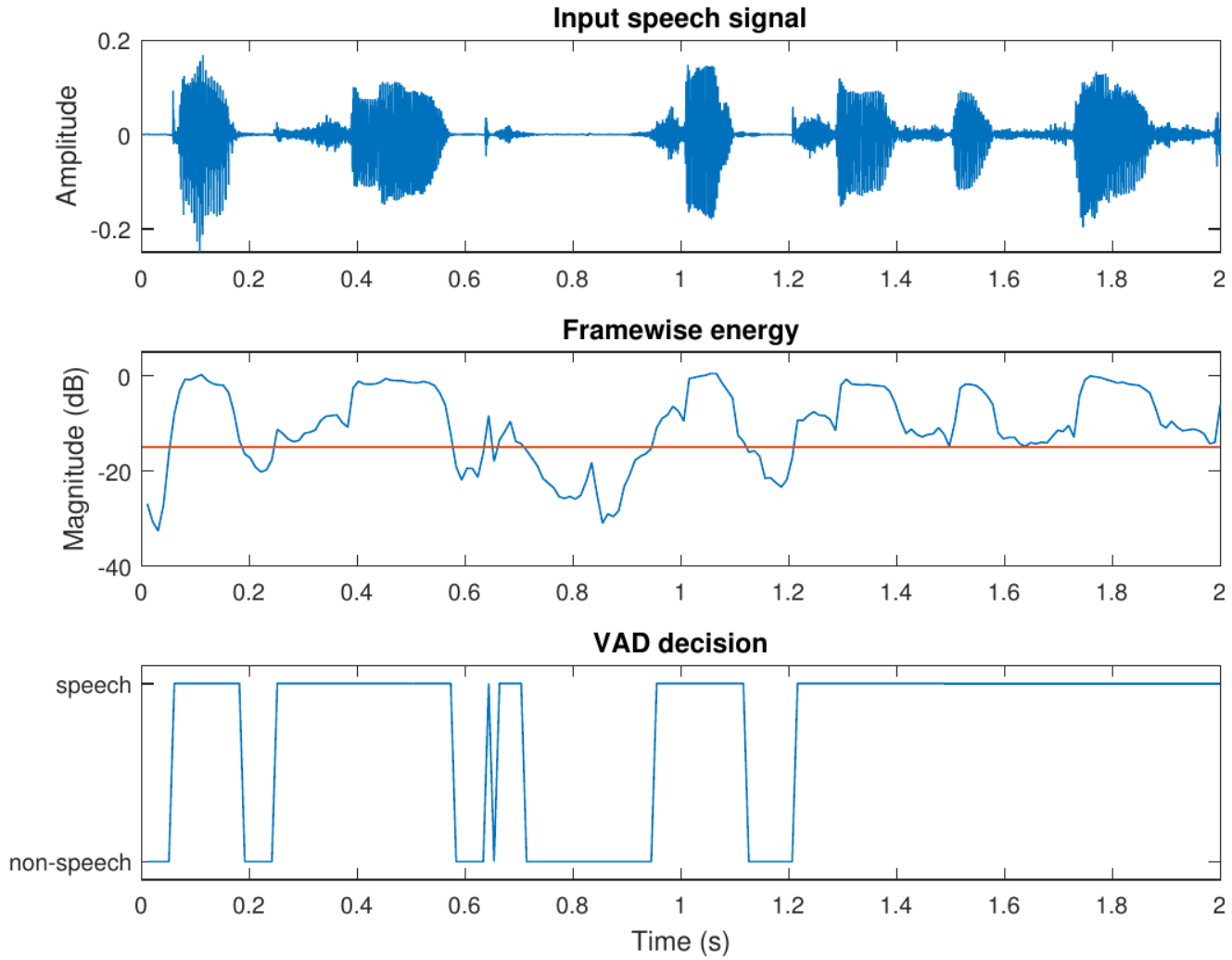
Where, $\mu = E[x]$; the average of the signal *x*.

Since the amplitude of an oscillating signal varies through the period of the oscillation, it does not usually make sense to estimate the instantaneous energy, but only **averaged over some window**.

* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection (VAD)

Low-noise VAD = Trivial case



* See **additional materials** on https://speechprocessingbook.aalto.fi, https://www.audiolabs-erlangen.de/resources/MIR/FMP/C6/C6S1_NoveltyEnergy.html

Voice Activity Detection (VAD)

High-noise VAD = Not trivial case

- **Clean speech** (absolutely no background noise) is very rare if not impossible to achieve.
- **Real-life speech recordings** practically always have varying amounts of **background** noise.
  - **Performance** of energy thresholding **decreases** rapidly when the **SNR** drops**.
  - For example, weak offsets easily disappear in noise.
- We need more **advanced VAD** methods **for noisy speech**.
  - We need to identify **characteristics** that differentiate between speech and noise.
  - Measures for such characteristics are known as **features**.

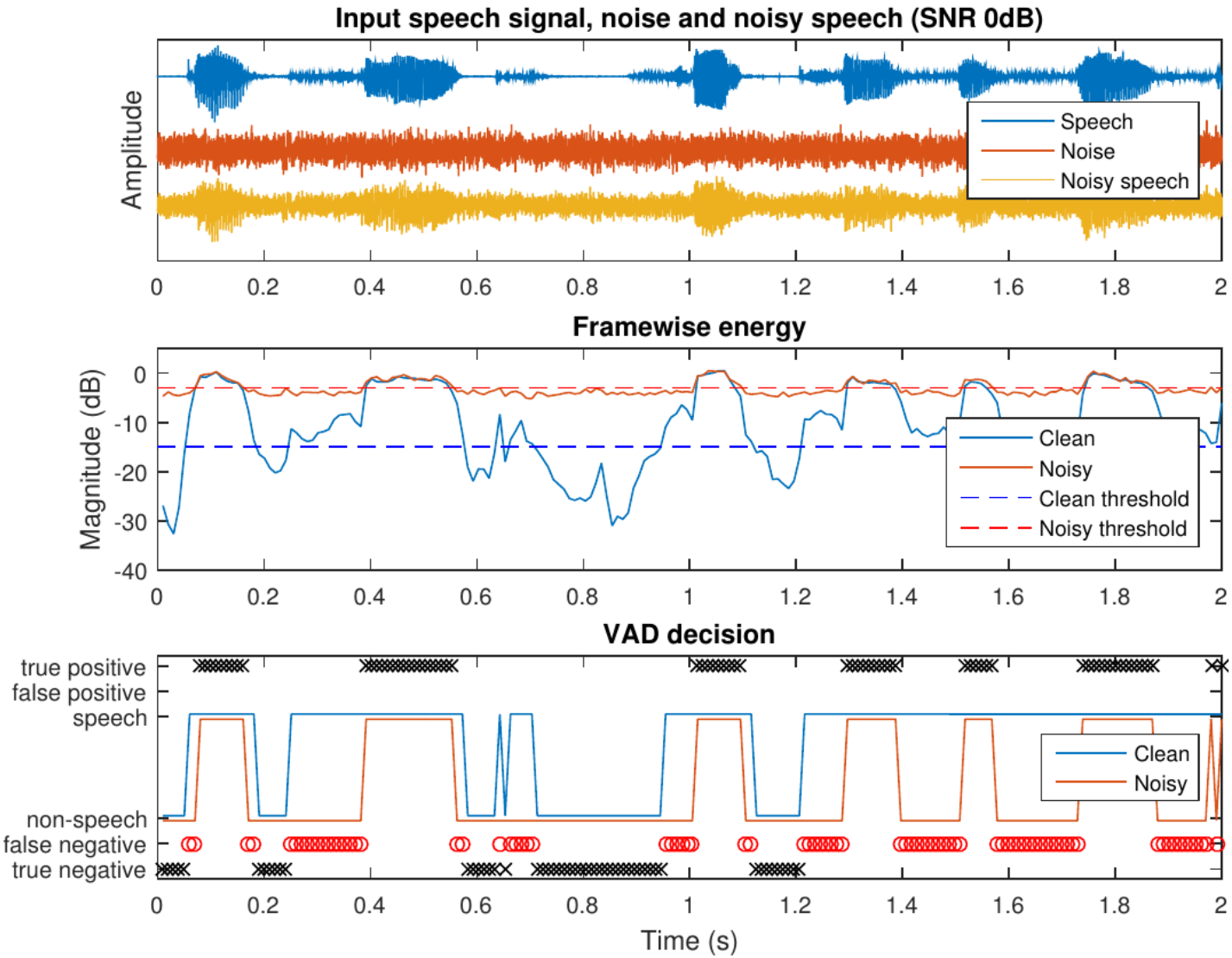
** SNR = Signal-to-noise ratio

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}$$

* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection (VAD)

High-noise VAD = Not trivial case

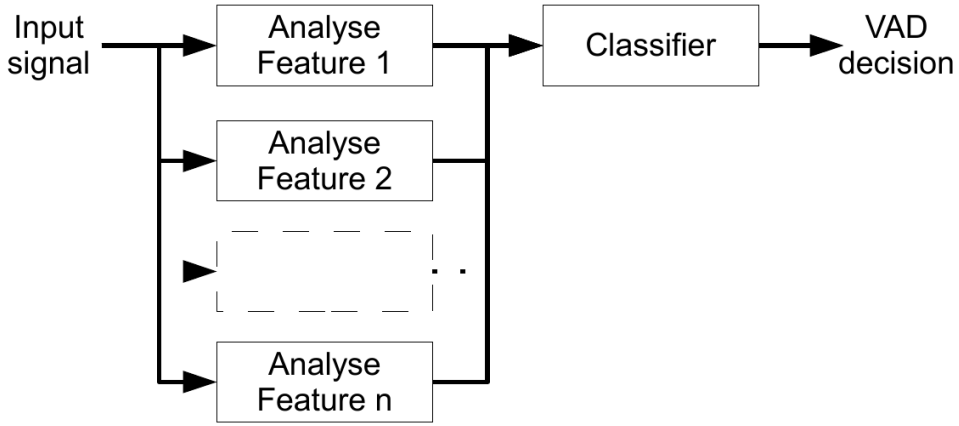


* See **additional materials** on https://speechprocessingbook.aalto.fi

Voice Activity Detection (VAD)

High-noise VAD = Not trivial case

- **MFCC**** can be used as a description of envelope information and it is thus a useful set of features.
- Classification is generic problem, with plenty of solutions such as
  - *decision trees* (low-complexity, requires manual tuning)
  - *linear classifier* (relatively low-complexity, training from data)
  - advanced methods such as *neural networks*, *Gaussian mixture* models etc. (high-complexity, high-accuracy, training from data)



** see next slides

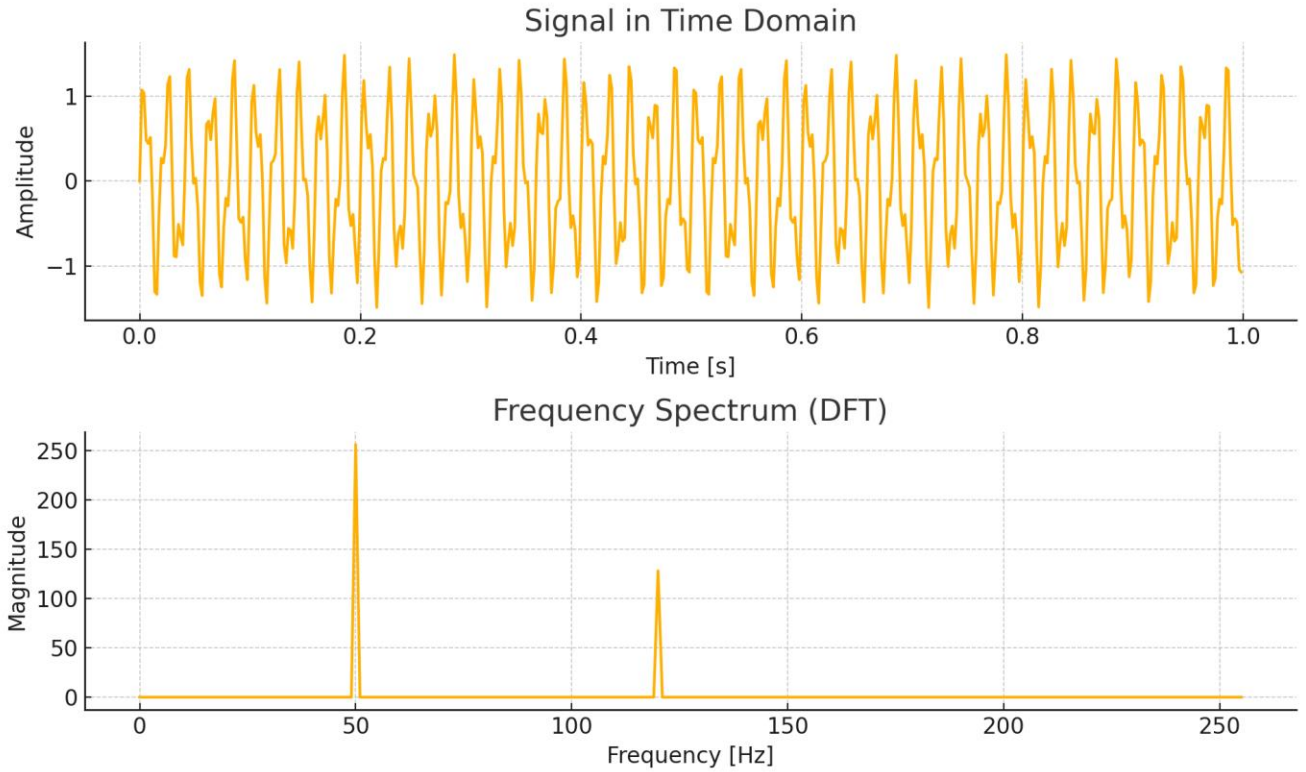
* See **additional materials** on https://speechprocessingbook.aalto.fi

Discrete Fourier Transform (DFT)

- **Input:**
  - A sequence of samples taken from a discrete signal *x[0]…x[N-1]*
- **Output:**
  - For each of K discrete frequency bands
  - A complex number *X[k]* represents the magnitude and phase of that frequency *k* component in the original signal
- **Discrete Fourier Transform (DFT)**
- $$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\frac{\pi}{N}kn}$$
- **Standard algorithm for computing DFT (**originally its complexity is $O(N^2)$**):**
  - Fast Fourier Transform (FFT) with complexity N*log(N)
  - In general, choose N=512 or 1024

Discrete Fourier Transform (DFT)

- **Example**



$$X[k] = |X[k]| \cdot e^{j \cdot \theta[k]}$$

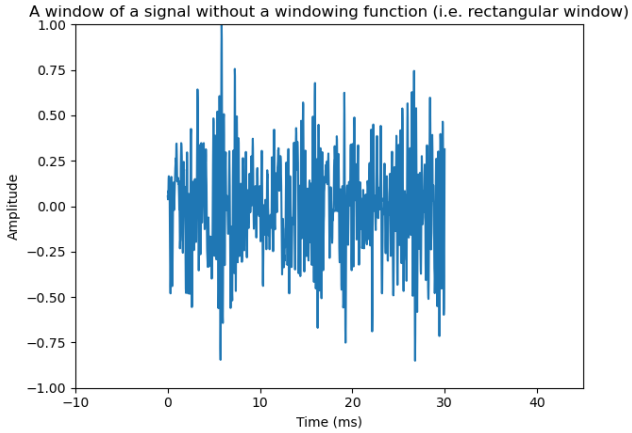
$$X[k] = \mathrm{Re}(X[k]) + j \cdot \mathrm{Im}(X[k])$$

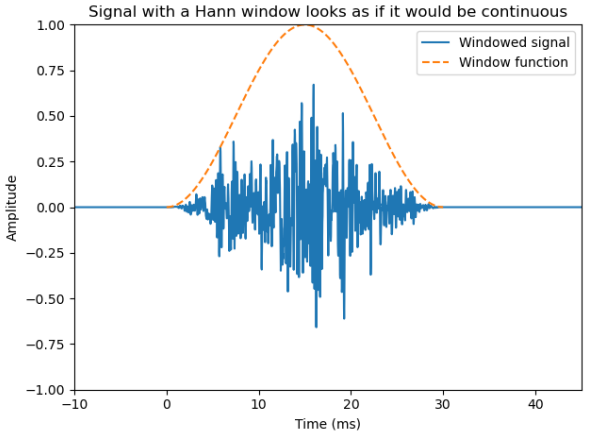
$$|X[k]| = \sqrt{\mathrm{Re}(X[k])^2 + \mathrm{Im}(X[k])^2}$$

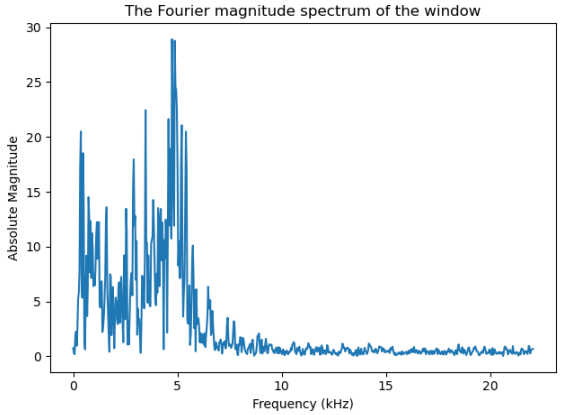
Spectrograms

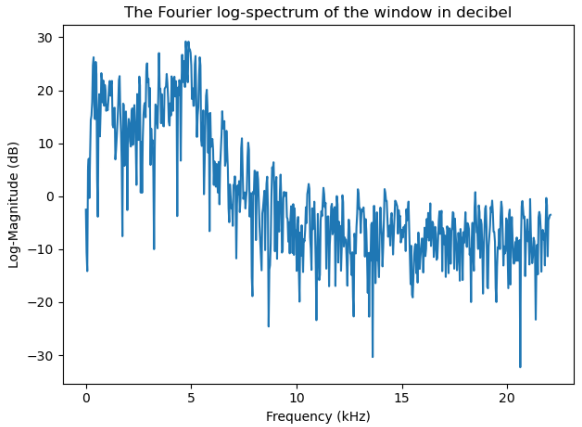
pipeline

By **splitting** the **signal** into **shorter segments**, we can focus on signal properties at a particular point in time.







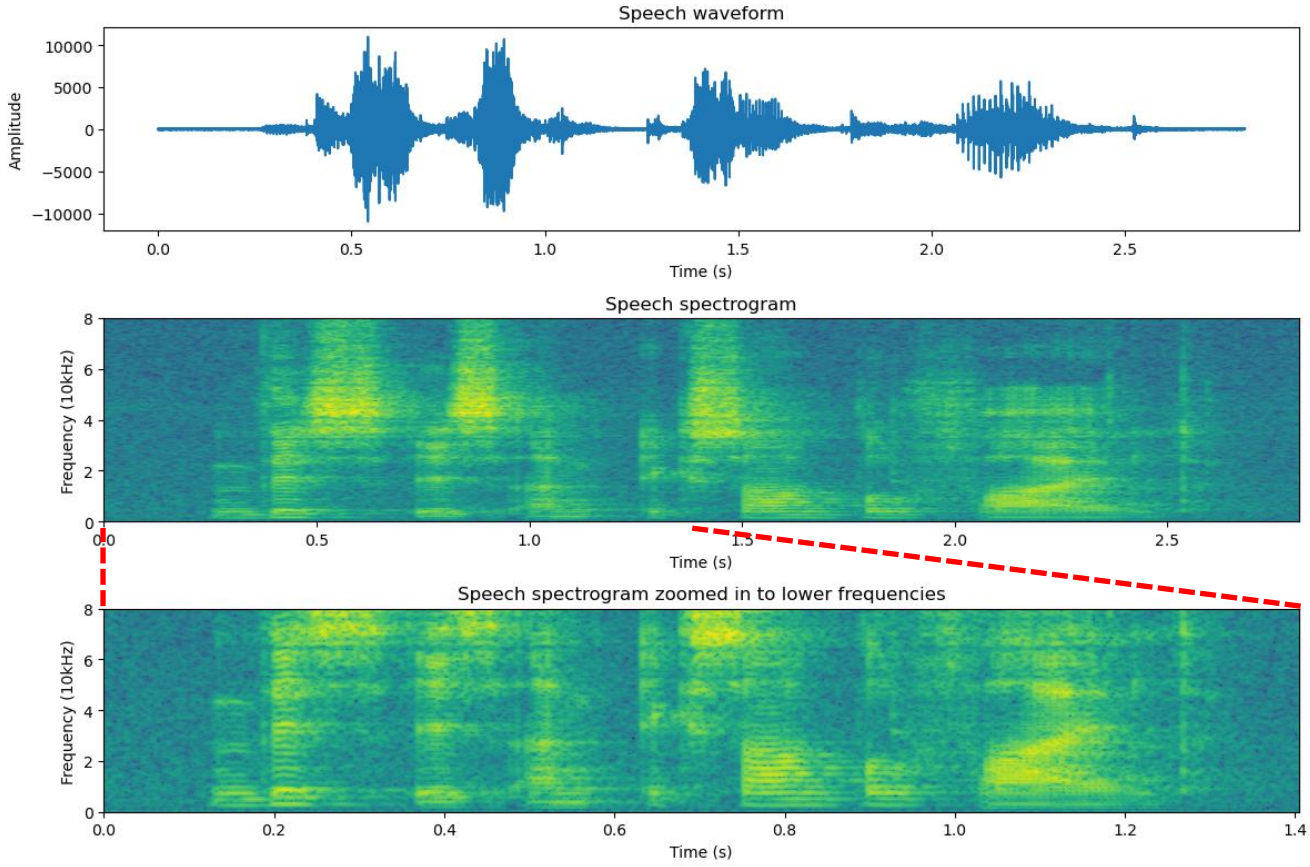


* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Spectrogram_and_the_STFT.html

Spectrograms

pipeline

By windowing and taking the **Discrete Fourier Transform** (**DFT**) of each window, we obtain the short-time Fourier transform (**STFT**) of the signal.

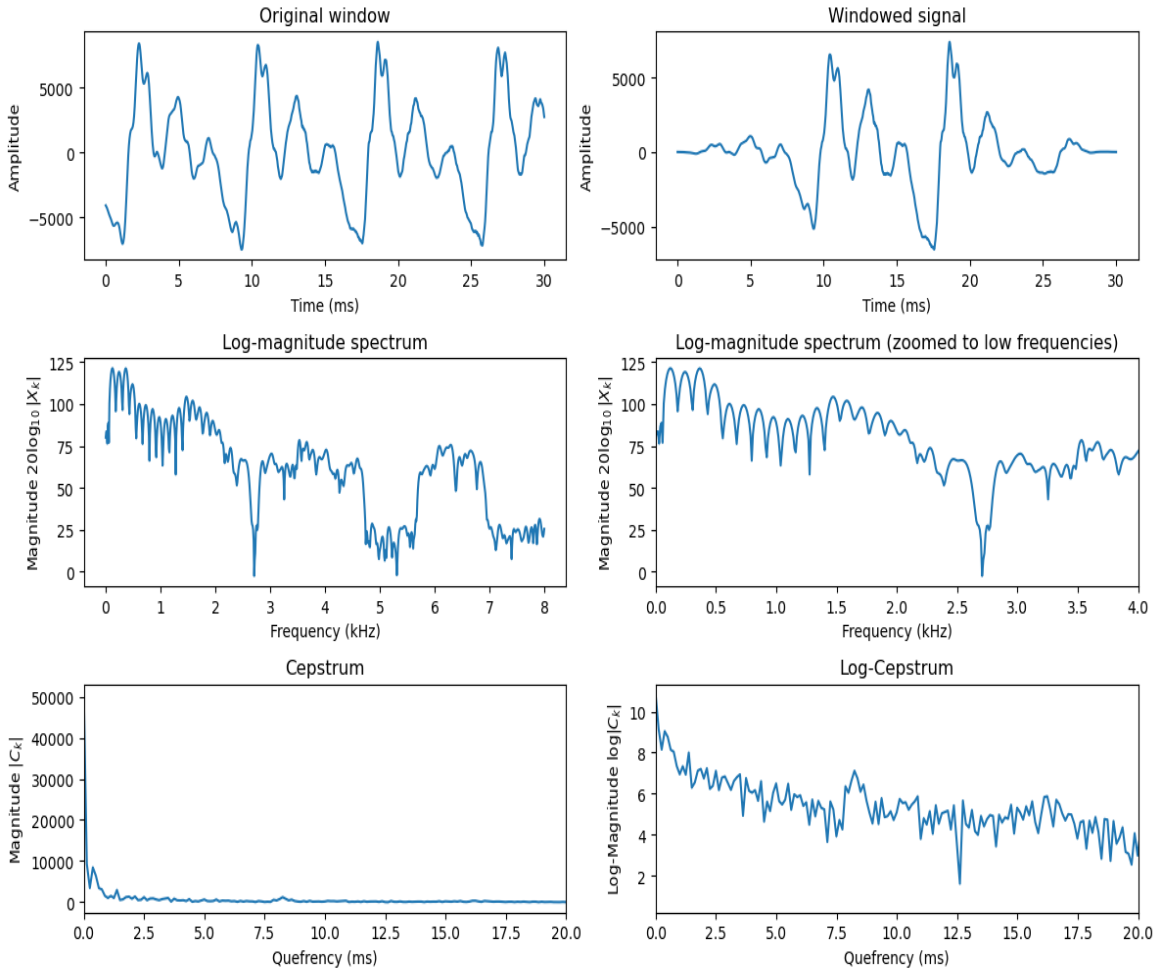


* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Spectrogram_and_the_STFT.html

The cepstrum, mel-cepstrum and mel-frequency cepstral coefficients (MFCC)

The algorithm is:

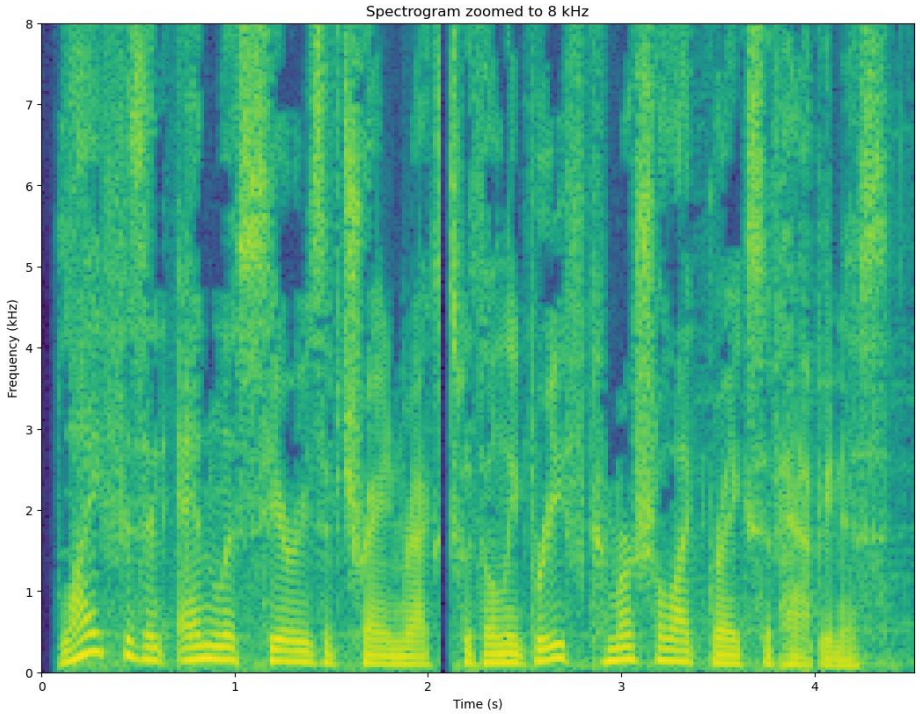
- Apply analysis **windowing** to signal
- Apply **time-frequency transform** (DFT or DCT)
- Take the **logarithm** (log + Mel scale) of the absolute value
- Apply second **time-frequency transform**



* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Melcepstrum.html?highlight=mfcc

The cepstrum, mel-cepstrum and mel-frequency cepstral coefficients (MFCC)

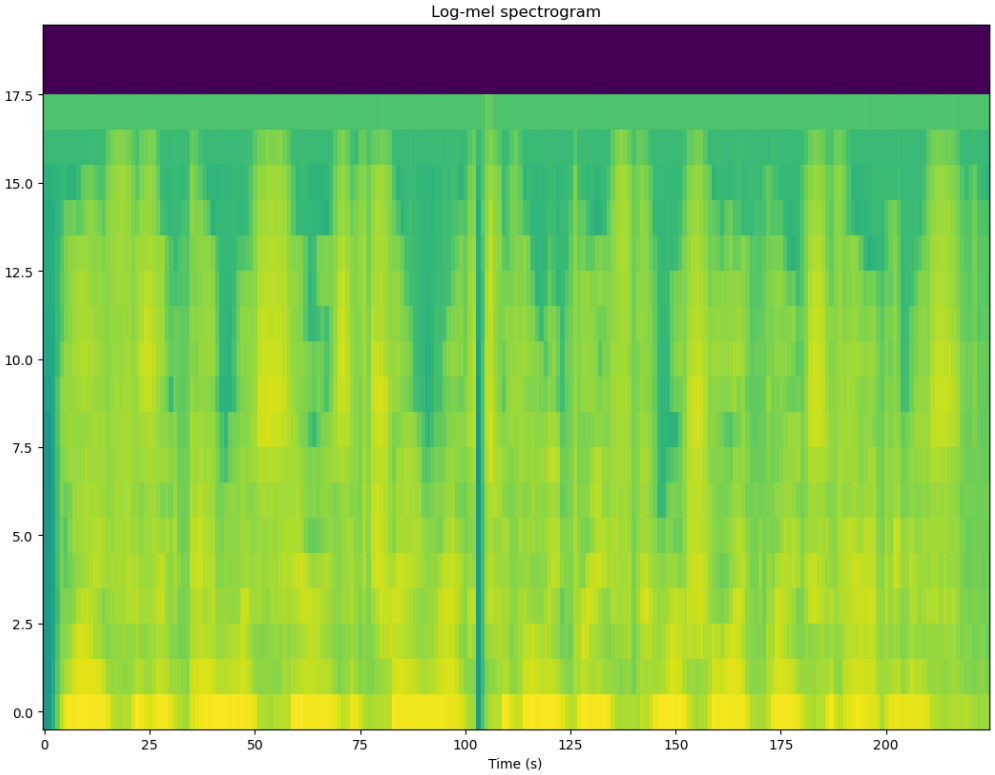
We calculate, for each segment, the magnitude of the Fast Fourier Transform (FFT). These are then concatenated in order to obtain the spectrogram. Each frequency bin of the spectrum is normalized by using the mean and standard deviation of the sample



* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Melcepstrum.html?highlight=mfcc

The cepstrum, mel-cepstrum and mel-frequency cepstral coefficients (MFCC)

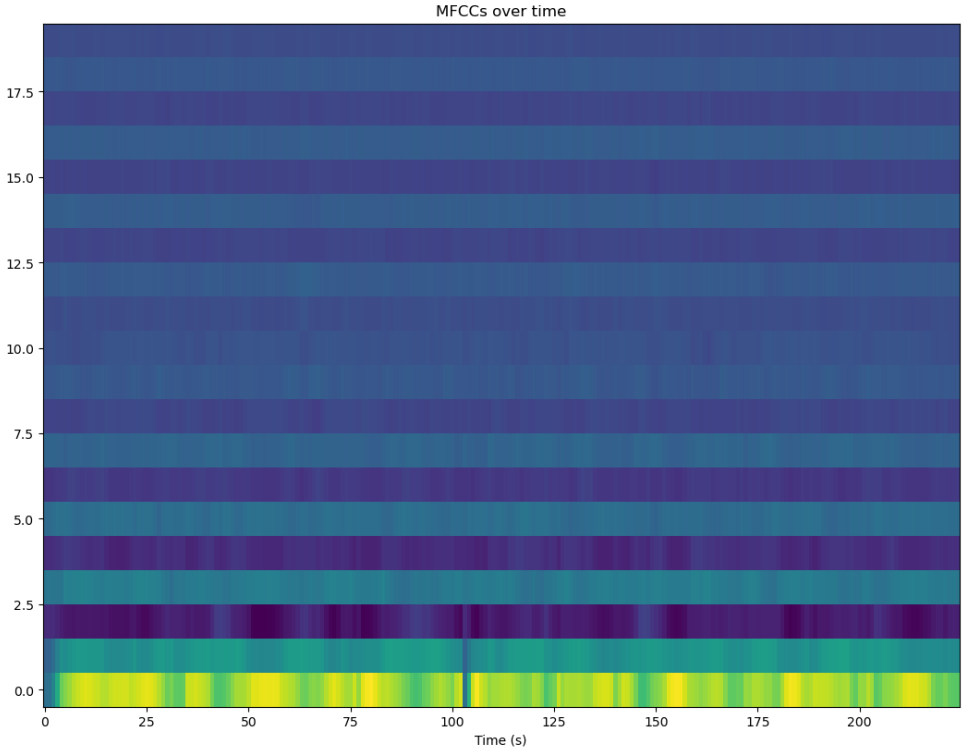
The spectrogram power is processed with a filter bank made of triangular band pass filters. This operation maps the power of the spectrogram onto the Mel scale. The Mel-spectrum is the log of the output of the filters



* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Melcepstrum.html?highlight=mfcc

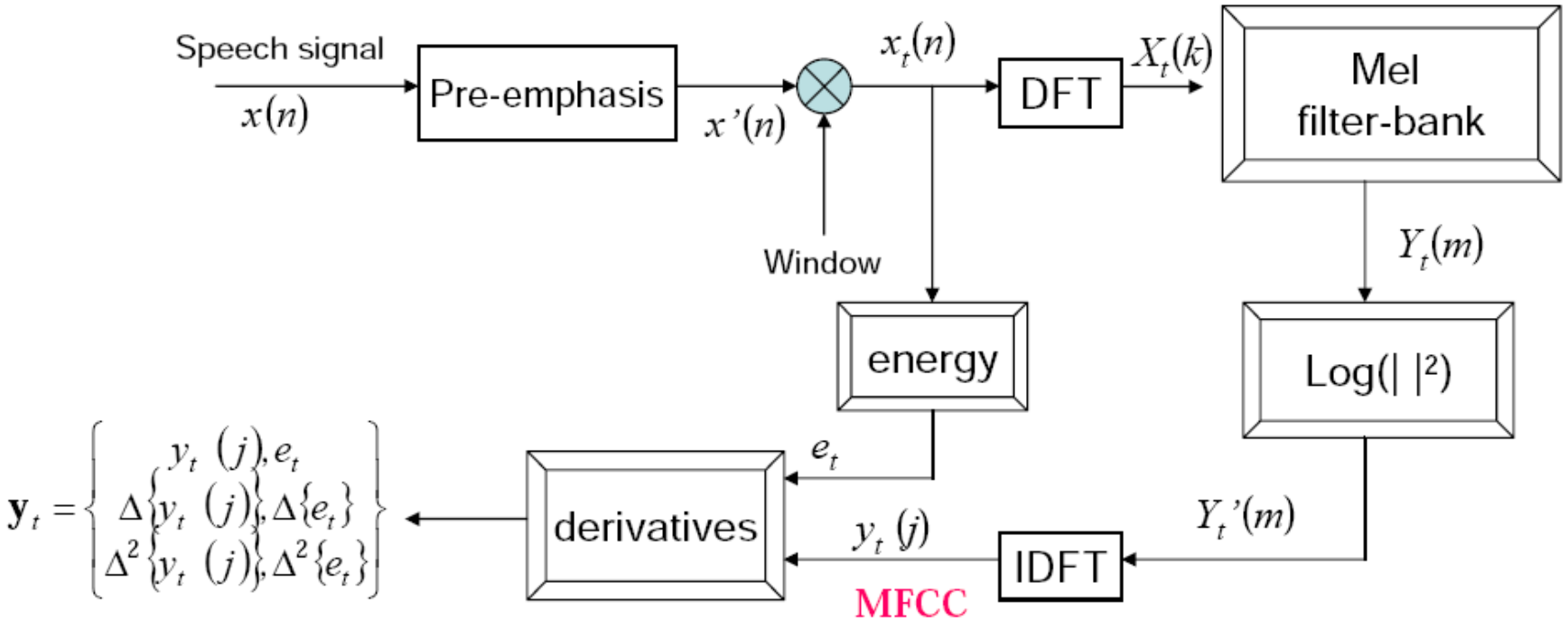
The cepstrum, mel-cepstrum and mel-frequency cepstral coefficients (MFCC)

The inverse Discrete Cosine Transform (IDCT) is applied to the Mel-power spectrum, obtaining the cepstrum and then the log cepstrum. The MFCCs are the first *N* mel frequency cepstral coefficients.



* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Melcepstrum.html?highlight=mfcc

The cepstrum, mel-cepstrum and mel-frequency cepstral coefficients (MFCC)



* See **additional materials** on https://speechprocessingbook.aalto.fi/Representations/Melcepstrum.html?highlight=mfcc

Wake-word and keyword detection

Wake-word and keyword detection

details

- **Wake-word** and **keyword** spotting refer to **small-vocabulary** speech **recognition** tasks. They are used either in:
  - very simple applications where **proper speech recognition** is **unnecessarily complex** (keyword spotting),
  - in **pre-processing tasks**, where we want to save resources by waiting for a “Hey computer!”. Wake-word spotting is thus a **trigger** for **more complex speech processing** tasks.

- Most typically wake-word and keyword spotting algorithms run on **devices** with **limited resources**. They can be limited in **memory footprint** and in **computation resources** (CPU power) or **often both**.

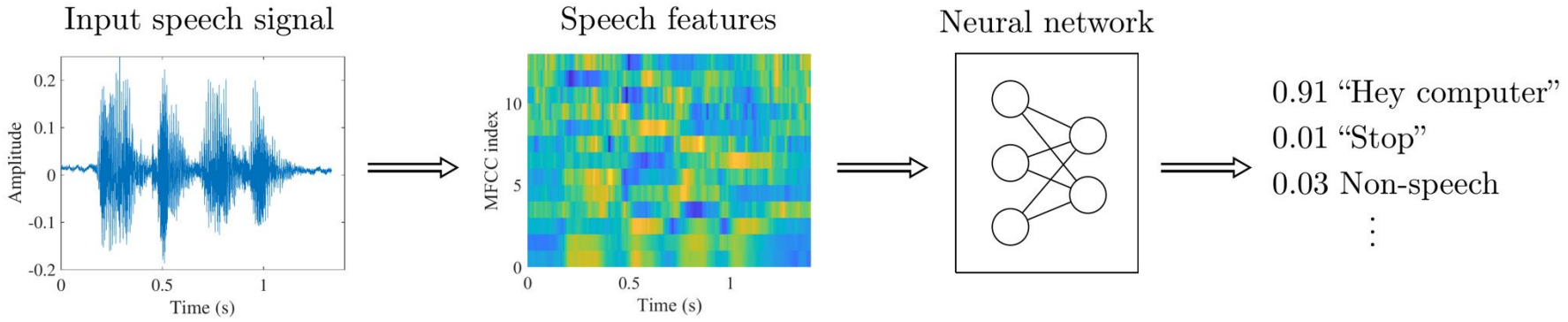
* See **additional materials** on https://speechprocessingbook.aalto.fi

Wake-word and keyword detection

pipeline

Pipeline

The input speech signal is first converted to a **feature representation**, such as **MFCCs**, which are fed to a **neural network**, and the output is the **likelihood** of each **keyword**.



* See **additional materials** on https://speechprocessingbook.aalto.fi

Wake-word and keyword detection

pipeline

- A **central challenge** in training keyword spotting algorithms is finding and **choosing training data**.
- To get good quality, you would typically need several **tens of thousands** of utterances of the keywords, spoken by a large range of **different speakers** and in different environments.
- For example, the "**Speech Commands Dataset**" by Google has 65.000 utterances of 30 short words. The dataset is designed to let you build basic but useful voice interfaces for applications, with common words like "**Yes**", "**No**", **digits**, and **directions** included.

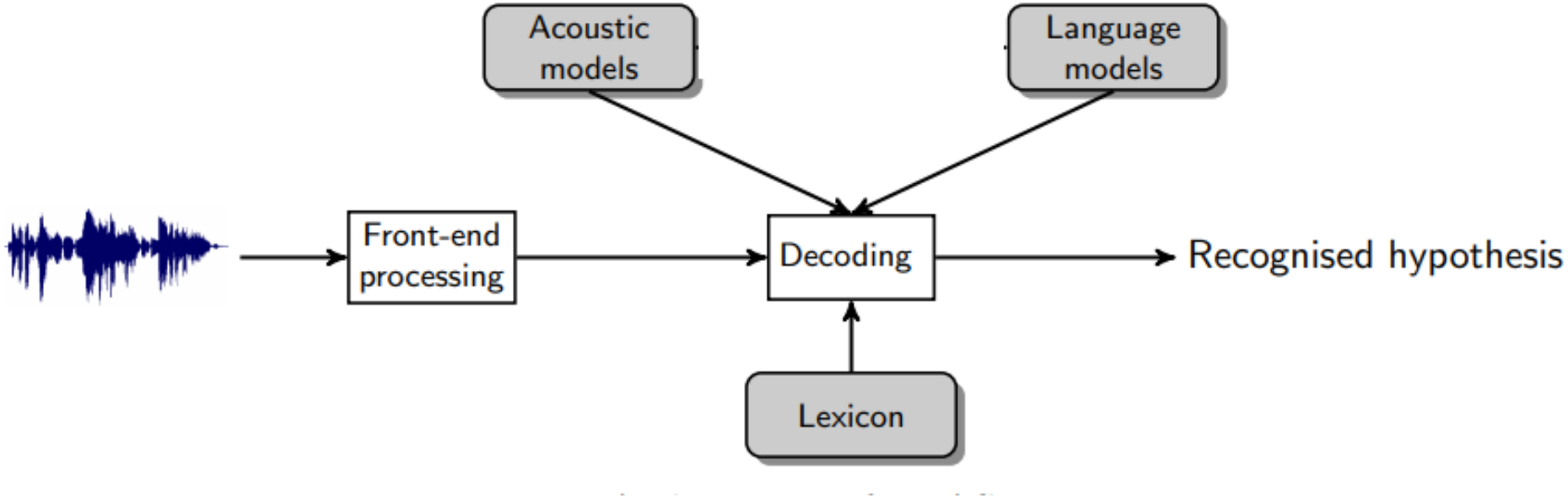
* See **additional materials** on https://blog.research.google/2017/08/launching-speech-commands-dataset.html

Speech Recognition

Speech recognition

pipeline

The problem of **speech recognition** is defined as the conversion of **spoken utterances** into **textual sentences** by a **machine**.

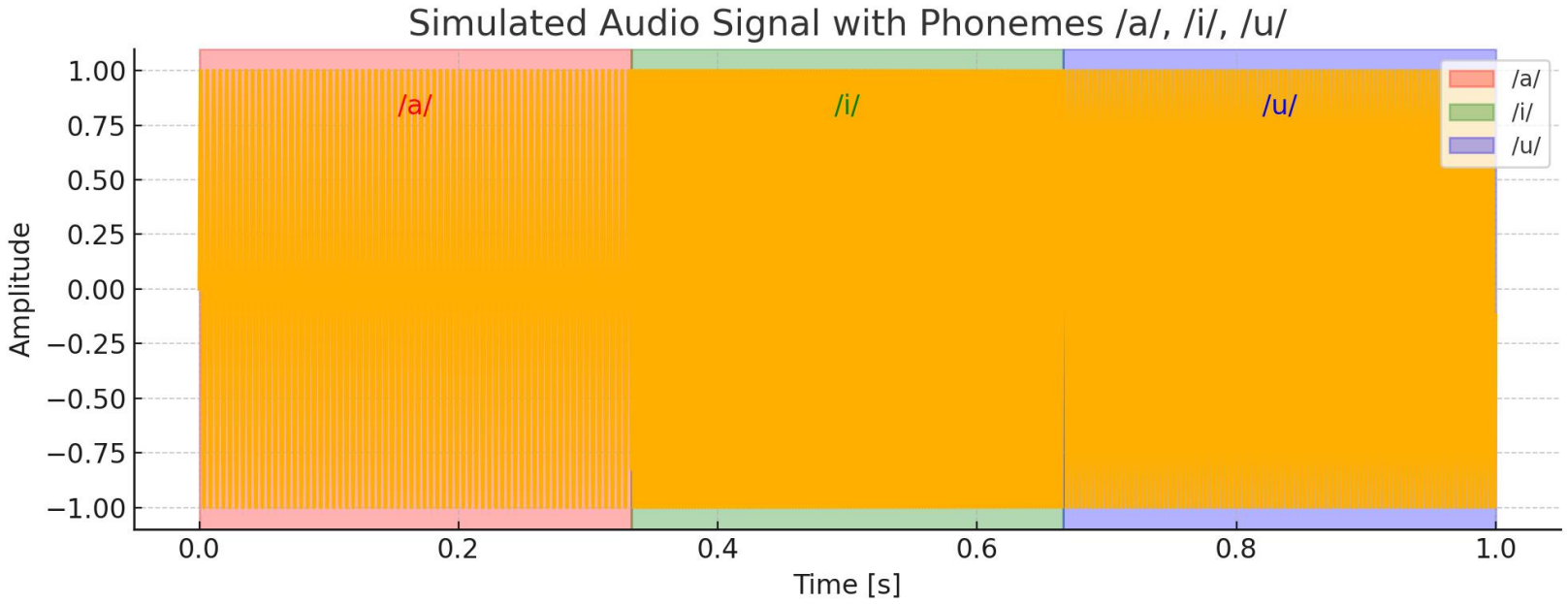


Speech recognition

pipeline

Feature Extraction: It converts the speech signal into a sequence of acoustic feature vectors. These observations should be compact and carry sufficient information for recognition in the later stage.

Acoustic Model: It contains a statistical representation of the distinct sounds that make up each word in the Language Model or Grammar. Each distinct sound corresponds to a phoneme.

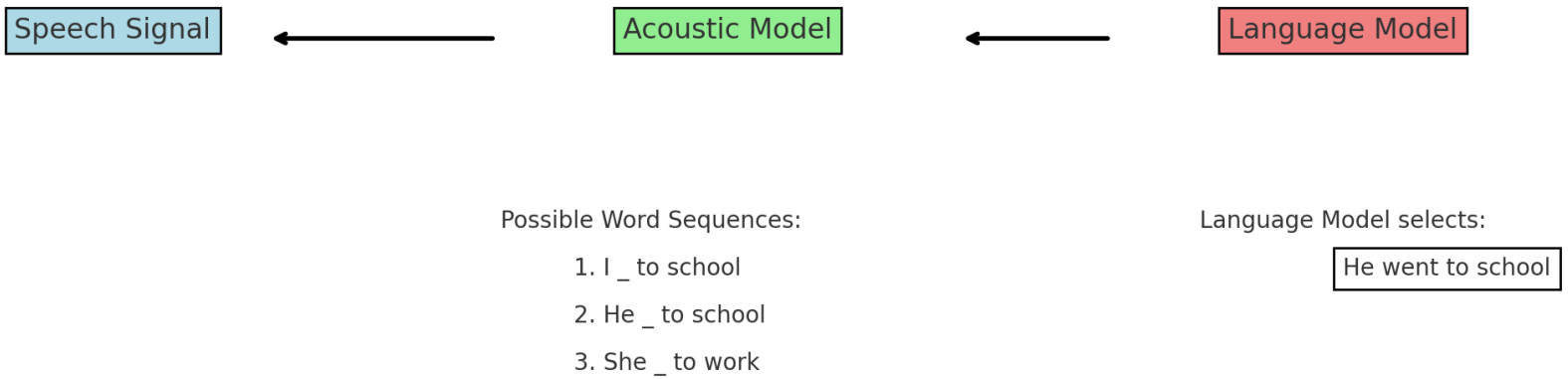


Speech recognition

pipeline

Language Model: It contain a very large list of words and their probability of occurrence in a given sequence.

Decoder: It is a software program that takes the sounds spoken by a user and searches the acoustic model for the equivalent sounds. When a match is made, the decoder determines the phoneme corresponding to the sound. It keeps track of the matching phonemes until it reaches a pause in the users speech. It then searches the language model for the equivalent series of phonemes. If a match is made, it returns the text of the corresponding word or phrase to the calling program.



Speech recognition

pipeline

Speech recognition systems can be classified on the basis of the **constraints** under which they are developed and which they consequently impose on their users.

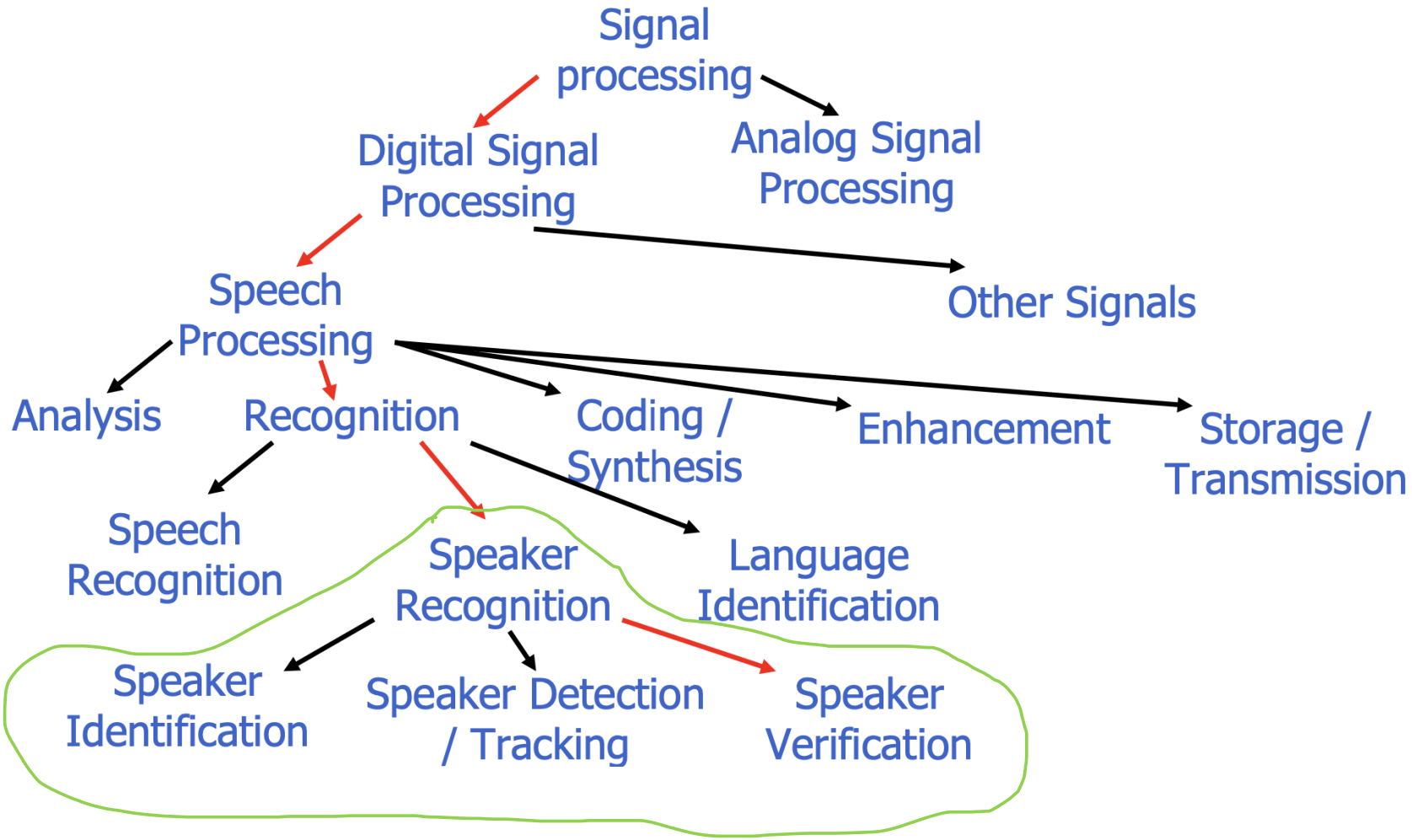
These constraints include:

- *Speaker dependence*
- *Type* of *Utterance*
- *Vocabulary Size*
- *Type* of *speech*
- *Environment*

Speaker Recognition and Verification

Speaker recognition and verification

pipeline



Partially taken from Sharat.S.Chikkerur

Speaker recognition and verification

pipeline

Speaker recognition is the task of identifying a speaker using their voice. Speaker recognition is classified into two parts:

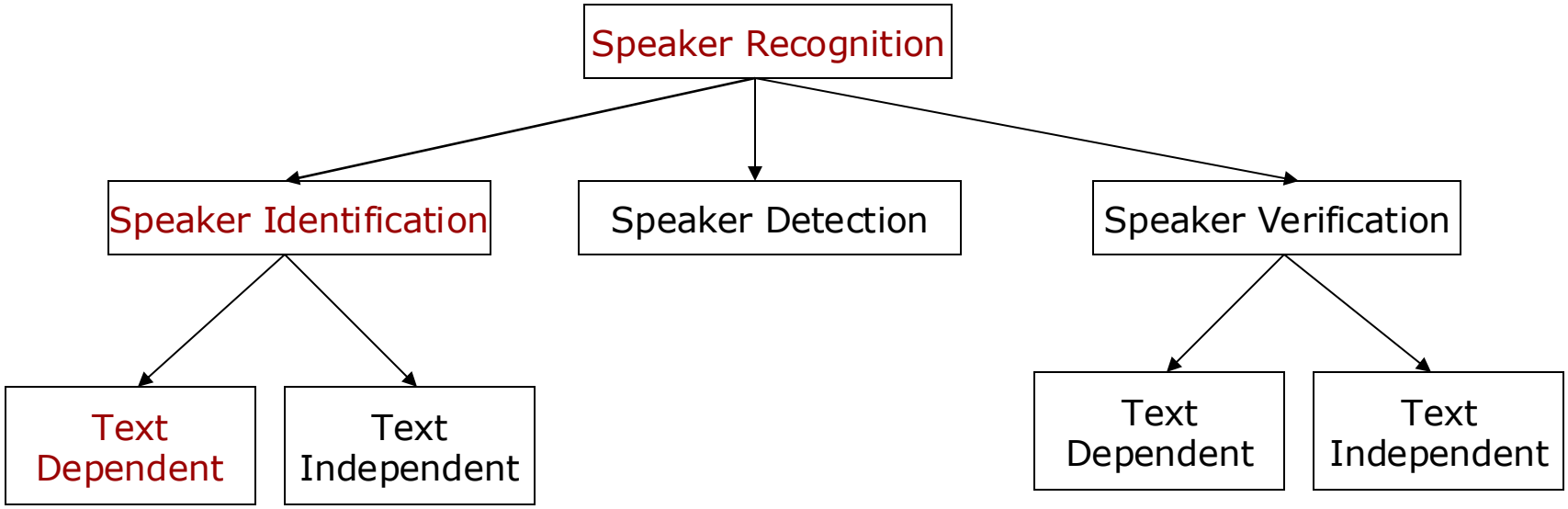
- **speaker identification** is the process of determining which voice in a group of known voices best matches the speaker,
- **speaker verification** is the task of accepting or rejecting the identity claim of a speaker by analyzing their acoustic samples.

- See **additional materials** on https://speechprocessingbook.aalto.fi/Recognition/Speaker_Recognition_and_Verification.html
- Bai, Z., & Zhang, X. L. (2021). Speaker recognition based on deep learning: An overview. *Neural Networks*, *140*, 65-99.

Speaker recognition and verification

pipeline

Speaker recognition is the task of identifying a speaker using their voice. Speaker recognition is classified into two parts:



Partially taken from Sharat.S.Chikkerur

Speaker recognition and verification

pipeline

Text-dependent recognition

- Recognition system knows text spoken by person
- Examples: fixed phrase, prompted phrase
- Used for applications with strong control over user input
- Knowledge of spoken text can improve system performance

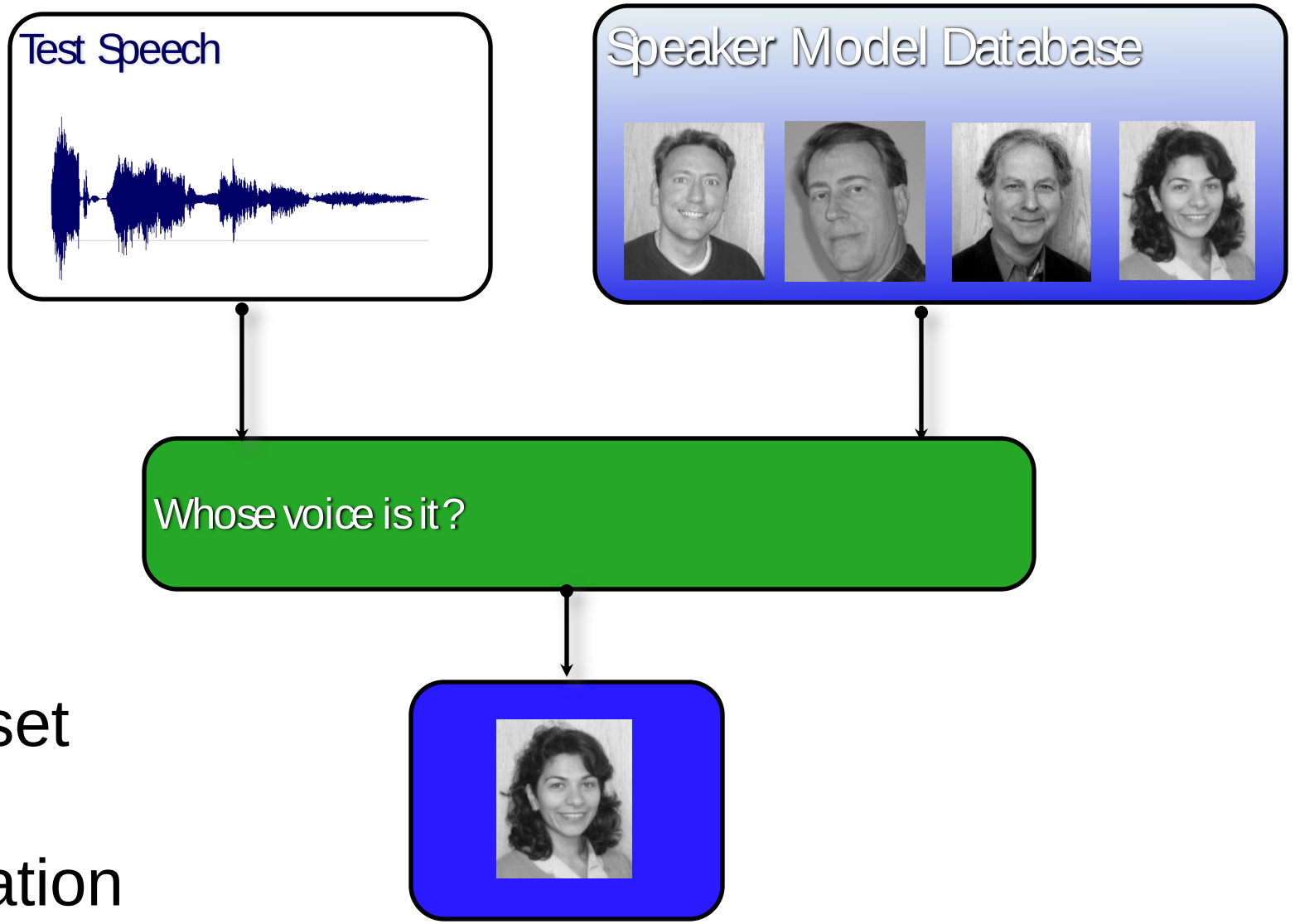
Text-independent recognition*

- 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
- Speech recognition can provide knowledge of spoken text

Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline



Closed-set Speaker Identification

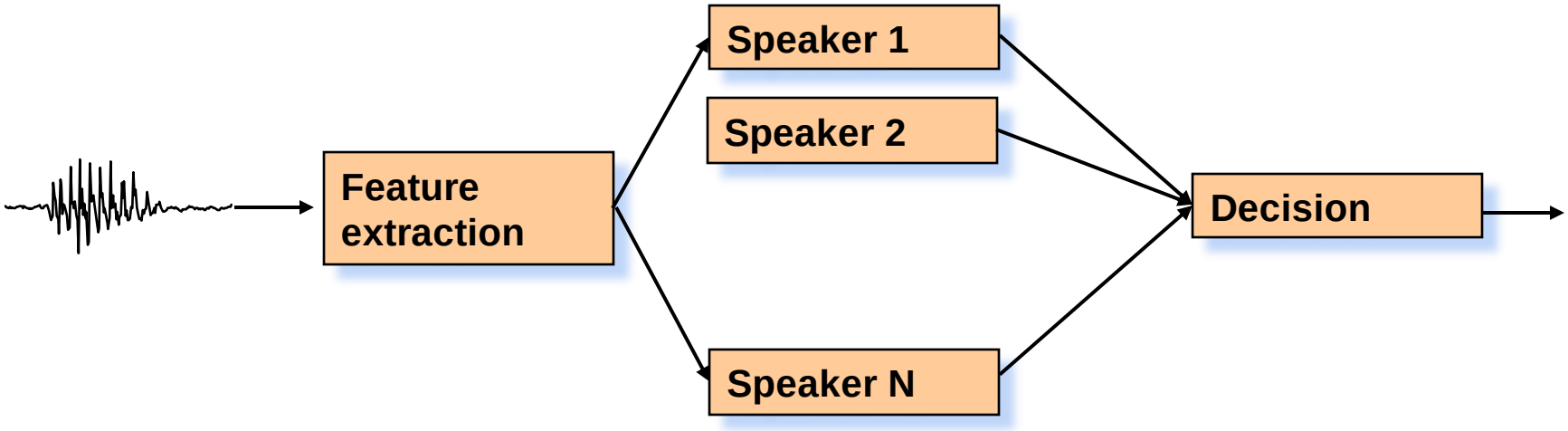
Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline

Selection between a set of known voices

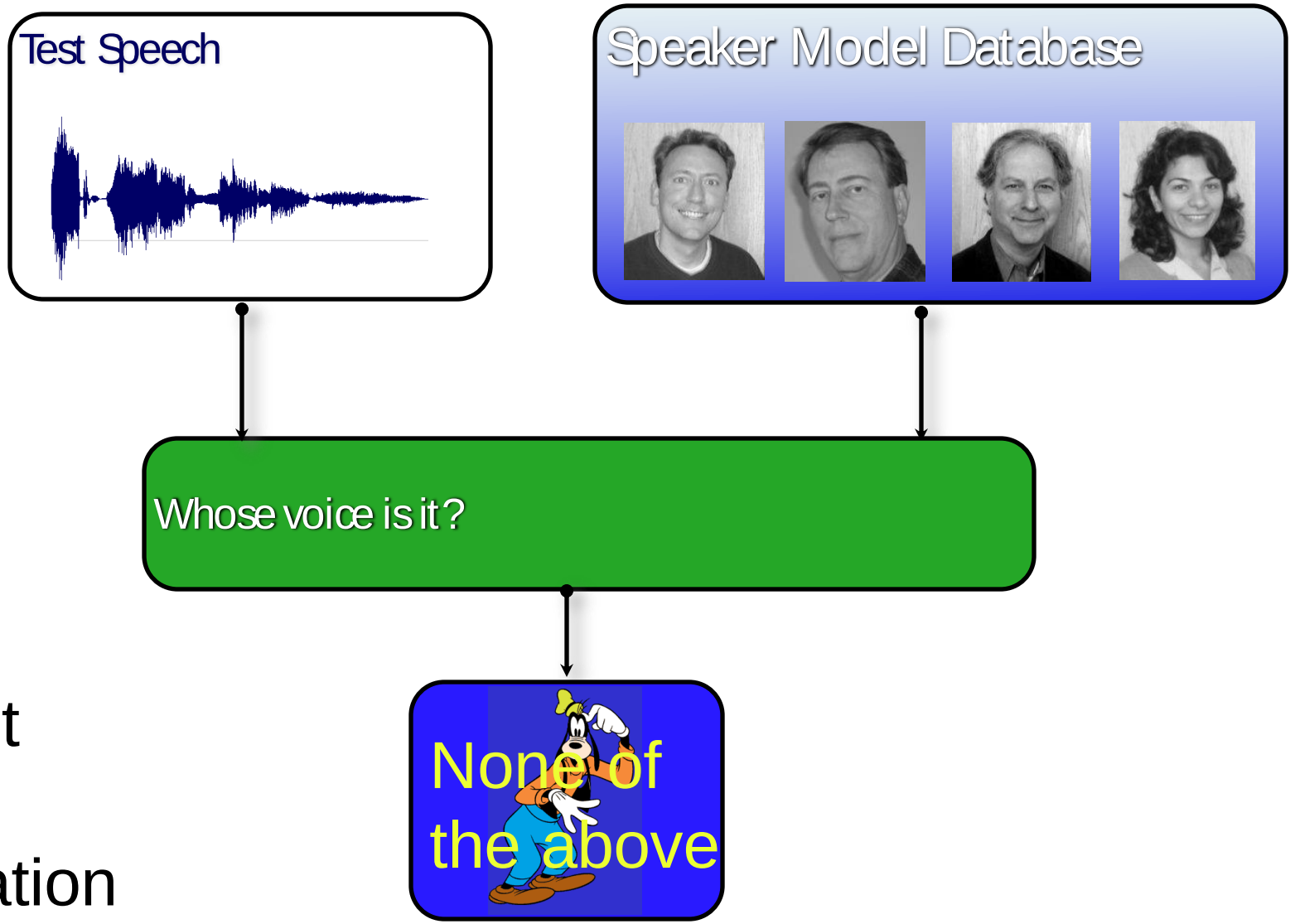
Identification: pick model (of N) with best score



Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline

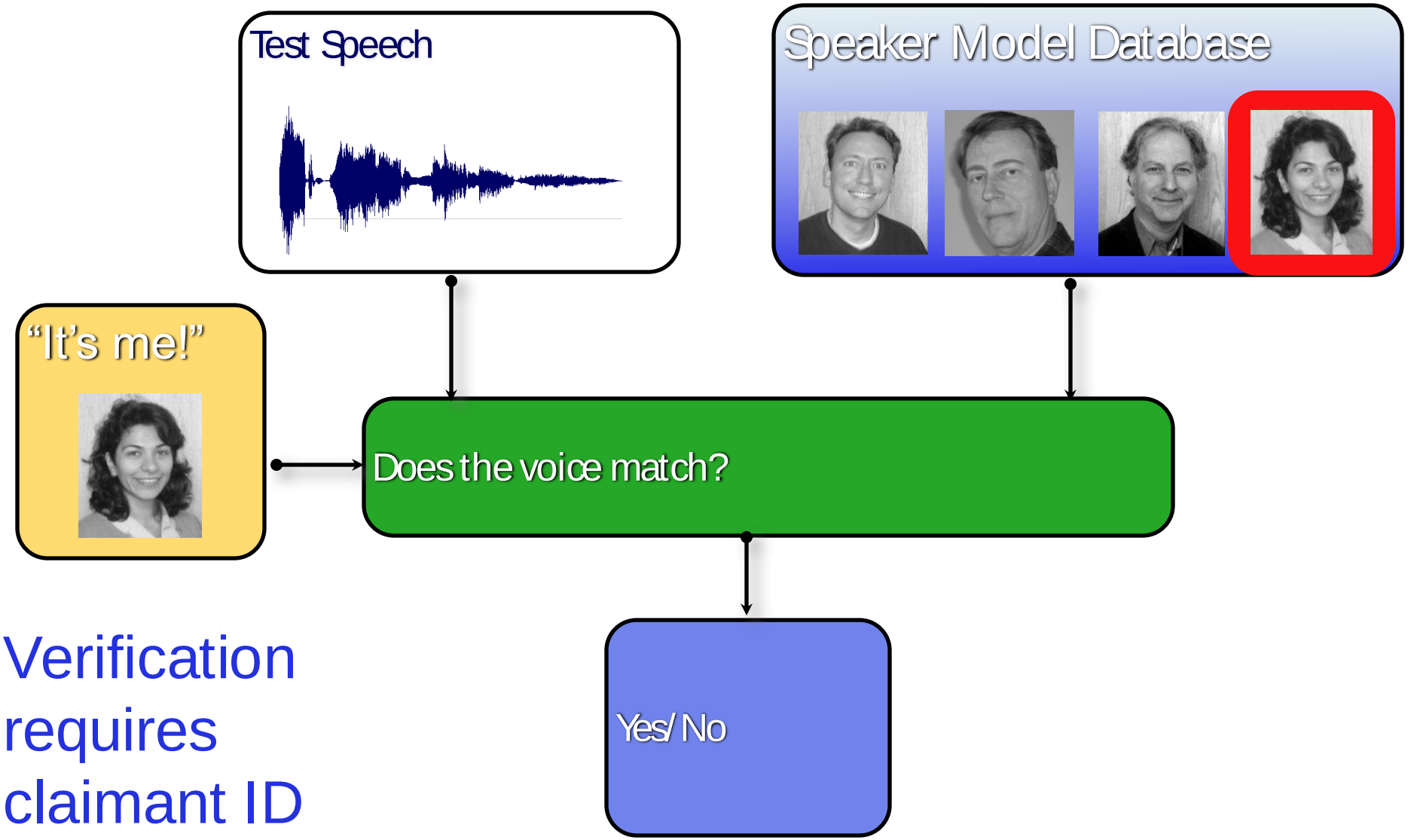


Open-set Speaker Identification

Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline



Verification requires claimant ID

Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline

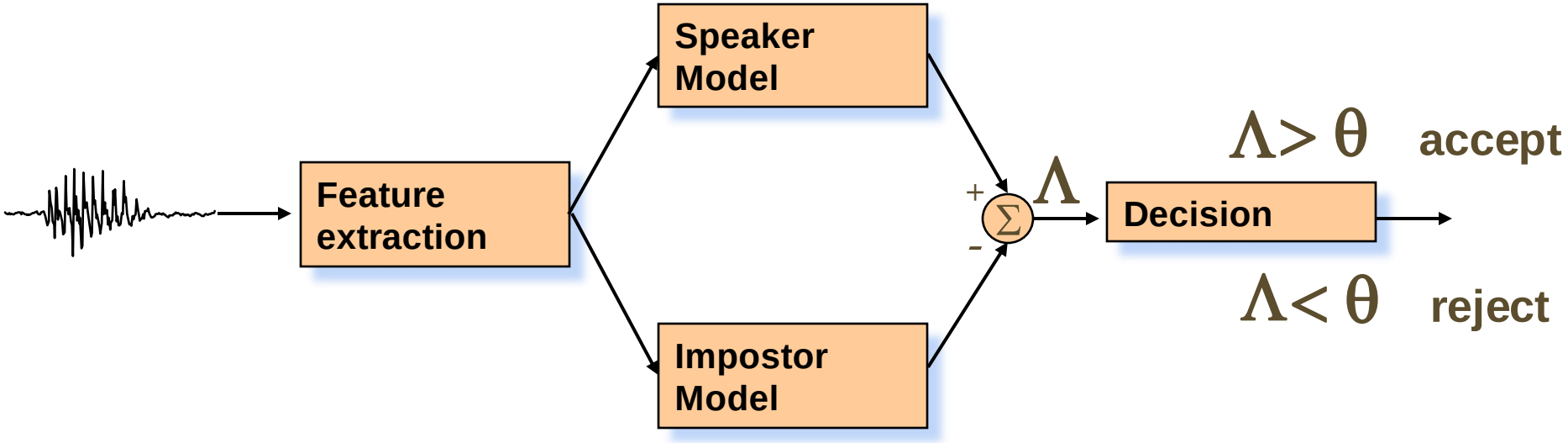
Verification decision approaches have roots in signal detection theory

- **2-class Hypothesis test:**

  H0: the speaker is an impostor
  H1: the speaker is indeed the claimed speaker.

- **Statistic computed on test utterance S as likelihood ratio:**

$$\Lambda = \log \frac{\text{Likelihood } \mathbf{S} \text{ came from speaker model}}{\text{Likelihood } \mathbf{S} \text{ did } \underline{\text{not}} \text{ come from speaker model}}$$



Partially taken from Nikki Mirghafori

Speaker recognition and verification

pipeline

Verification decision approaches have roots in signal detection theory

- **Verification**: usual approach is via likelihood ratio tests, hypothesis testing, i.e.:
  - By Bayes:

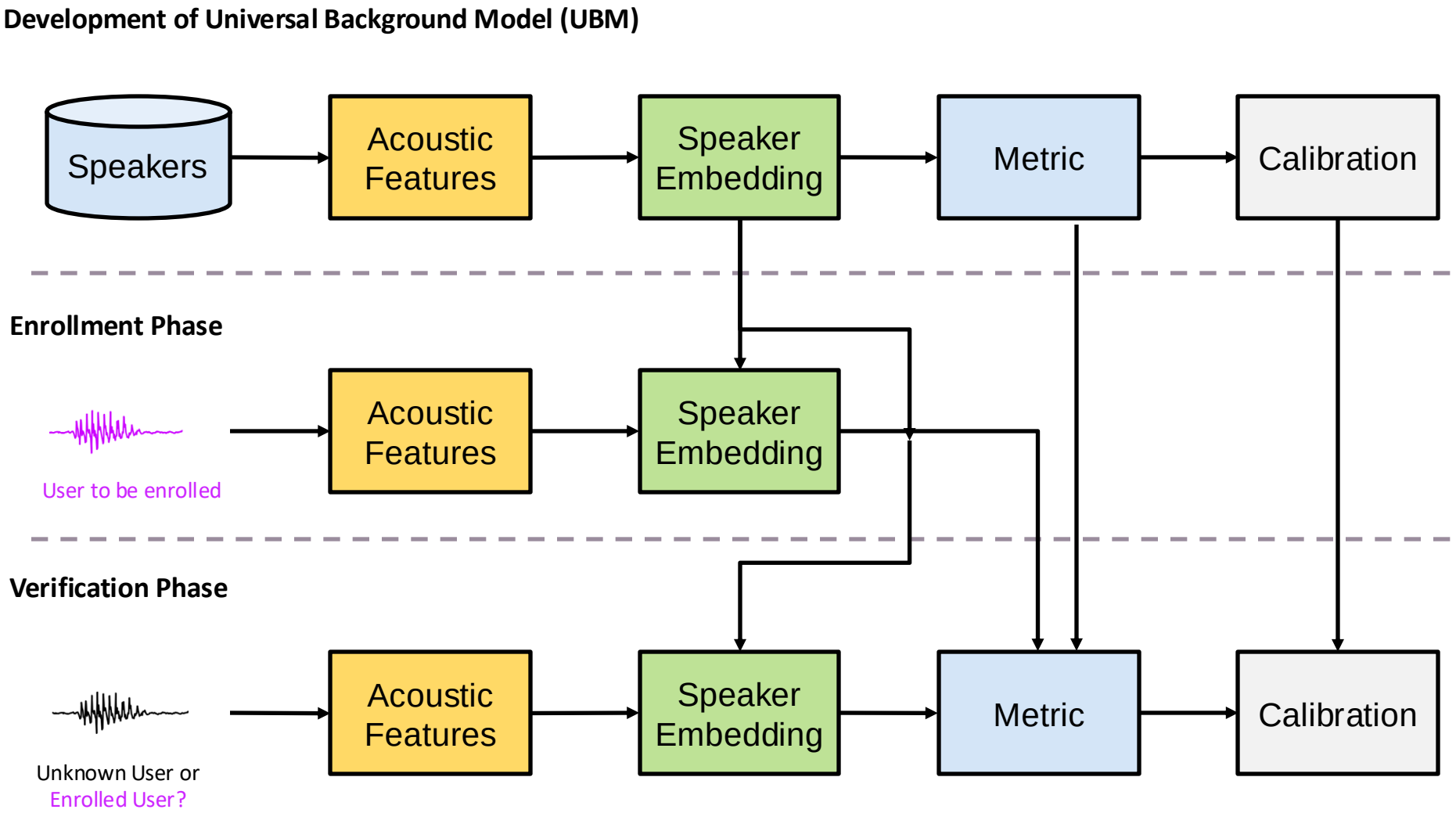
$$\frac{P(target|x)}{P(nontarget|x)} = \frac{P(x|target)P(target)}{P(x|nontarget)P(nontarget)}$$

  - accept if > *threshold*, *reject* otherwise
  - Can't sum over all non-target talkers -- world for SV!
    - Use "**cohorts**" (collection of impostors)
    - Train "**universal**"/"**world**"/"**background**" model (speaker independent, it's trained on many speakers)

Partially taken from Nikki Mirghafori

Speaker verification

pipeline



Partially taken from Sharat.S.Chikkerur

Speaker recognition and verification

Acoustic features

Spectral Features represent how the energy of different frequencies changes over time: Mel-Frequency Cepstral Coefficients (MFCCs), Spectrogram, Formants.

Temporal Features capture information related to the timing and duration of speech sounds: Zero Crossing Rate, Speech Rate, Speech Activity Detection.

Pitch and Prosody Features are related to the pitch (fundamental frequency) and prosodic aspects of speech. They capture information about intonation, stress, and rhythm: Fundamental Frequency, Pitch Contours, Pitch Periodicity.

Energy features measure the magnitude of the audio signal: Short-Term Energy, Long-Term Average Energy (LTAE), Root Mean Square Energy.

Linear Predictive Coding (LPC) Features is a method for modeling the spectral envelope of speech. LPC coefficients are used to capture the resonant characteristics of the vocal tract: LPC Coefficients, Reflection Coefficients.

Speaker recognition and verification

Acoustic features

Cepstral Coefficients represent the smoothed spectrum of speech and are derived from the Fourier transform of the log spectrum: MFCCs are a common example.

Higher-Level Features capture linguistic and phonetic information in speech: Phoneme or Phone-based features, Prosodic Features (e.g., speaking rate), Syllable-Based Features.

Time-Domain Features are derived directly from the time-domain representation of audio signals: Autocorrelation Coefficients, Temporal Zero Crossing Rate

Statistical Features involve statistical measures of the distribution of values within a speech segment: Mean, Variance, Skewness, and Kurtosis, Percentile-Based Features

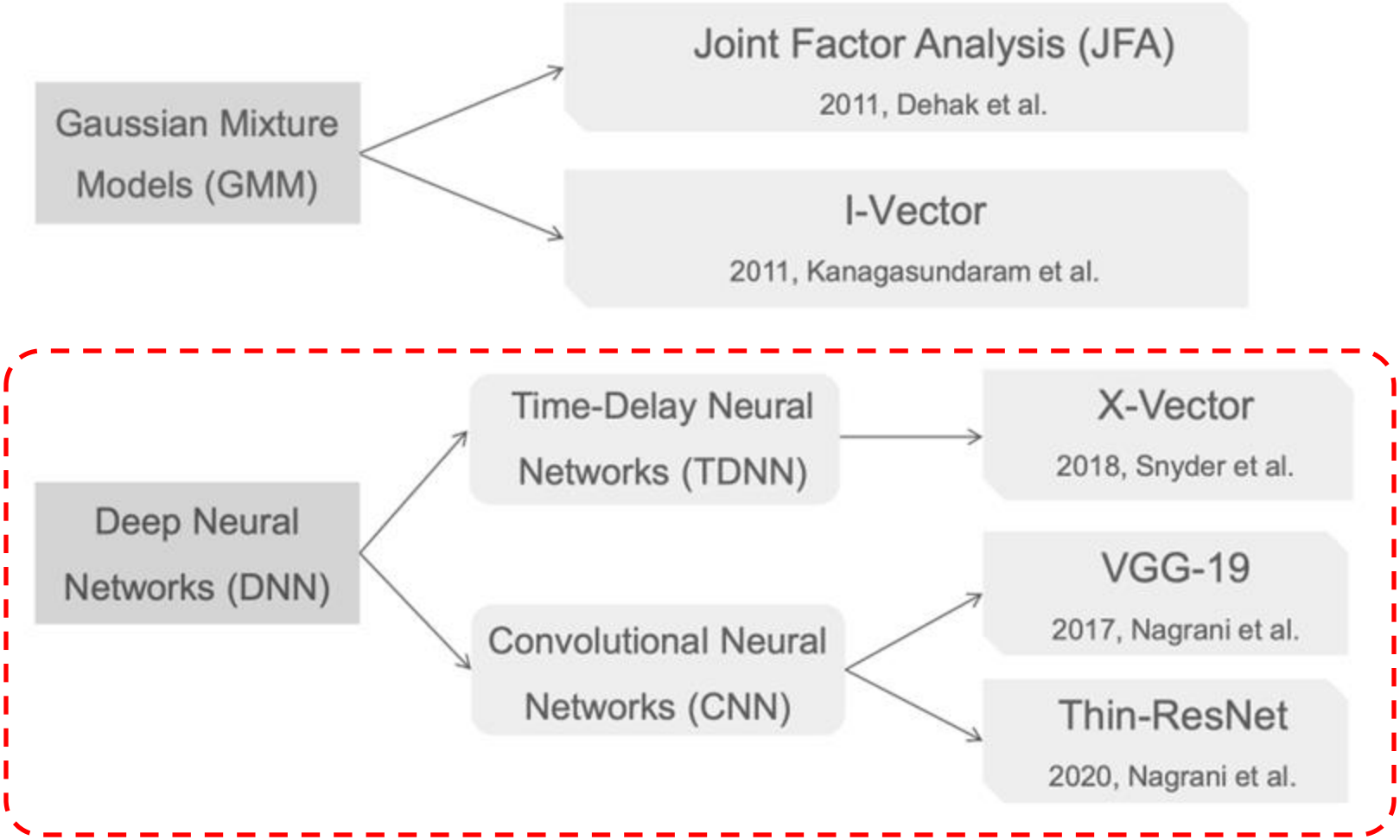
Speaker recognition and verification

Acoustic features

Delta and Delta-Delta Features are used to capture temporal dynamics by calculating the rate of change of other feature values over time. They are often applied to features like MFCCs, PLP coefficients, or spectral features.

Perceptual Features aim to model how humans perceive sound and include features like: Perceptual Linear Prediction (PLP) coefficients, Auditory Spectrogram-Based Features

Speaker embedding



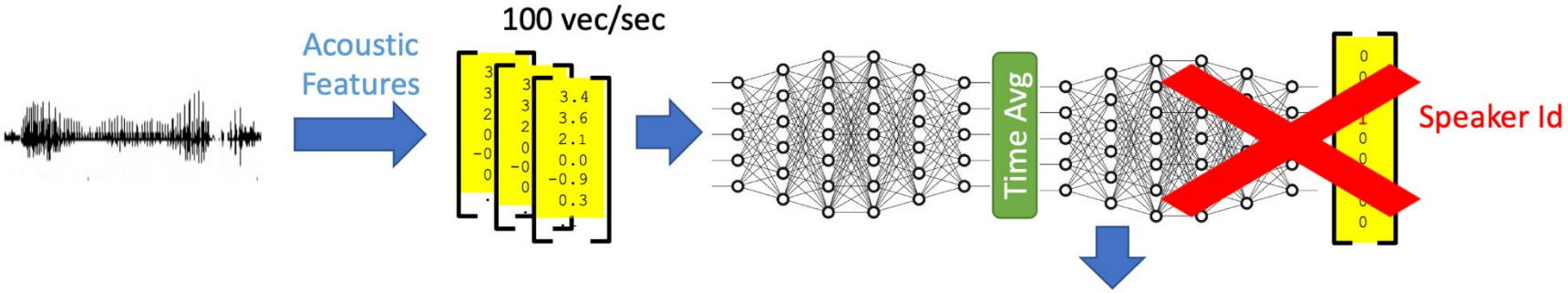
Slide taken from Mirko Marras (**Exploring Algorithmic Fairness in Deep Speaker Verification**)

Speaker recognition and verification

embedding

Speaker embedding with DNNs

- Transform variable length recording into a single vector – Embedding
- Embedding retains the speaker identity information



- Train the network to classify speakers

  Large dataset > 1K speakers
- After training:

  Take intermediate layer as speaker embedding

Partially taken from Jesus Villalba

Speaker embedding with DNNs

- One of the most used dataset for speaker recognition and verification is the **VoxCeleb2** dataset.
- VoxCeleb2 contains over **1 million utterances** for **6,112 celebrities**, extracted from videos uploaded to **YouTube**. The development set of VoxCeleb2 has no overlap with the identities in the VoxCeleb1 or SITW data

	dev	test
# of speakers	5,994	118
# of videos	145,569	4,911
# of utterances	1,092,009	36,237

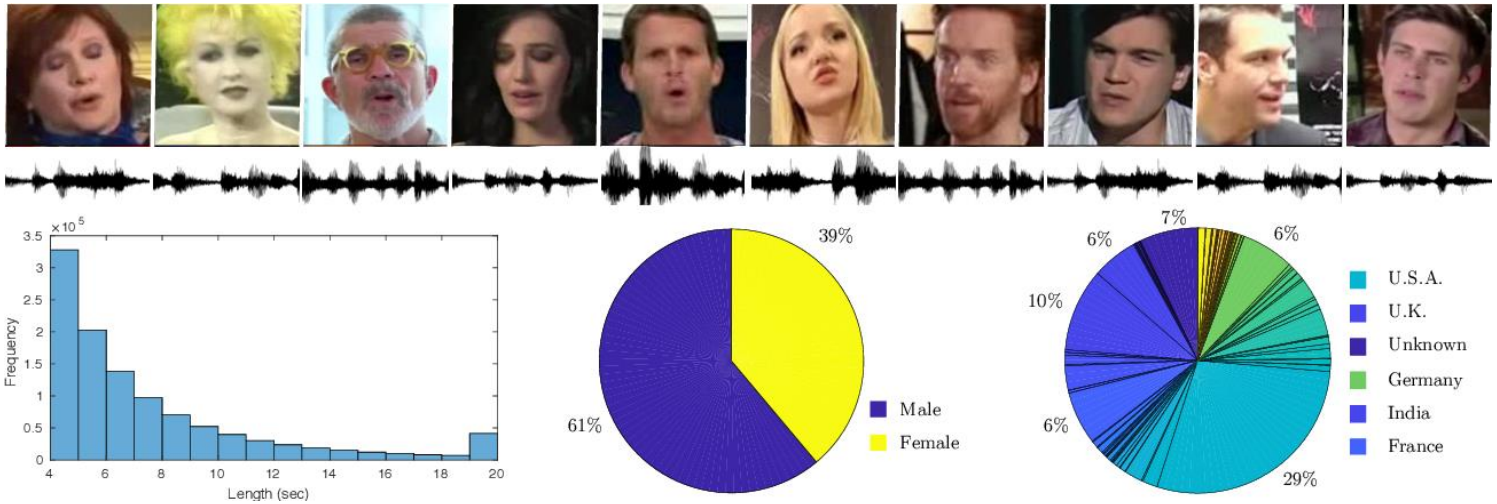
More information here https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox2.html

Speaker recognition and verification

embedding

Speaker embedding with DNNs

- One of the most used dataset for speaker recognition and verification is the **VoxCeleb2** dataset.
- VoxCeleb2 contains over **1 million utterances** for **6,112 celebrities**, extracted from videos uploaded to **YouTube**. The development set of VoxCeleb2 has no overlap with the identities in the VoxCeleb1 or SITW data



More information here https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox2.html

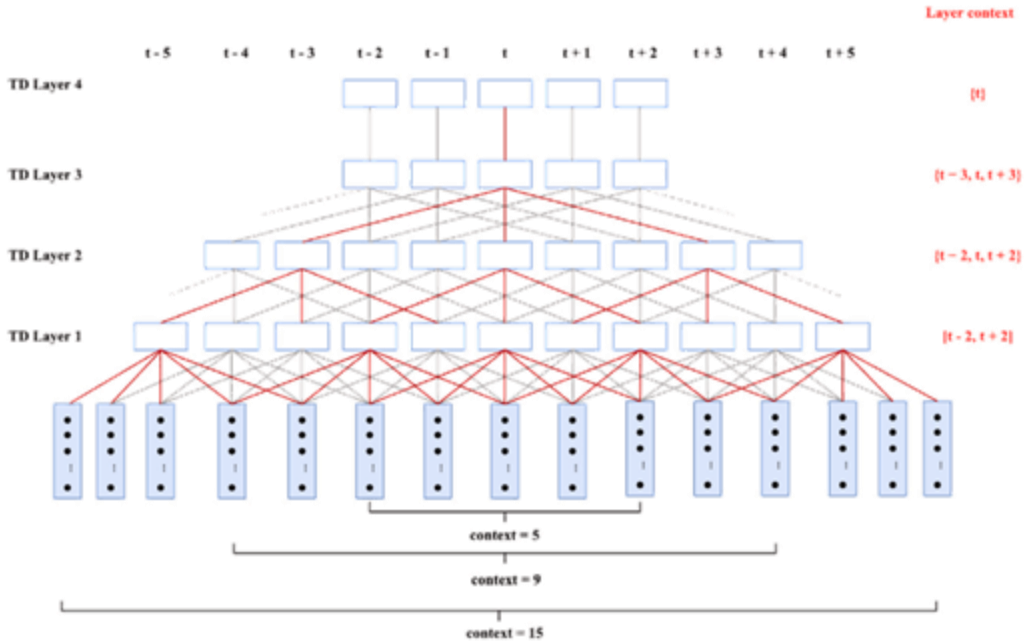
Speaker recognition and verification

embedding

x-vectors are based on TDNN

The use of **longer sound sequences** is more effective than **short speech frames** (of less than half of the second), the phonetic content rather than speaker characteristics are the dominant source of variability. To work with the whole utterance a Time Delay Neural Network (TDNN) has been used and the resulting embedding is referred to as x-vector.

Basically is 1-D CNN



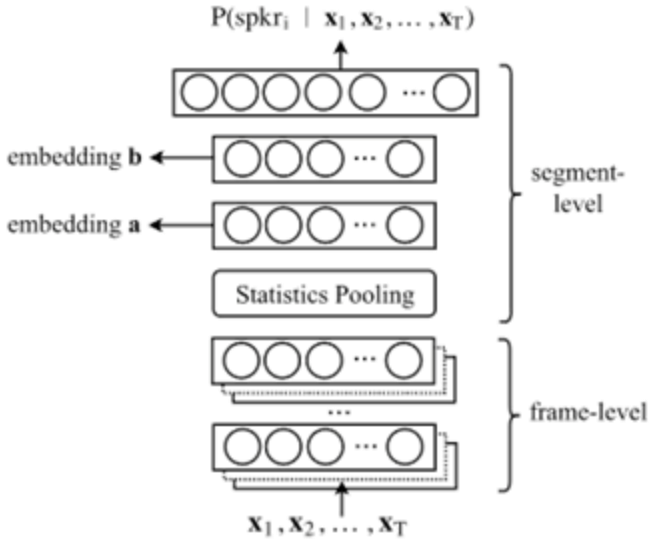
* See **additional materials** on Jakubec, M., Jarina, R., Lieskovska, E., & Kasak, P. (2024). Deep speaker embeddings for Speaker Verification: Review and experimental comparison. *Engineering Applications of Artificial Intelligence*, *127*, 107232.

Speaker recognition and verification

Neural embedding

x-vectors

- The features are 24 dimensional MFCC with a **frame-length** of 25ms, mean-normalized over a sliding window of up to 3 seconds
- Suppose an input segment has T **frames**. The first five layers operate on speech frames, with a small temporal context centered at the current frame t.
- *(For example, the input to layer frame3 is the spliced output of frame2, at frames $t-3$, t and $t+3$. This builds on the temporal context of the earlier layers, so that frame3 sees a total context of 15 frames.)*



Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

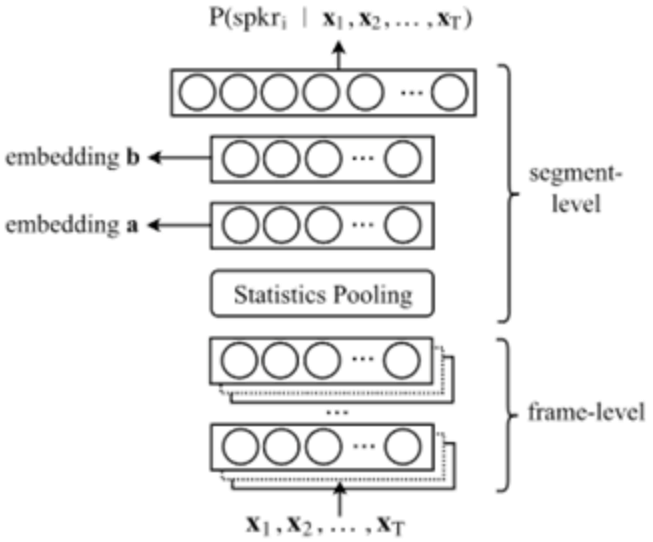
* See **additional materials** on Snyder, David, et al. "Deep Neural Network Embeddings for Text-Independent Speaker Verification." Interspeech. 2017.

Speaker recognition and verification

Neural embedding

x-vectors

- The *statistics pooling* layer aggregates all T frame-level outputs from layer *frame5* and computes its mean and standard deviation. The statistics are 1500 dimensional vectors, computed once for each input segment.
- The mean and standard deviation are concatenated together and propagated through segment-level layers and finally the softmax output layer.
- The DNN is trained to classify the N speakers in the training data. A training example consists of a chunk of speech features (about 3 seconds average), and the corresponding speaker label. After training, embeddings are extracted from the affine component of layer segment6.



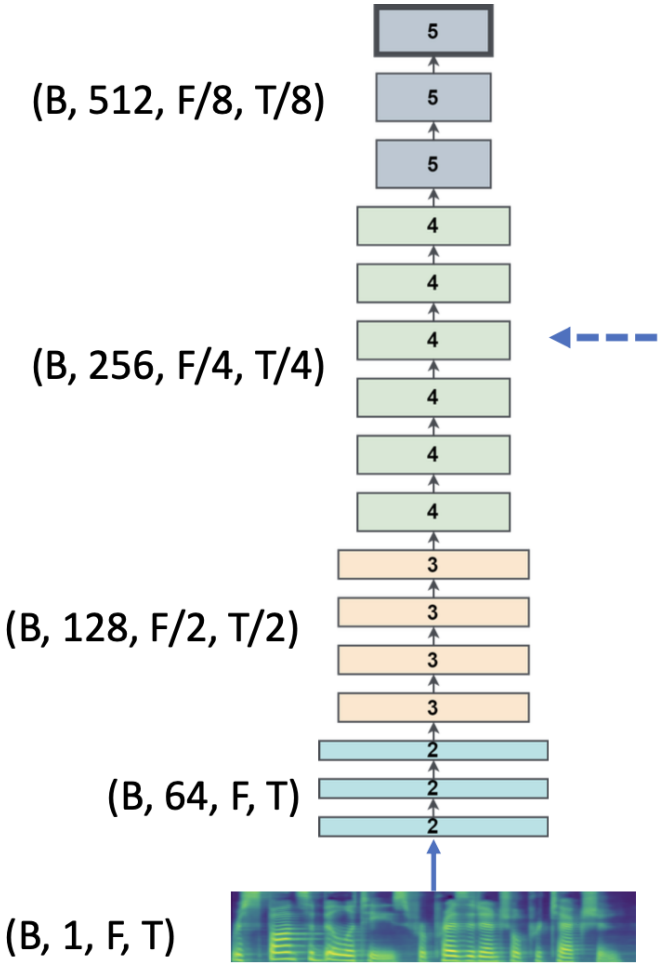
Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

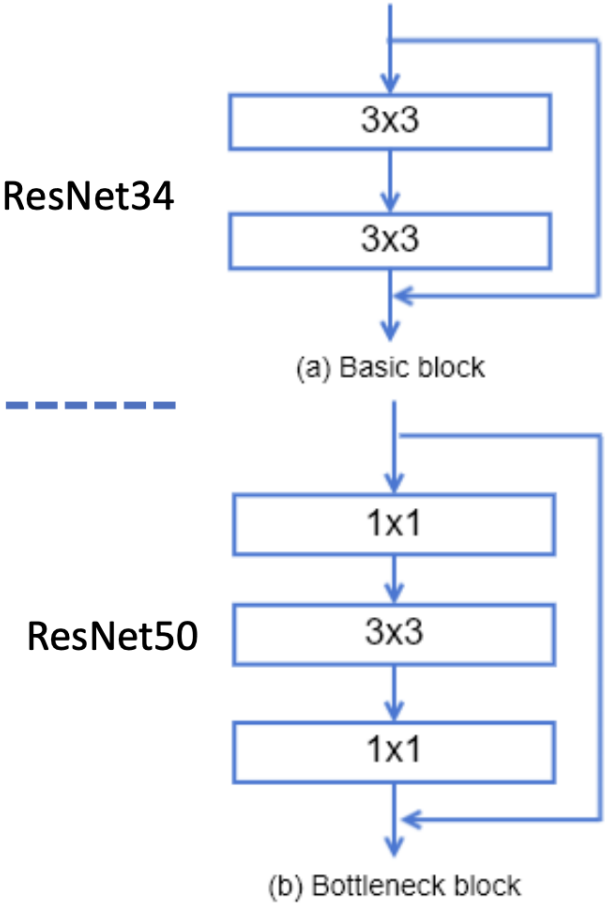
* See **additional materials** on Snyder, David, et al. "Deep Neural Network Embeddings for Text-Independent Speaker Verification." Interspeech. 2017.

Speaker recognition and verification

embedding

ResNet 2D



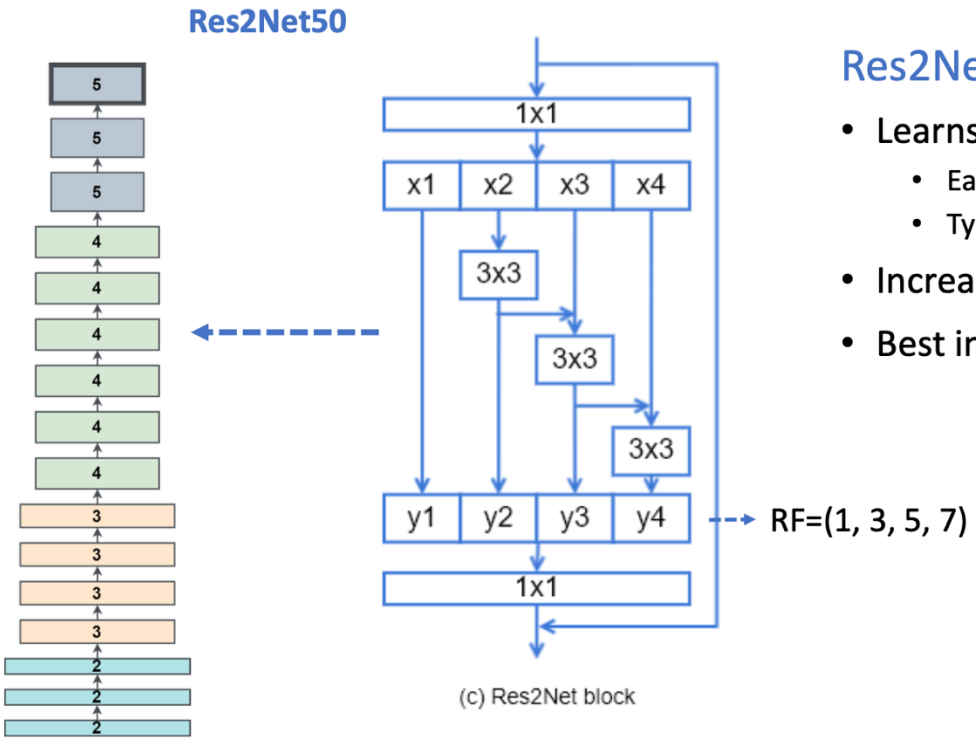


He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

Speaker recognition and verification

embedding

Res2Net 2D



Res2Net

- Learns multi-scale features:
  - Each channel group observes a different receptive field
  - Typically use scale=4 or 8
- Increases global receptive field in the full network
- Best in VoxCeleb

	Receptive Field (secs.)
ResNet34	2.4
ResNet50	1.2
Res2Net50 scale=4	3.6
Res2Net50 sale=8	8.3

Gao, Shang-Hua, et al. "Res2net: A new multi-scale backbone architecture." IEEE transactions on pattern analysis and machine intelligence 43.2 (2019): 652-662.

Speaker recognition and verification

embedding

Global Average Pooling (GAP)

- Mean of Encoder representations along time dimension
- For 1D Encoders: (B, C, T) -> (B, C)
- For2DEncoders:(B,C,F,T)->(B,CxF,T)->(B,CxF)

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x_t$$

Global Statistics Pooling (GSP)

- Concatenate:
  - Mean along time dimension
  - Standard Deviation along time dimension
- For1DEncoders:(B,C,T)->(B,2xC)
- For2DEncoders:(B,C,F,T)->(B,CxF,T)->(B,2xCxF)

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x_t$$

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T} x_t \odot x_t - \mu \odot \mu}$$

Snyder, David, et al. "Deep Neural Network Embeddings for Text-Independent Speaker Verification." Interspeech. 2017.

Speaker recognition and verification

embedding

Self attentive statistical pooling (SAP)

Statistics Pooling with different weight for each frame.

$$\mu = \frac{1}{T}\sum_{t=1}^{T} w_t x_t$$

Attentive Statistical Pooling (ASP)

w_t is the attention

Attentive Statistics Pooling with H heads

- Each heads look at different types of frames.
- Computes a different set of weights per head
- Computes weighted statistics per head
- Concatenate mean and std. dev of all heads.

$$\mu = \frac{1}{T}\sum_{t=1}^{T} w_t x_t$$

$$\sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T} w_t x_t \odot x_t - \mu \odot \mu}$$

Okabe, Koji, Takafumi Koshinaka, and Koichi Shinoda. "Attentive statistics pooling for deep speaker embedding." arXiv preprint arXiv:1803.10963 (2018).

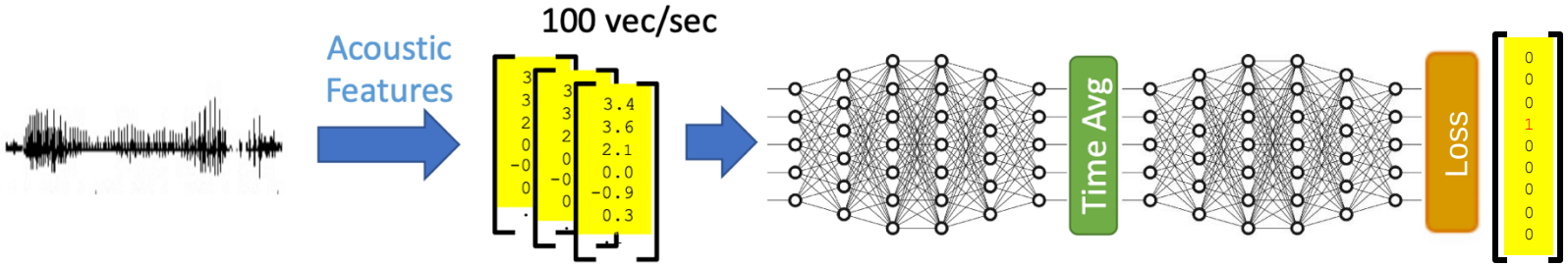
Speaker recognition and verification

Identification task

- Train the network to classify speakers

  Large dataset > 1K speakers

- The choice of the loss function is crucial



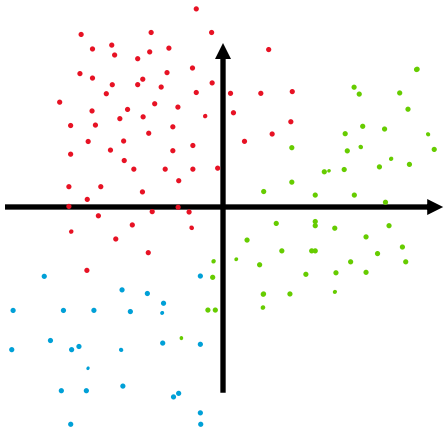
- The most common one for classification is the **Cross-Entropy** loss, but in speaker recognition the most used ones are *angular margin losses*.

Partially taken from Jesus Villalba

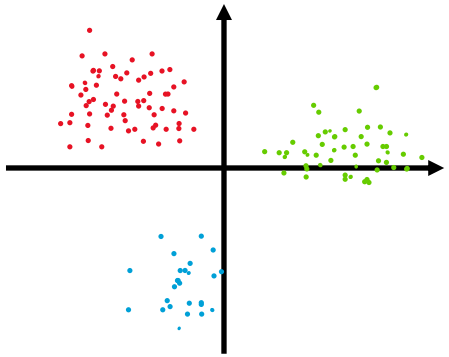
Speaker recognition and verification

Embedding space

- The loss function for speaker recognition should **map samples** into a **feature space** so that samples from the **same identity** are **more compact** and samples from **different identities** are **more separated**.



Cross Entropy



Other losses

Speaker recognition and verification

Loss functions

Multi-class Cross Entropy

It is the basic loss function for classification, so it is not assured to provide separable embeddings, that are discriminative for unseen speakers.

- Given N training samples
- Given the input features i and its label y_i
- f_j denotes the j-th element of the class score vector

$$L = \frac{1}{N}\sum_{i}^{N} -log(\frac{e^{f_{i,y_i}}}{\sum_{j} e^{f_{i,j}}})$$

$$p_i = \left[\frac{e^{f_{1,y_1}}}{\sum_{j=1}^{C} e^{f_{i,j}}}, \cdots, \frac{e^{f_{C,y_C}}}{\sum_{j=1}^{C} e^{f_{c,j}}}\right]$$

$$t_i = [0, \cdots, 1, \cdots, 0]$$

$$\Rightarrow \quad L = -\frac{1}{N}\sum_{i}^{N}\sum_{i}^{C} t_i log(p_i) = -\frac{1}{N}\sum_{i}^{N} log(\frac{e^{f_{i,y_i}}}{\sum_{j=1}^{C} e^{f_{i,j}}})$$

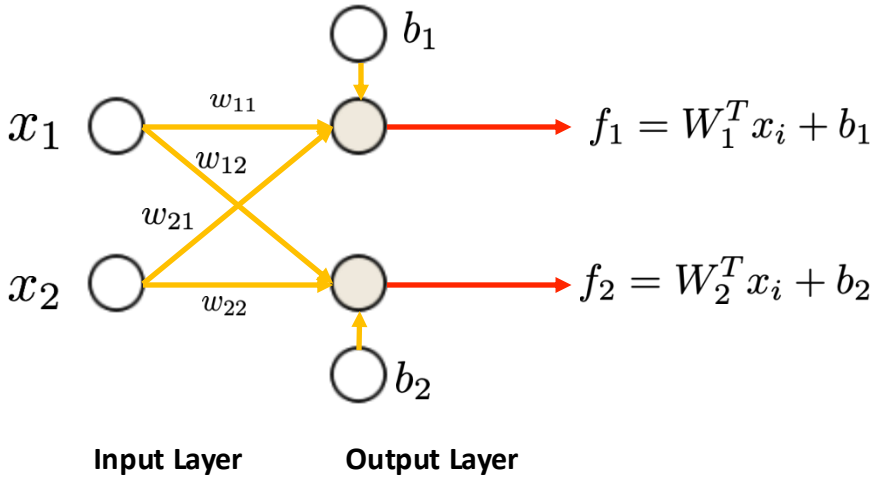
Wang, Hao, et al. "Cosface: Large margin cosine loss for deep face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

Angular Softmax (A-softmax)

A-softmax loss introduces a margin between the target class and the non-target class into the softmax loss. The margin is controlled by a hyper-parameter m.

$$L = \frac{1}{N}\sum_{i}^{N} -log(\frac{e^{f_{i,y_i}}}{\sum_j e^{f_{i,j}}}) = -\frac{1}{N}\sum_{i}^{N} log(p_i)$$

To illustrate A-softmax loss, we consider the **two-class** case and 1 layer Net. It is trivial to generalize the following analysis to multi-class cases.



Angular Softmax

A-softmax loss introduces a margin between the target class and the non-target class into the softmax loss. The margin is controlled by a hyper-parameter m.

$$L = \frac{1}{N}\sum_{i}^{N} -log(\frac{e^{f_{i,y_i}}}{\sum_{j} e^{f_{i,j}}}) = -\frac{1}{N}\sum_{i}^{N} log(p_i) = -\frac{1}{N}\sum_{i}^{N} \log\left(\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}}\right)$$

The posterior probabilities in the two-class case given by softmax loss are p_1 and p_2

$$p_1 = \frac{e^{W_1^T x_i + b_1}}{e^{W_1^T x_i + b_1} + e^{W_2^T x_i + b_2}}$$

$$p_2 = \frac{e^{W_2^T x_i + b_2}}{e^{W_1^T x_i + b_1} + e^{W_2^T x_i + b_2}}$$

The predicted label will be assigned to class 1 if $p_1 \geq p_2$ and class 2 if $p_1 < p_2$

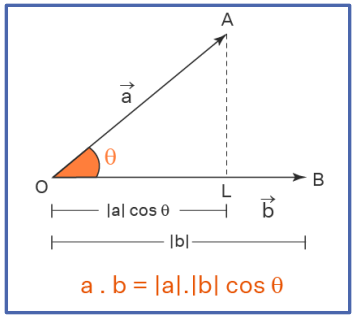
Angular Softmax

The predicted label will be assigned to class 1 if $p_1 \geq p_2$ and class 2 if $p_1 < p_2$

The decision boundary is:

$$p_1 - p_2 = 0 \quad \Longrightarrow^{b_1 = b_2 = 0} \quad (W_1^T - W_2^T)x_i = 0$$

$$(||W_1||cos(\theta_1) - ||W_2||cos(\theta_2))\,||x_i|| = 0$$



Here θ_1, θ_2 are the angles between x_i and W_1, W_2 respectively

There are **two** steps of modifications in defining A-softmax

Firstly:

$$||W_1|| = ||W_2|| = 1 \quad \text{and} \quad b_1 = b_2 = 0$$

The decision boundary then becomes angular boundary

$$cos(\theta_1) - cos(\theta_2) = 0$$

Angular Softmax

There are **two** steps of modifications in defining A-softmax

Secondly:

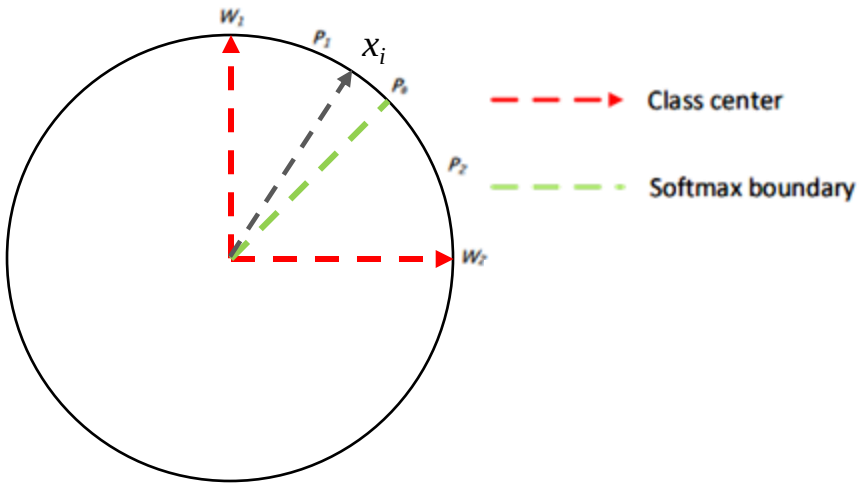
An integer m $(m \geq 2)$ is introduced to quantitatively control the size of angular margin.

The decision condition for class 1 becomes:

$$cos(m\theta_1) - cos(\theta_2) > 0$$

And for class 2:

$$cos(m\theta_2) - cos(\theta_1) > 0$$



This means when $cos(m\theta_1) > cos(\theta 2)$, we assign the sample to class 1; when $cos(m\theta_2) > cos(\theta_1)$, we assign the sample to class 2.

Angular Softmax

There are **two** steps of modifications in defining A-softmax

Secondly:

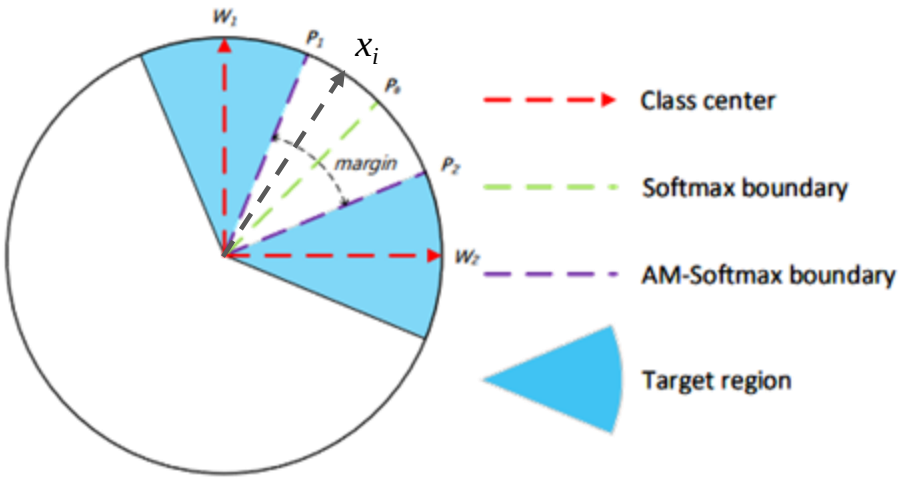
An integer m $(m \geq 2)$ is introduced to quantitatively control the size of angular margin.

The decision condition for class 1 becomes:

$$cos(m\theta_1) - cos(\theta_2) > 0$$

And for class 2:

$$cos(m\theta_2) - cos(\theta_1) > 0$$



This means when $cos(m\theta_1) > cos(\theta 2)$, we assign the sample to class 1; when $cos(m\theta_2) > cos(\theta_1)$, we assign the sample to class 2.

Speaker recognition and verification

Loss functions

Angular Softmax (SphereFace)

$$L = -\frac{1}{N}\sum_{i}^{N}\log\left(\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}}\right) = -\frac{1}{N}\sum_{i=1}^{N}\left(\frac{e^{||x_i||cos(m\theta_{y_i,i})}}{e^{||x_i||cos(m\theta_{y_i,i})} + \sum_{j\neq i} e^{||x_i||cos(\theta_{j,i})}}\right)$$

Since the cosine function is monotonically decreasing in $[0,\pi]$, to relax the constraint about θ ranging in $[0,\pi/m]$, we replace the cosine function with another one that allows to have a function monotonically decreasing for all values of the angles.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left(\frac{e^{||x_i||\phi(\theta_{y_i,i})}}{e^{||x_i||\phi(\theta_{y_i,i})} + \sum_{j\neq i} e^{||x_i||cos(\theta_{j,i})}}\right)$$

$$\phi(\theta_{y_i,i}) = (-1)^k cos(m\theta_{y_i,i}) - 2k \qquad k \in [0, m-1]$$

Li, Y., Gao, F., Ou, Z., & Sun, J. (2018, November). Angular softmax loss for end-to-end speaker verification. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)* (pp. 190-194). IEEE.

Speaker recognition and verification

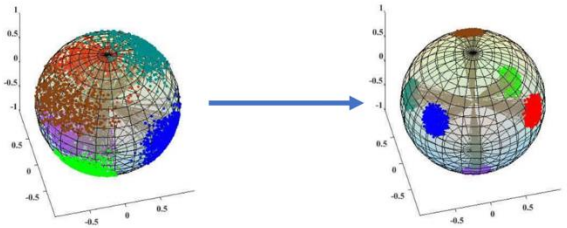
Loss functions

Additive Margin Softmax (AMSoftmax, CosFace)

$$L_i = -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j}}\right) = -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i}) + b_j}}\right)$$

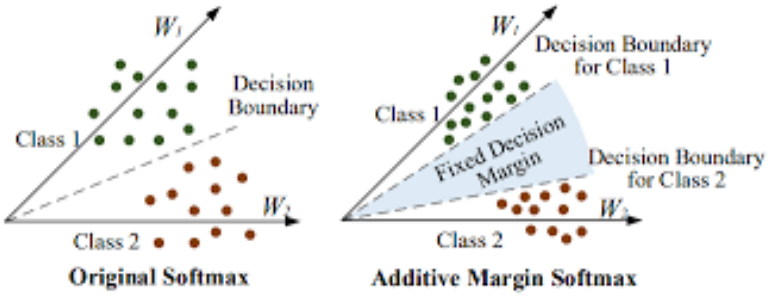
$$\xrightarrow[\|x_i\| = 1, \forall i]{\|\boldsymbol{W}_j\| = 1, \forall j}$$

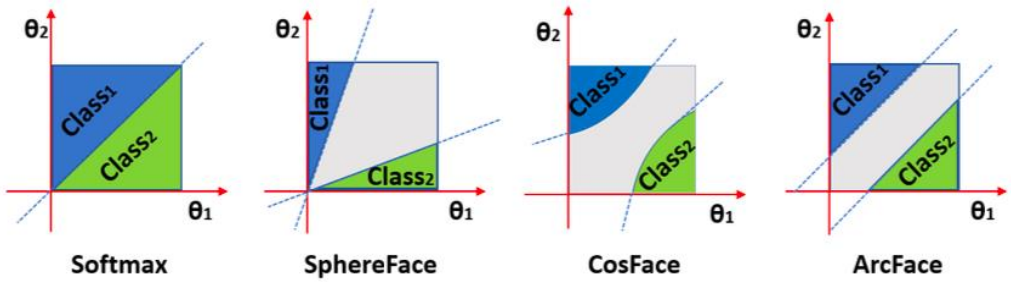
$$L_{lmc} = \frac{1}{N}\sum_i -\log\frac{e^{s(\cos(\theta_{y_i,i}) - m)}}{e^{s(\cos(\theta_{y_i,i}) - m)} + \sum_{j \neq y_i} e^{s\cos(\theta_{j,i})}}$$



Additive Angular Margin Softmax (ArcFace)

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^{n} e^{s\cos\theta_j}}$$

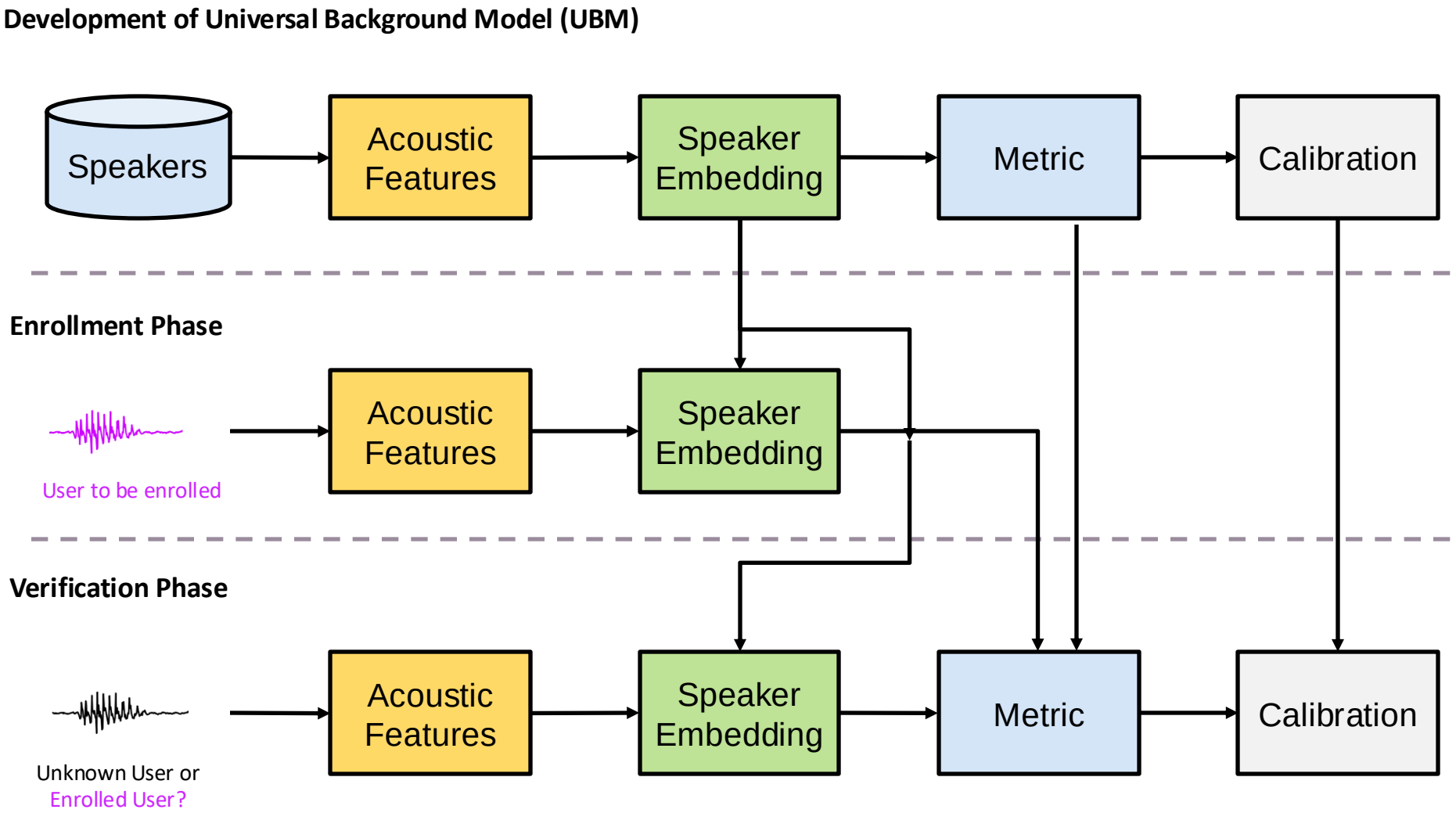




Wang, Hao, et al. "Cosface: Large margin cosine loss for deep face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018

Speaker verification

pipeline

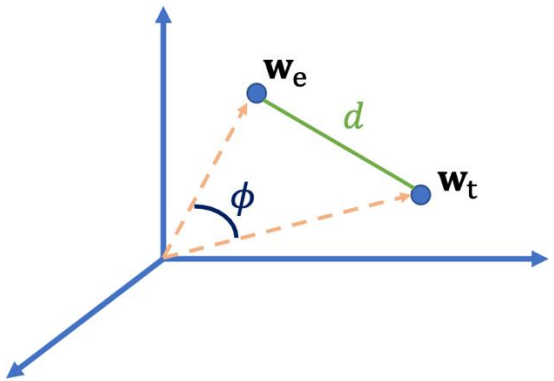


Partially taken from Sharat.S.Chikkerur

Metric

- Assume that:
  - $\mathbf{w}_e$ spk. embedding from enrollment utterance of speaker **X**
  - $\mathbf{w}_t$ spk. embedding from test utterance of person that claims to be speaker **X**
- The Metric compares enrollment and test embeddings $\mathbf{w}_e$, $\mathbf{w}_t$
  - **Cosine scoring** (simplest one):

$$s = \cos(\phi) = \frac{\mathbf{w}_e^T \mathbf{w}_t}{\|\mathbf{w}_e\|_2 \|\mathbf{w}_t\|_2}$$



  - **PLDA** (more complex)

Partially taken from Jesus Villalba

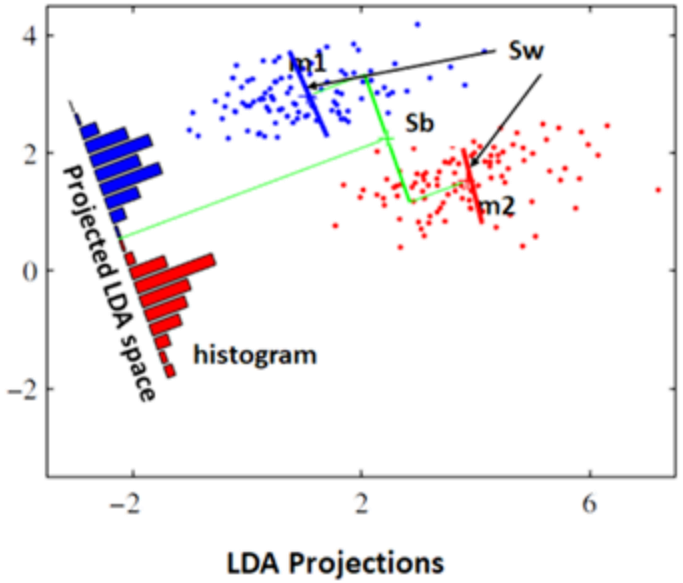
Linear Discriminant Analysis (LDA)

embedding

- LDA projects embeddings into a new vector space, so it finds directions along which:
  - Maximize the separation between classes.
  - Minimizes the covarianze within each class
- Between/Within-class covariances:

$$\mathbf{S}_b = \frac{1}{M}\sum_{i=1}^{M}(\mu_i - \mu)(\mu_i - \mu)^{\mathrm{T}}$$

$$\mathbf{S}_w = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i}\sum_{j=1}^{N_i}(\mathbf{x}_{ij} - \mu_i)(\mathbf{x}_{ij} - \mu_i)^{\mathrm{T}}$$



- Solve the generalized eigenvalue problem:

$$\max_{\mathbf{v}:\,\|\mathbf{v}\|=1}\frac{\mathbf{v}^{\mathrm{T}}\mathbf{S}_b\mathbf{v}}{\mathbf{v}^{\mathrm{T}}\mathbf{S}_w\mathbf{v}} \quad \rightarrow \quad \mathbf{S}_b\mathbf{v} = \lambda\mathbf{S}_w\mathbf{v}$$

Partially taken from Jesus Villalba

Probabilistic LDA

embedding

- Probabilistic Linear Discriminant Analysis (**PLDA**) is a probabilistic extension of LDA
- *It considers the feature vector as a **composition** of several factors:*

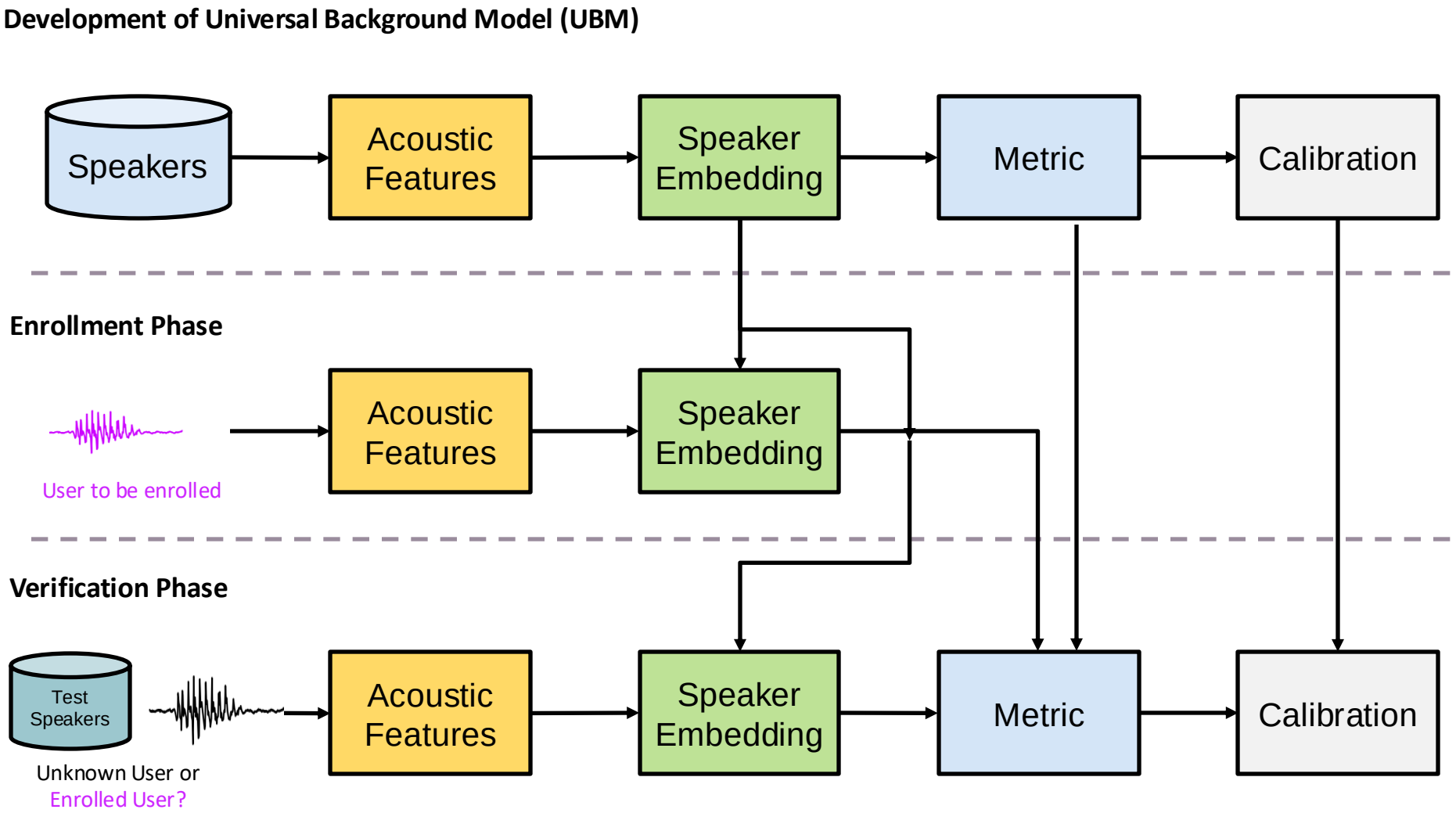
$$x_{ij} = \mu + Uh_i + Vw_{ij} + \epsilon_{ij}$$

- Uh_i *depending only on the identity of the speaker, models **between-speaker variation**.*
- $V w_{ij} + \varepsilon_{ij}$ *depending on the particular audio for that speaker, models **within-speaker variation**, and* μ *is the global dataset mean.*
- *U contains the basis for the between-speaker subspace while V the within-speaker one.*
- *PLDA assumes **Gaussian** priors for the latent variables*

Partially taken from Jesus Villalba

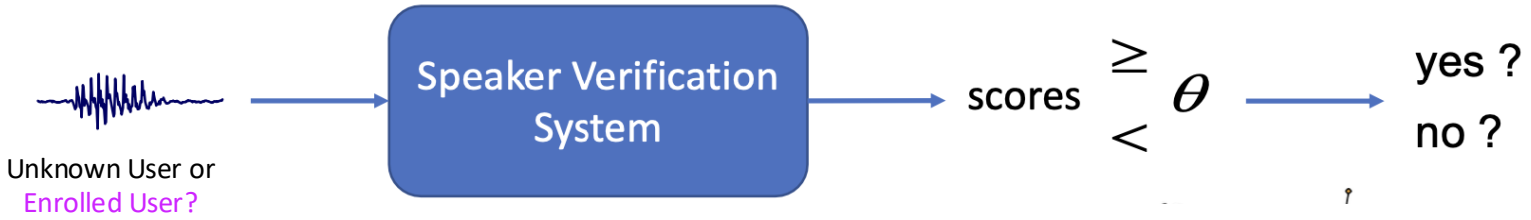
Speaker verification

pipeline



Partially taken from Jesus Villalba

Calibration



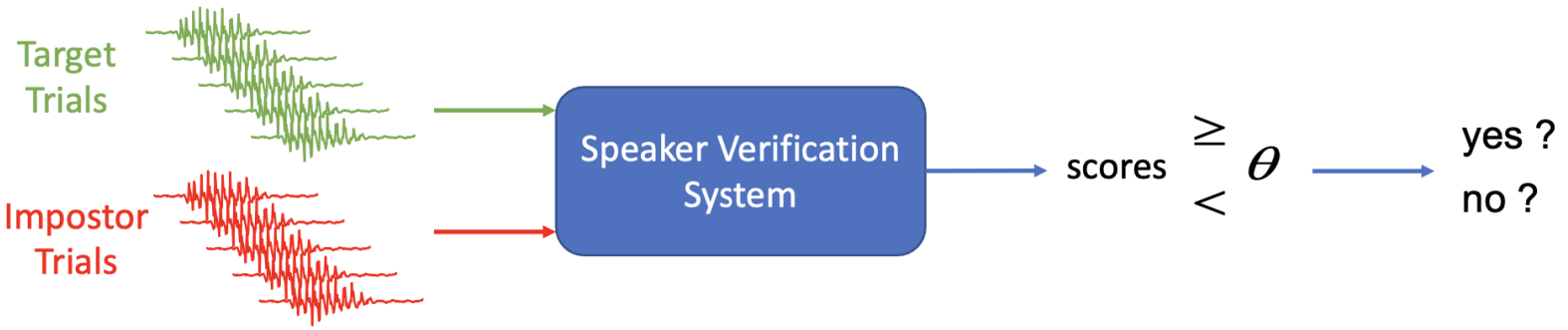
- How do we choose the decision threshold?
- High Security Application -> High decision threshold.
- Low Security Application: Low decision threshold

Partially taken from Jesus Villalba

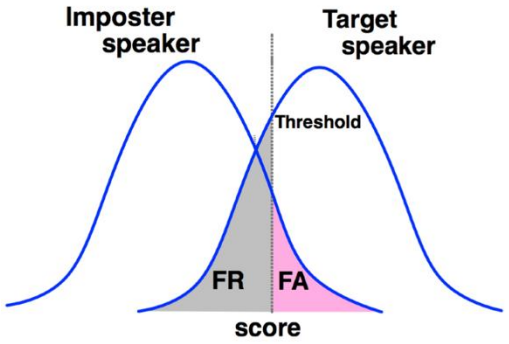
Speaker recognition and verification

How to take the decision

Way to evalutate



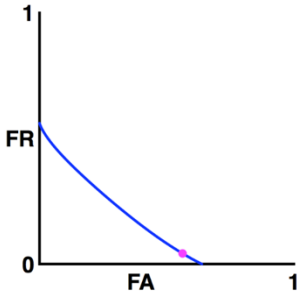
- Put >1k of target and impostor trials into the systems and count the errors
- Types of Errors:
  - Miss/False rejection (FR):
    - True speakers classified as impostor
    - Metric: Miss rate P_{MISS}
  - False alarm (FA):
    - Impostors classified as the true speaker
    - Metric: False alarm rate P_{FA}

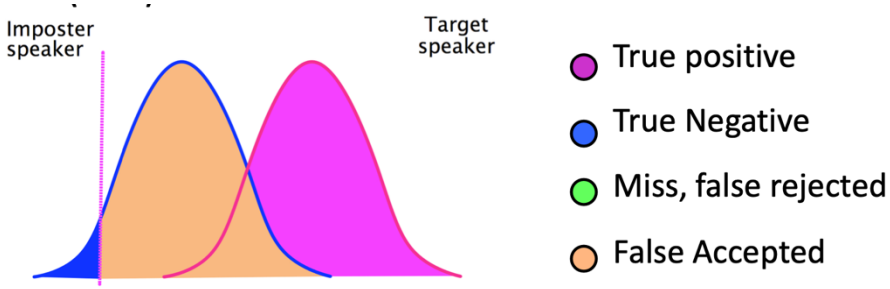


Partially taken from Jesus Villalba

Metrics

- Detection Error Trade-Off (DET)





- Equal Error Rate (EER)

$$P_{\text{Miss}}(\theta_{\text{EER}}) = P_{\text{FA}}(\theta_{\text{EER}})$$

$$\theta = \text{decision threshold}$$

- Detection Cost Function (DCF)

$$C_{\text{Det}}(\theta) = P_{\text{Miss}}(\theta) + \beta P_{\text{FA}}(\theta) \quad \text{with } \beta = \frac{1-P_{\text{target}}}{P_{\text{target}}}$$

$$\text{Minimum } C_{\text{Det}} = \min_{\theta} C_{\text{Det}}(\theta)$$

Partially taken from Jesus Villalba

Speaker diarization

Speaker diarization

definition

Speaker recognition is the "who spoken when" task: given a recording, divide it into segments, where each segment corresponds to speech of a single speaker

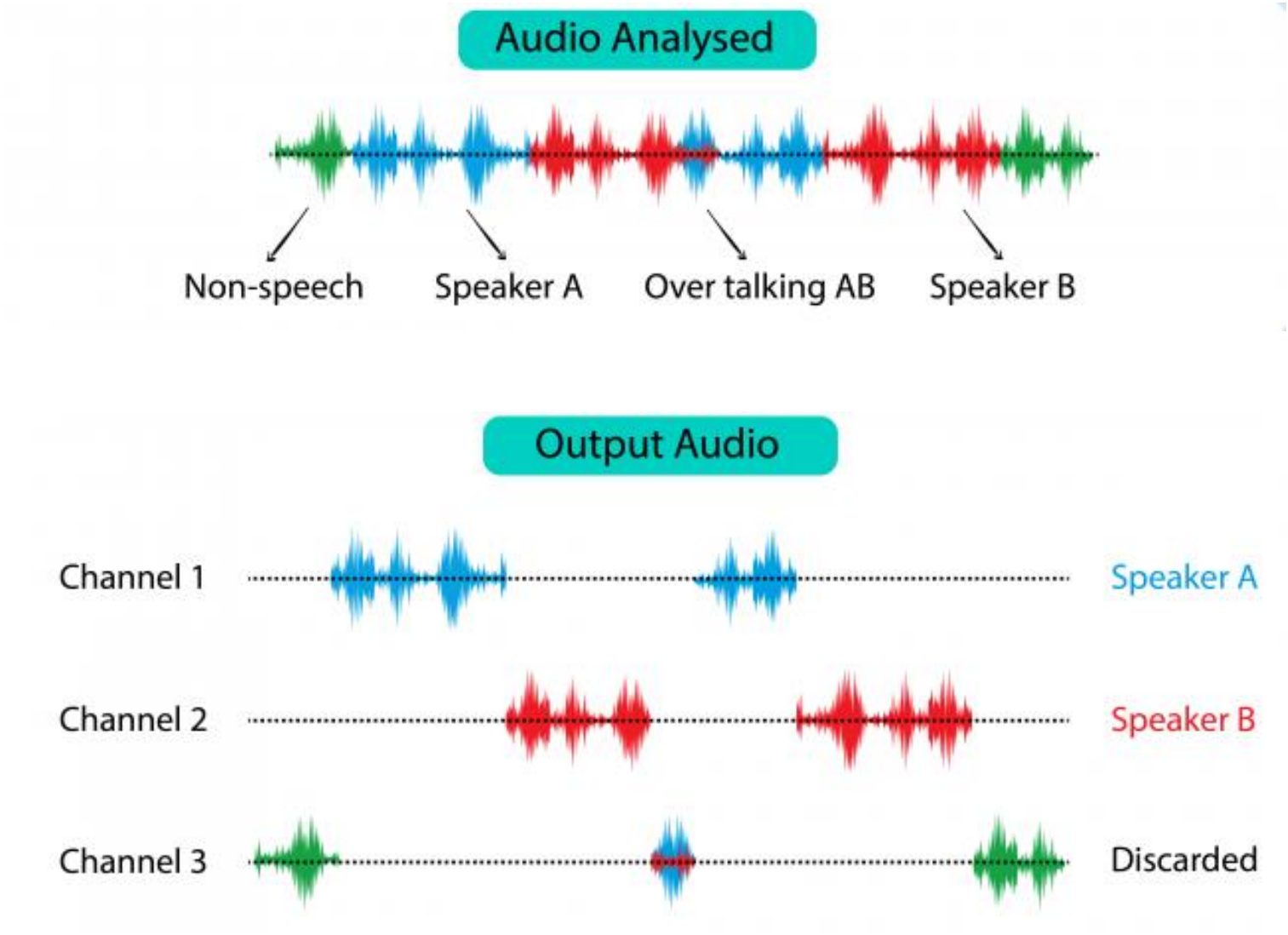
Each recording contains **multiple speakers** – unlike what we have assumed so far for speech recognition and speaker verification

Multiple speakers in a recording is realistic – many possible domains, e.g.:

- Broadcast media
- Telephone conversations
- Call centers
- Meeting recordings

Speaker diarization

definition

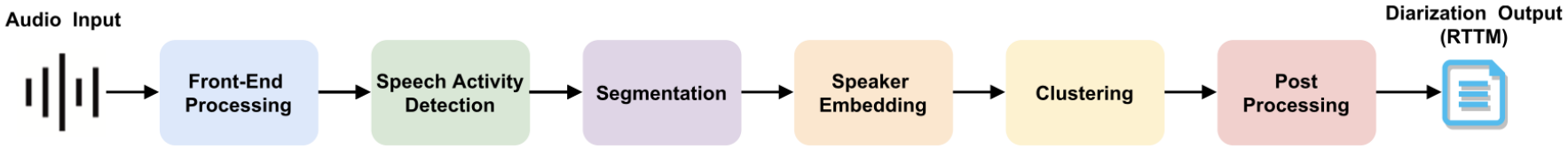


Speaker diarization

pipeline

A basic approach to diarization:

Segment the recording into a sequence of short pieces, each assumed to be a single speaker. Then treat as a speaker verification task between all pairs of segmented utterances

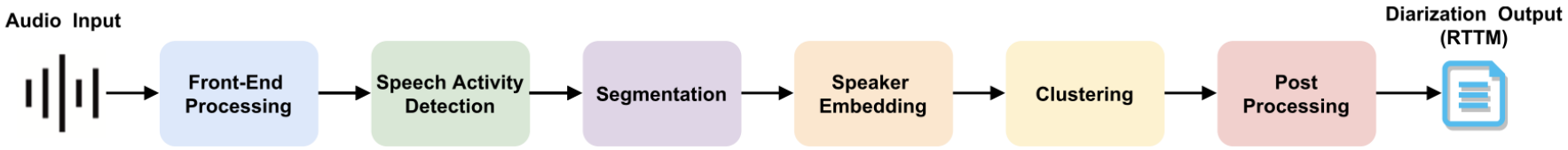


Each of these sub-modules is optimized individually in general.

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

Front-end

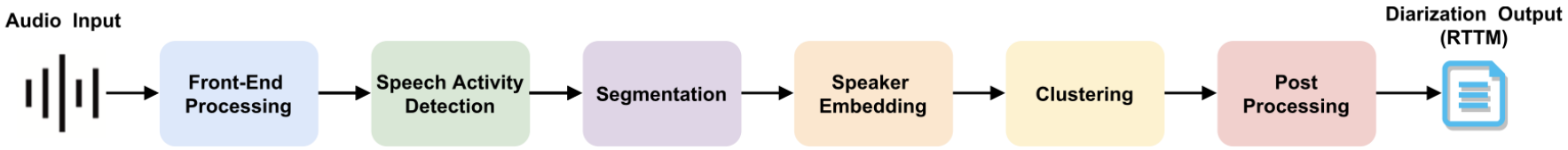


Front-end processing is used for speech enhancement, dereverberation, denoising, speech separation, and speech extraction

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

SAD



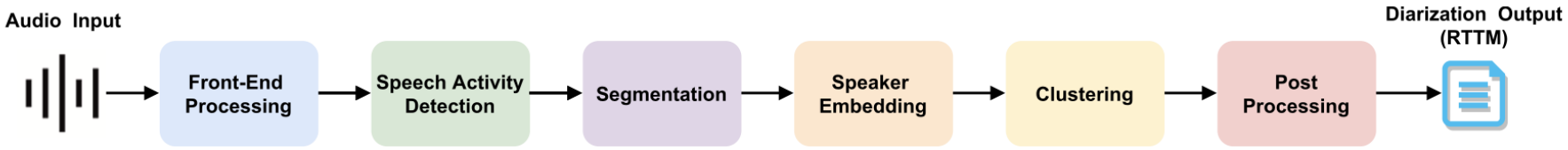
SAD, also known as voice activity detection (VAD), distinguishes speech from non-speech such as background noise.

Since SAD is a pre- processing step that can create errors that propagate through the whole pipeline.

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

Segmentation



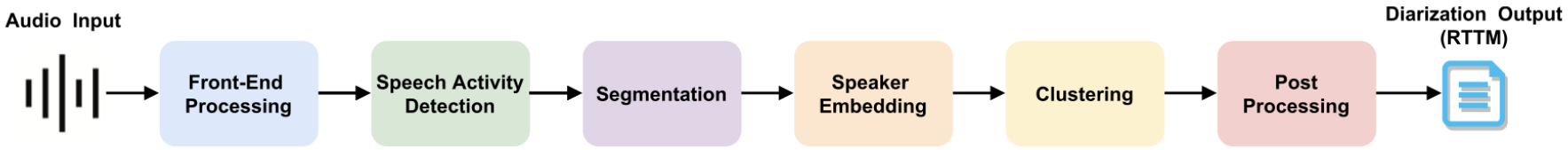
Segmentation is the activity of partitioning the audio tracks into sub-segments.

- *Uniform segmentation*: which generate audio sub-segments of the same length
- *Speaker-change point detection*: based on hypothesis testing, which search for sub-segments of the maximum length with the restriction of containing one speaker only.

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

Embedding



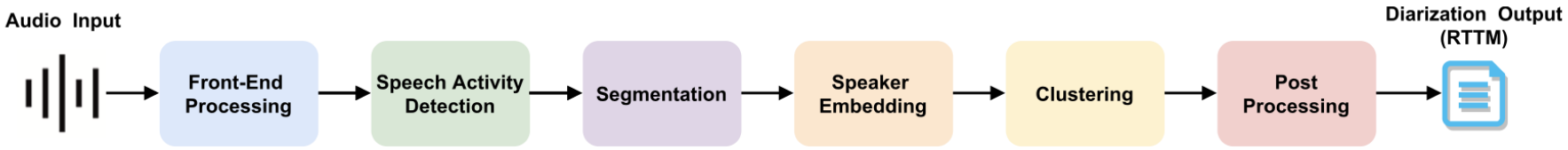
The **Speaker Embedding** phase involves transforming speech segments into multidimensional vectors, aiming to generate embeddings that exhibit similarity for segments linked to the same speakers.

Usually: x-vectors, neural-based embeddings

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

Clustering



Clustering are employed to group speech segments into clusters, where each cluster represents a different speaker. There are several clustering techniques used in speaker diarization:

Agglomerative Hierarchical, K-Means, DBSCAN, Gaussian Mixture Models (GMM), Affinity Propagation, Spectral Clustering, Agglomerative Clustering with Cosine Similarity, etc.

* See **additional materials** on Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language, 72, 101317.

Speaker diarization

Agglomerative vs Divisive Clustering

Clustering Hierarchical methods again come in two varieties, **agglomerative** and **divisive**.

Agglomerative methods:

- Start with partition P_n where each object forms its own cluster.
- Merge the two closest clusters, obtaining P_{n-1}.
- Repeat merge until only one cluster is left.

Agglomerative methods require a **rule** to decide which clusters to **merge**.
Typically, one defines a distance between clusters and then **merges** the two clusters that **are closest**.

Speaker diarization

Agglomerative vs Divisive Clustering

Clustering Hierarchical methods again come in two varieties, **agglomerative** and **divisive**.

Divisive methods

- Start with P_1.
- Split the collection into two clusters that are as homogenous (and as different from each other) as possible.
- Apply splitting procedure recursively to the clusters.

Divisive methods require a **rule** for **splitting** a **cluster**.

Speaker diarization

Hierarchical Agglomerative Clustering

Need to define a **distance** $d(P,Q)$ between groups, given a distance measure $d(x,y)$ between observations.

Commonly used **distance measures**:

1. $d_1(P,Q) = \min d(x,y)$, for x in P, y in Q (single linkage)
2. $d_2(P,Q) = \text{ave } d(x,y)$, for x in P, y in Q (average linkage)
3. $d_3(P,Q) = \max d(x,y)$, for x in P, y in Q (complete linkage)
4. $d_4(P,Q) = \left\| \bar{x}_P - \bar{x}_Q \right\|$ (centroid method)
5. $d_5(P,Q) = 2\frac{|P||Q|}{|P|+|Q|}\left\| \bar{x}_P - \bar{x}_Q \right\|^2$ (Ward's method)

d_5 is called Ward's distance.

Speaker diarization

Hierarchical Agglomerative Clustering

- Let $\boldsymbol{P}_k = P_1, \ldots, P_k$ be a partition of the observations into k groups.
- Measure goodness of a partition by the sum of squared distances of observations from their cluster means:

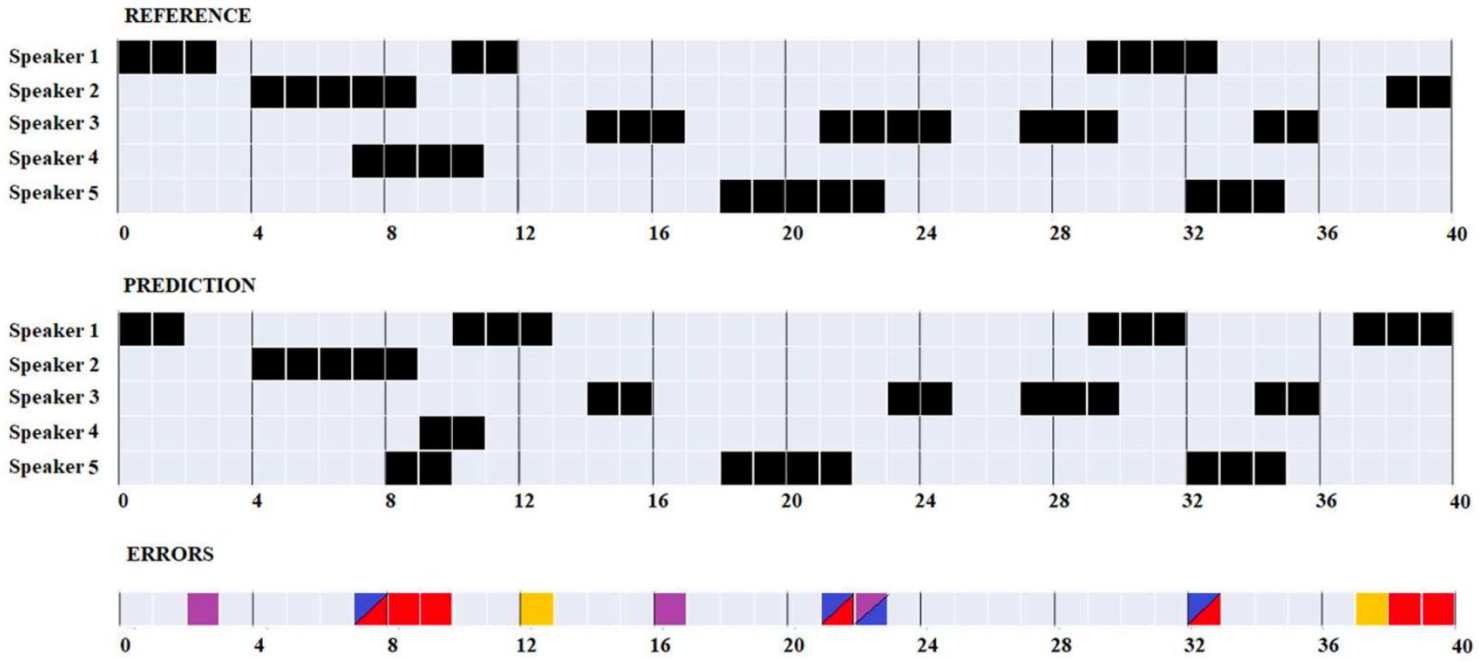
$$RSS(\mathbf{P}_k) = \sum_{i=1}^{k} \sum_{j \in P_i} \left\| x_j - \bar{x}_{P_i} \right\|^2$$

- Consider all possible $(k\text{-}1)$-partitions obtainable from $\boldsymbol{P}_k$ by a merge
- Merging two clusters with smallest Ward's distance optimizes goodness of new partition.

Speaker diarization

performance

Diarization Error Rate (DER): It is measured as the fraction of time that is not attributed correctly to a speaker or non-speech



$$\text{DER} = \frac{\text{FA} + \text{Missed} + \text{Speaker-Confusion}}{\textit{Total Duration of Time}}$$

Speaker Emotion Recognition

Personalized experience

Smart home

Smart home devices may be aware of our **emotions** and consequently they may adapt the behavior based on our mood

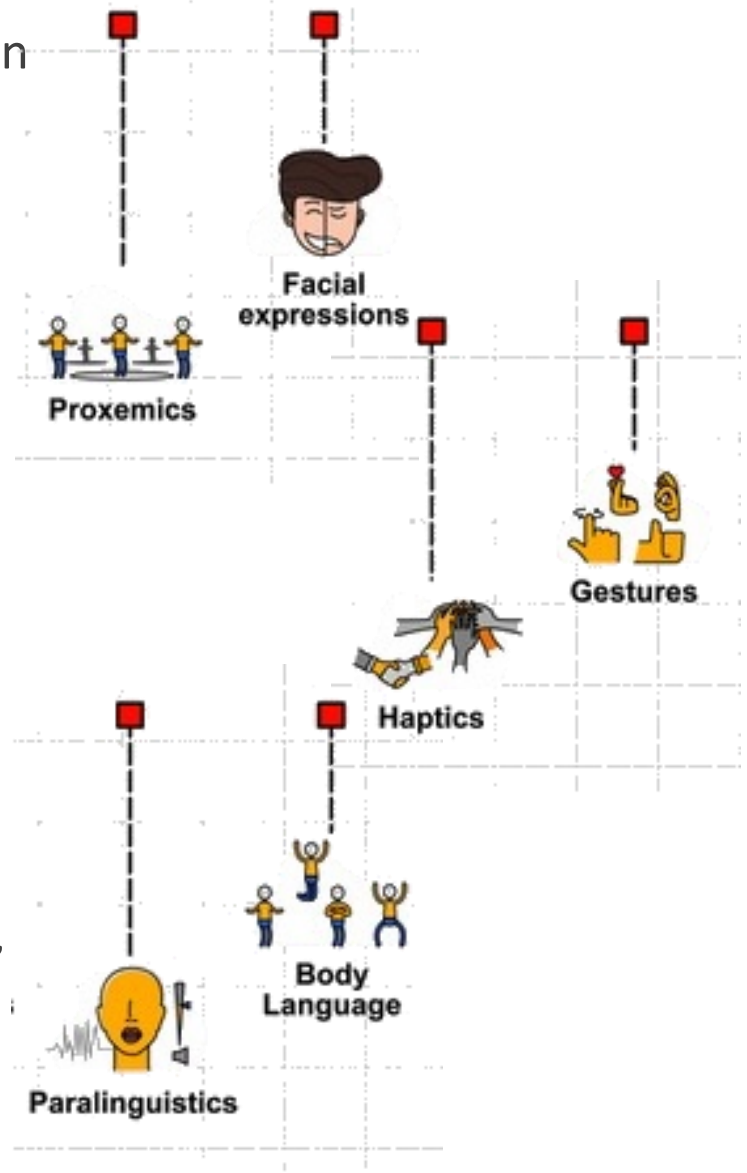
Introduction

Nonverbal behavior

- **Nonverbal behavior** expresses and reveals human emotions and represents the 93% of our interactions with others;

- Nonverbal behavior can be divided into several categories, such as: **proxemics**, **haptics**, **body language**, **gestures**, **facial expressions**, **paralanguage**,

- We focus on **Speech Emotion Recognition** (SER) which can be largely employed in several application domains such as: *human computer interface*, *robotics*, *audio surveillance, e-learning*, *computer games*, *decisional system for job interview*, etc.

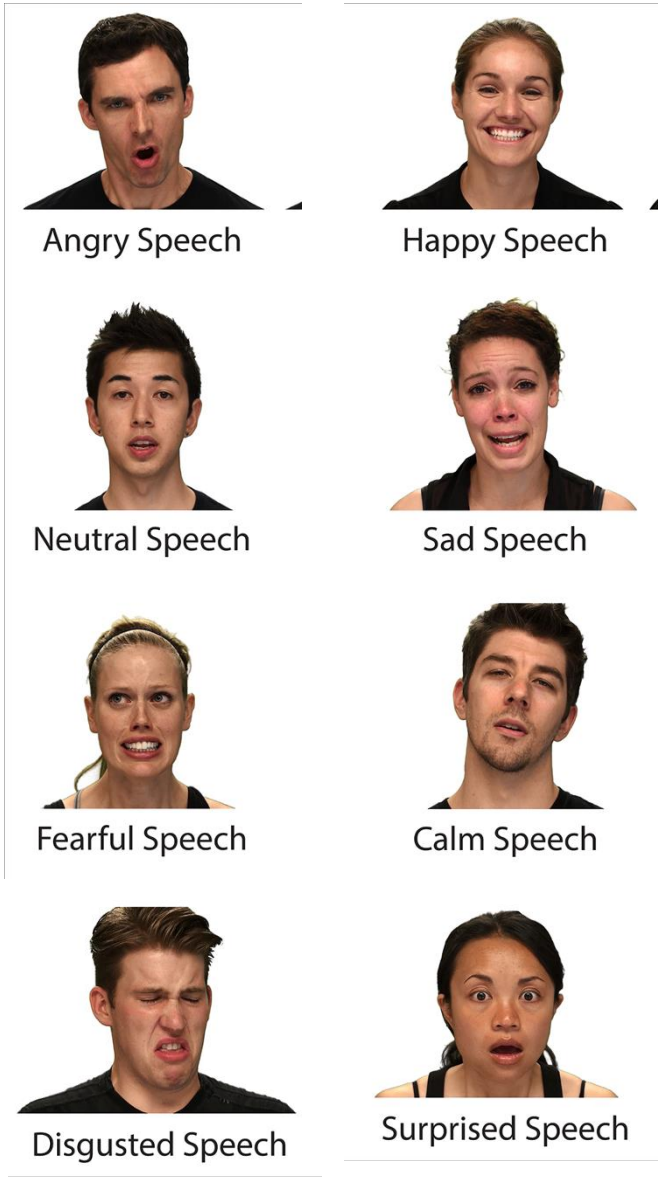


Introduction

Types of Emotions

- Vocal cues

Emotions	Vocalic cues
Neutral	Speech is articulated clearly with adequate pauses between words.
Happy	Raised precision of articulation, more breathy sounding voice. Pitch mean, pitch range and pitch variance increase.
Sad	Overall reduction in articulation, lower than normal average pitch. Pitch range is narrow with a slow tempo.
Anger	Highest energy level and pitch, faster rate of speech.
Disgust	Increased precision of articulating and stressing certain words, descending pitch inflection at the end of words. Slow Rate of speech slow with many pauses, longer phonation time.
Surprised	Pitch median and tempo are high with quite a wide pitch range.



Introduction

Emotions in the voice

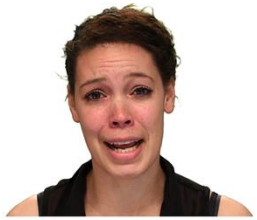
- Human speech is influenced by the **physiology of the speaker**, such as: shape of vocal tract, tone of the voice and the phonetic content such as pitch, intensity, energy and duration.

- Since these characteristics are influenced by emotions, the selection of the right **speech descriptors** is fundamental to achieve an automatic discrimination and recognition of distinct emotions such as: **neutral**, **calm**, **happy**, **sad**, **angry**, **fearful**, **disgust** and **surprised**.

Angry Speech

Happy Speech

Neutral Speech



Sad Speech

Fearful Speech

Calm Speech

Disgusted Speech

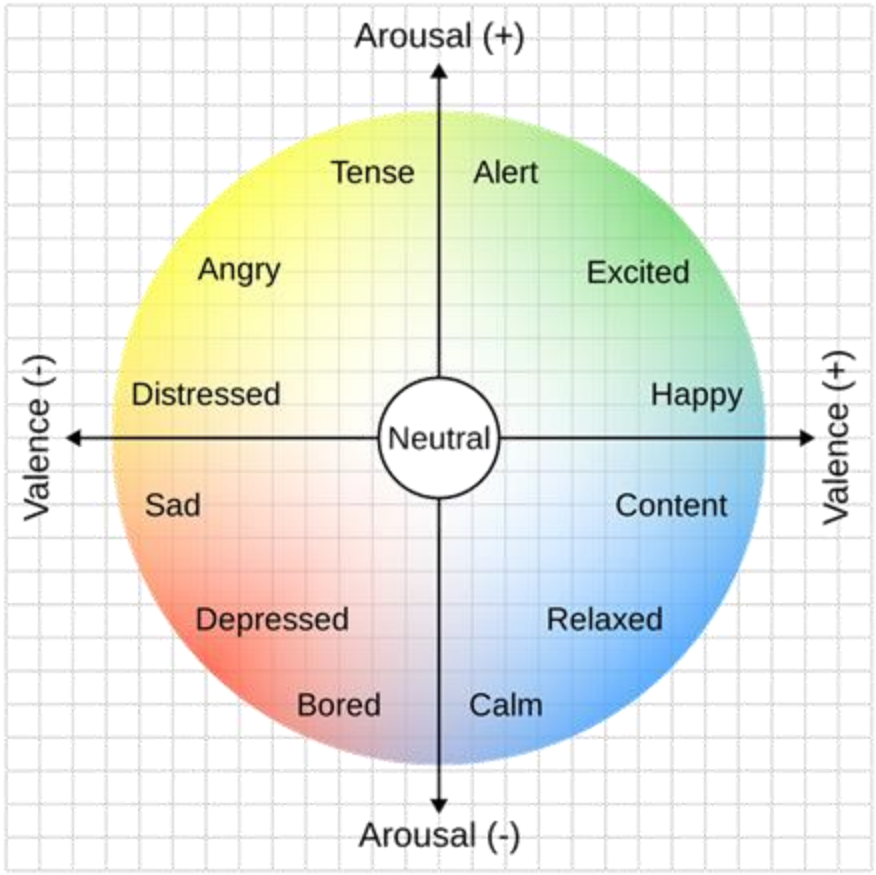
Surprised Speech

Introduction

Two-dimensional feeling

- Most scholars agree with Russell and Lisa Feldman Barrett's theory. This theory states that every sense can be **decomposed** into **primary emotions**, as each color can be produced with original colors. The primary emotions include happiness, sadness, fear, wonder, hatred, and anger. These feelings are the most prominent of distinctive emotions

- Emotions have **two dimensions**: **arousal** and the amount of vitality (**valence**)



* See **additional materials** on Russell, J. A., & Barrett, L. F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. Journal of personality and social psychology, 76(5), 805.

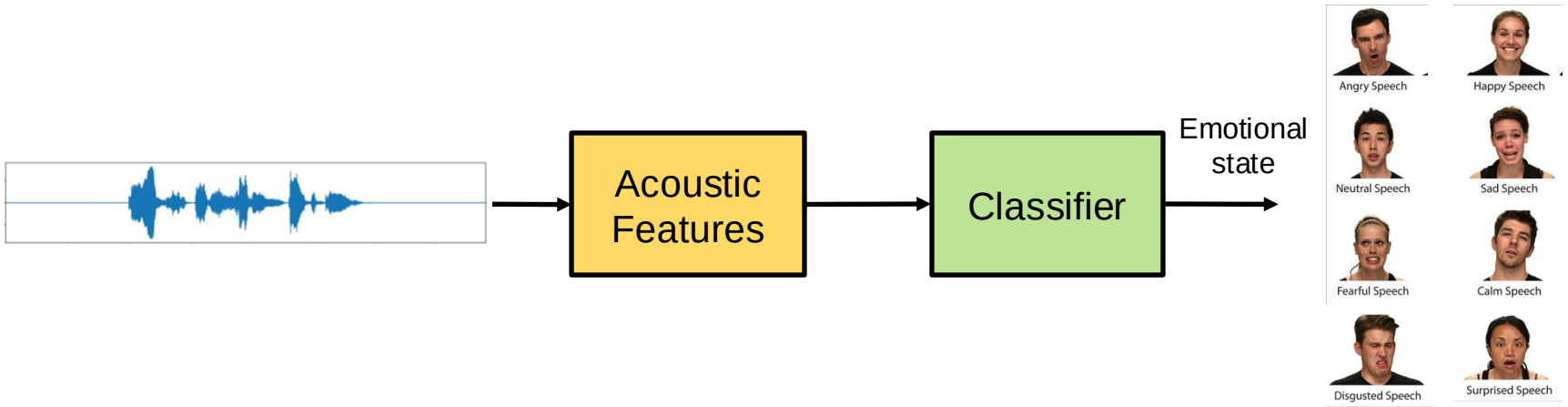
Speaker emotion recognition

pipeline

1. The audio sample is initially **pre-processed** by removing silence and by separating it into windows of 20ms with a step of 10ms.

2. For each segment of the acustic **features are extracted** and then sent to the classifier.

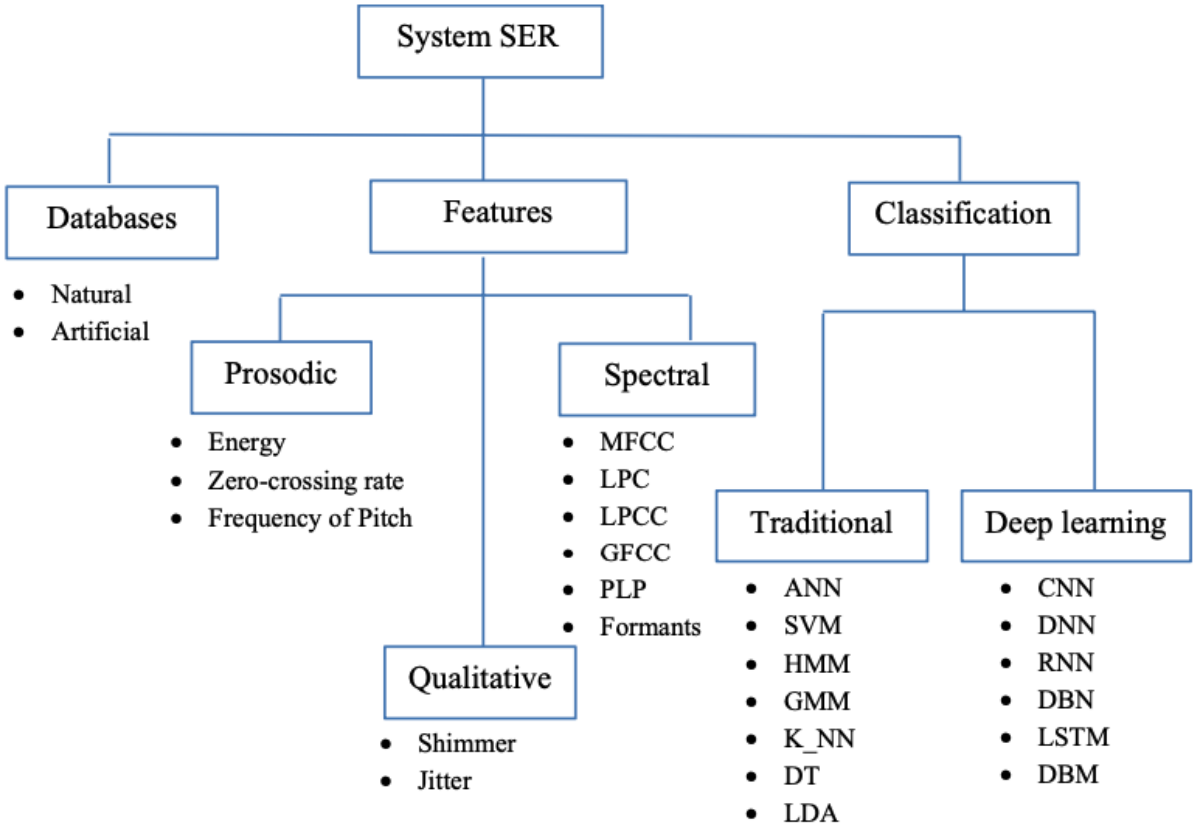
3. The feature vector is the input of the **classifier**. The output of the classifier is one of the possible emotion states, i.e. neutral, calm, happy, sad, angry, fearful, disgust and surprised.



Speaker emotion recognition (SER)

literature

An overview of speech emotion recognition systems



* See **additional materials** on Al-Dujaili, M. J., & Ebrahimi-Moghadam, A. (2023). Speech emotion recognition: a comprehensive survey. Wireless Personal Communications, 129(4), 2525-2561.

Speaker emotion recognition

Available dataset

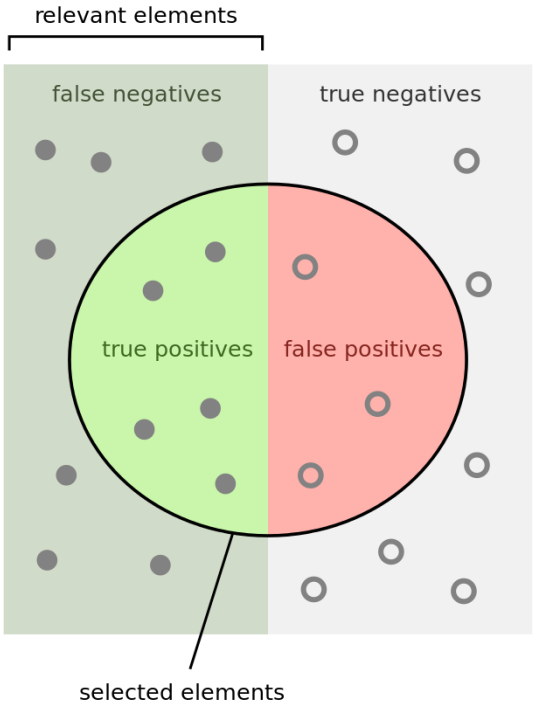
Information on databases in the field of recognizing the sense of speech

Database	Language	Type	Size and language	Emotions
Berlin emotional Database (EMO-DB)	German	Artificial	7 Emotions × 10 speakers(5 male, 5 female) × 10 utterances	Anger, boredom, disgust, fear, happiness, sadness, neutral
LDC Emotional Speech Database	English	Artificial	7 speakers (4 male, 3 female), 470 utterances	Hot anger, cold anger, disgust, fear, contempt, happiness, sadness, neutral, panic, pride, despair, elation, interest, shame, boredom
IITKGP-SEHSC	Hindi	Artificial	12,000 utterances by 10 professionals	Happy, anger, fear, disgusted, surprised, sad, sarcastic, neutral
Multilingual emotional speech database of north east India (MESDNEI)	Assamese	Artificial	4200 utterances of 5 native languages of Assam by 30 speakers	Sadness, anger, fear, disgust, happiness, surprise, neutral
CHEAVD	Chinese	Artificial	140 min emotional segments extracted from talk shows, T.V. plays, and films by 238 speakers	Angry, fear, happy, neutral, sad, surprise
BAUM-1 Speech Database	Turkish	Artificial and Natural	31 speakers (18 male, 13 female), 288 acted, 1222 spontaneous video clip	Happiness, anger, sadness, disgust, fear, surprise, bothered, boredom, contempt, unsure, being thoughtful, concentration, interest
Speech Under Simulated and Actual Stress Database (SUSAS)	English	Artificial and Natural	32 speakers (19 male, 13 female), 16,000 utterances also include the speech of Apache Helicopter pilots	Four states of speech under stress: Neutral, Angry, Loud, and Lombard
Surrey Audio-Visual Expressed Emotion (SAVEE)	English	Artificial	14 speakers (male) × 120 utterances	Anger, disgust, fear, happiness, sadness, surprise, neutral, common
Chinese Emotional Speech Corpus (CASIA)	Mandarin	Artificial	6 Emotions × 4 Speakers (2 male, 2 female) × 500 utterances (300 parallel, 200 non-parallel texts)	Surprise, happiness, sadness, anger, fear, neutral
Toronto Emotional Speech Database (TESS)	English	Artificial	2 speakers (female), 2800 utterances	Anger, disgust, neutral, fear, happiness, sadness, pleasant, surprise
Danish Emotional Speech Database (DES)	Danish	Artificial	4 speakers (2 male, 2 female)10 min of speech	Neutral, surprise, anger, happiness, sadness
Keio University Japanese Emotional Speech Database (Keio-ESD)	Japanese	Artificial	71 speaker (male) 940 utterances	Anger, happiness, disgusting, downgrading, funny, worried, gentle, relief, indignation, shameful, etc. (47emotions)
RECOLA Speech Database	French	Natural	46 speakers (19 males, 27 females) 7 h of speech	Five social behaviors (agreement, dominance, engagement, performance, rapport); arousal and valence
Italian Emotional Speech Database (EMOVO)	Italian	Artificial	6 speakers (3 male, 3 female) × 14 sentences × 7 emotions = 588 utterances	Disgust, happiness, fear, anger, surprise, sadness, neutral
Spanish emotional database	Spanish	Artificial	Contains more data (4528 utterances in total)	Anger, sadness, joy, fear, disgust, surprise, Neutral/normal

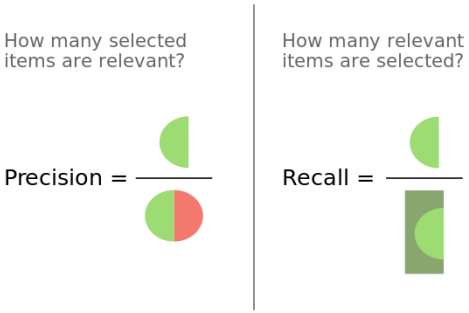
Metrics

Classification: confusion matrix (two classes)

metrics



		predicted: negative	predicted: positive
actual examples: negative		*a* TN - True Negative correct rejections	*b* FP - False Positive false alarms type I error
actual examples: positive		*c* FN - False Negative misses, type II error overlooked danger	*d* TP - True Positive hits



- **Precision**, predicted positive value = d/(b + d) = TP/predicted positive
- **Recall**, predicted positive value = d/(c + d) = TP/actual positive

Classification: confusion matrix (two classes)

metrics

Performance measures calculated from the confusion matrix entries:

		predicted	
		negative	positive
actual examples	negative	*a* TN - True Negative correct rejections	*b* FP - False Positive false alarms type I error
	positive	*c* FN - False Negative misses, type II error overlooked danger	*d* TP - True Positive hits

- **Accuracy** = (a+d)/(a+b+c+d) = (TN + TP)/total
- **Sensitivity, true positive rate**, **recall** = d/(c + d) = TP/actual positive
- **Specificity**, **true negative rate** = a/(a + b) = TN/actual negative
- **False positive rate**, false alarm = b/(a + b) = FP/actual negative = 1 - specificity
- **False negative rate** = c/(c + d) = FN/actual positive = 1-sensitivity

Classification: F - measure

Available dataset

A measure that combines precision and recall is the **harmonic mean of precision and recall**, the traditional **F-measure** or balanced F-score

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\beta^2 \cdot \mathrm{precision} + \mathrm{recall}}$$

A special case of this is the F1 measure with **beta = 1**:

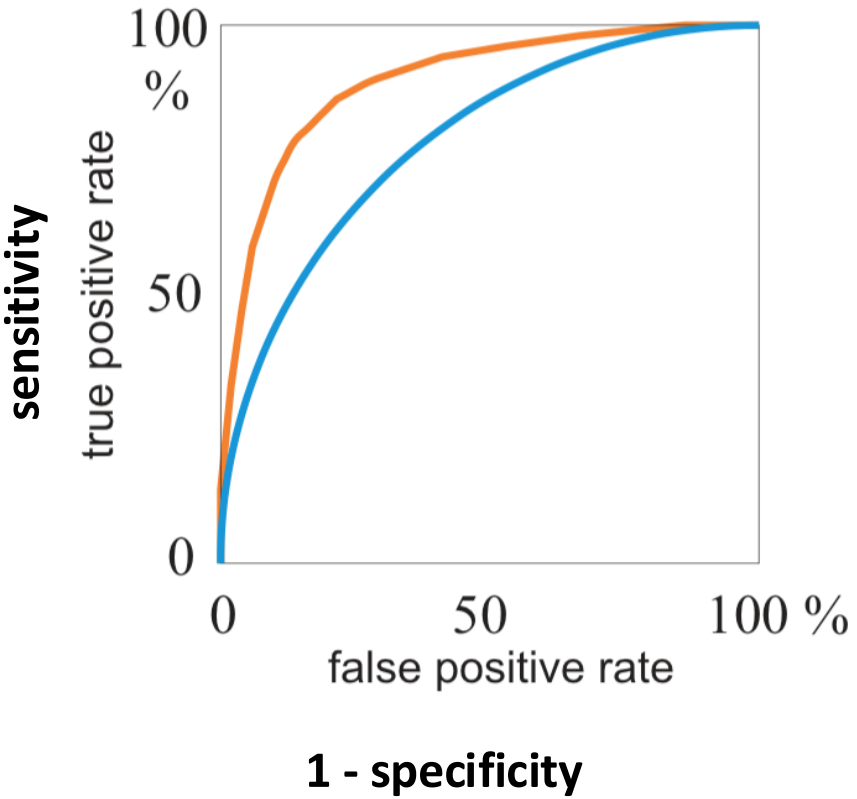
$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

It reaches its best value at 1 (perfect precision and recall) and worst at 0. Note, however, that the F-measures do not take the true negatives into account

Receiver Operating Characteristic

ROC curve

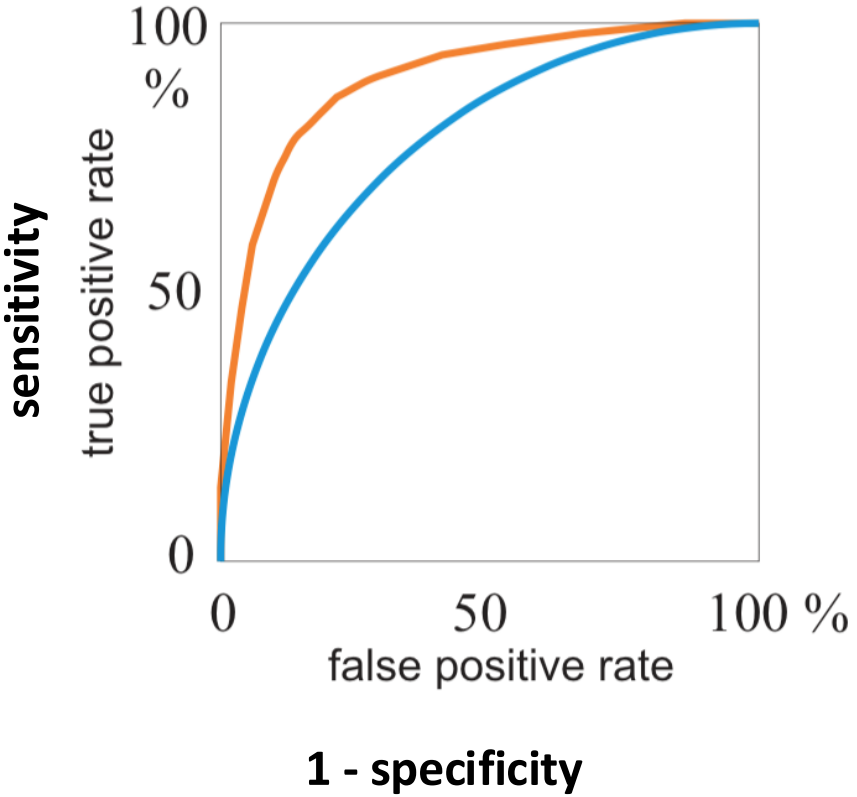
- Two numbers – **true positive rate** and **false positive rate** – are much more informative than the single number.
- These two numbers are better visualized by a curve, e.g., by a Receiver Operating Characteristic (ROC), which informs about:
  - Performance for all possible misclassification costs.
  - Performance for all possible class ratios.
  - Under what conditions the classifier c1 outperforms the classifier c2?



Receiver Operating Characteristic

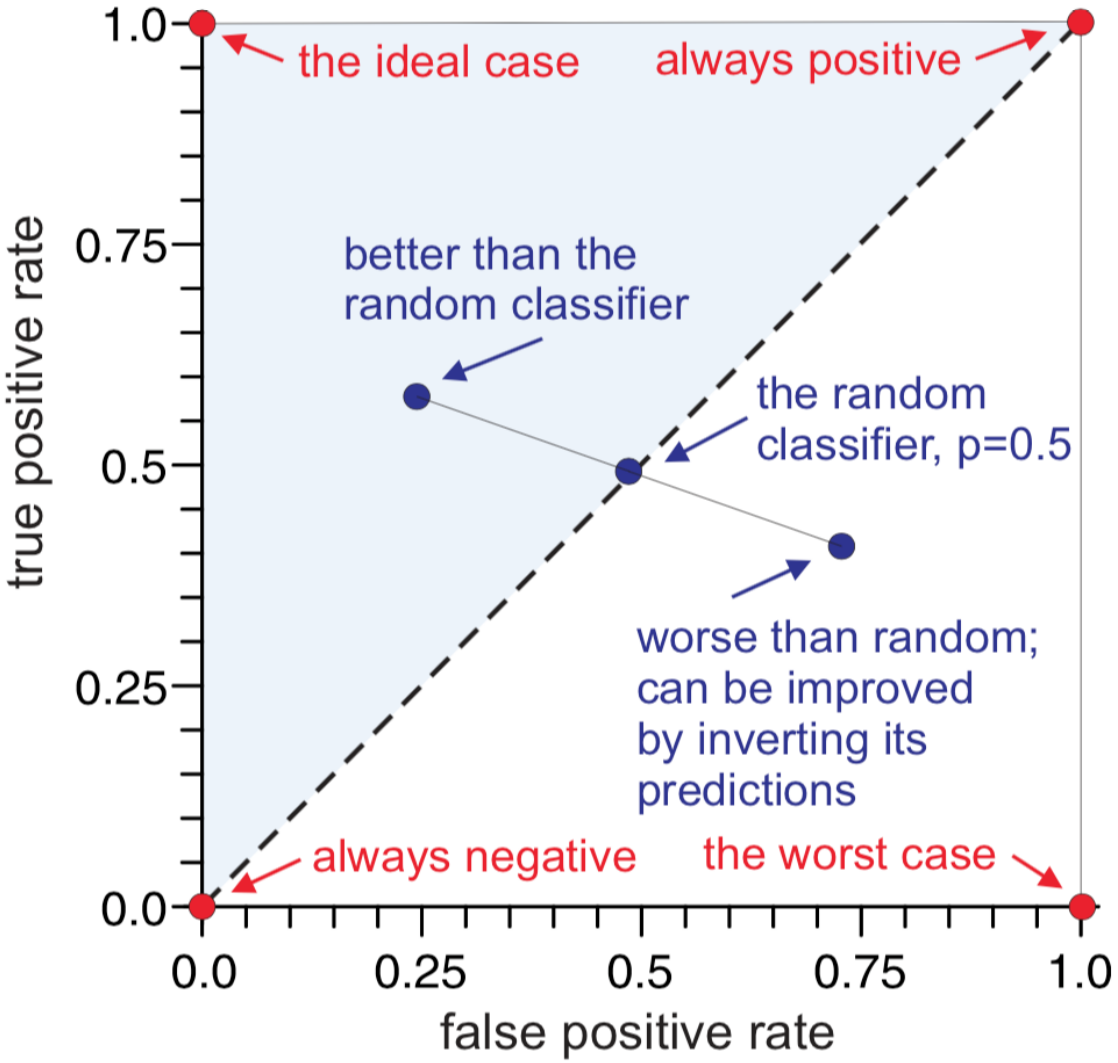
ROC curve

- Originates in the field of processing of radar signals.
- Useful for the evaluation of dichotomic classifiers performance.
- Characterizes degree of overlap of classes for a single feature.
- Decision is based on a single threshold Θ (called also operating point).
- Generally, false alarms go up with attempts to detect higher percentages of true objects.
- A graphical plot showing (hit rate, false alarm rate) pairs.
- Different ROC curves correspond to different classifiers. The single curve is the result of changing threshold Θ.



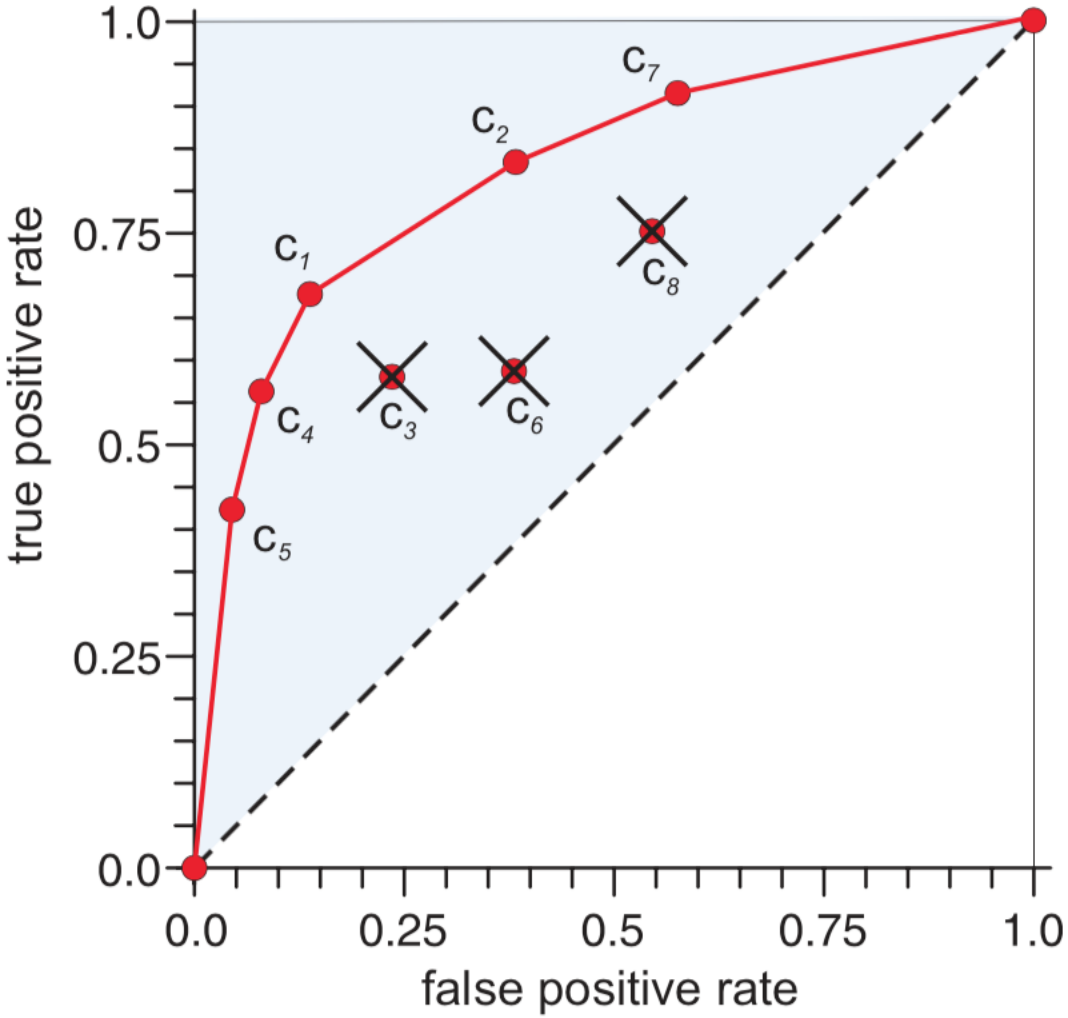
Receiver Operating Characteristic

ROC curve



Receiver Operating Characteristic

ROC curve



Summing up

QUESTIONS?