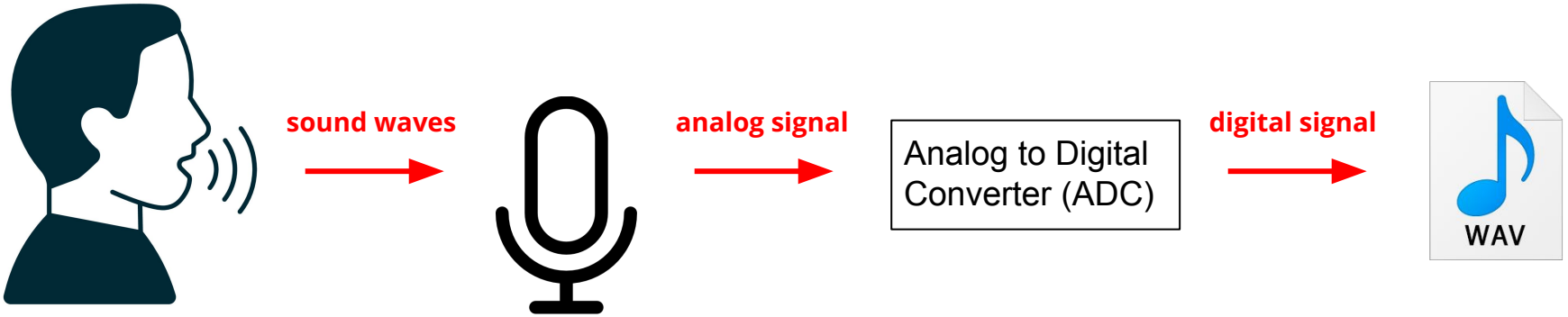
Università degli
Studi di Milano-Bicocca

Audio

Prof. Flavio Piccoli - Dr. Mirko Paolo Barbato

Introduction

What is an audio signal?



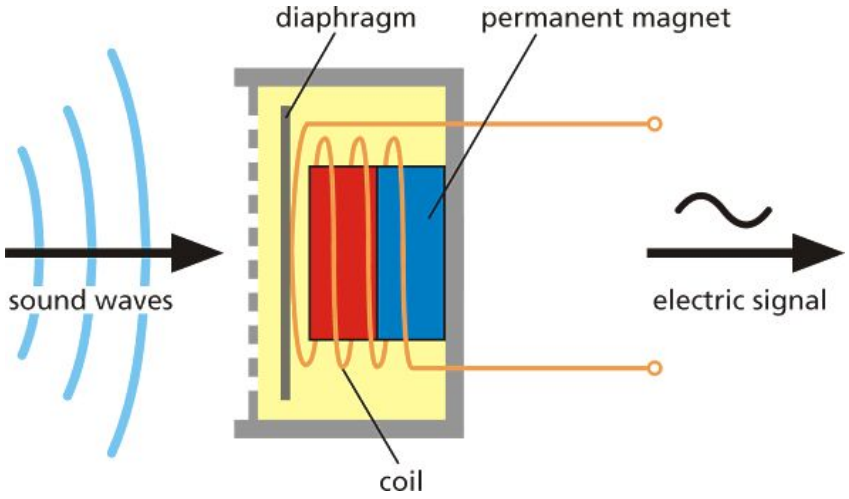
Microphone

- Sound waves are converted into an electrical signal through the microphone

- There are three types
  - dynamic microphones
  - condenser microphones
  - ribbon microphones

Dynamic microphones

- sound wave moves the diaphragm back and forth
- coil moves near the magnet, inducing an alternating electric current
- transmitted through the microphone cable to a preamplifier where its level is boosted



Condenser Microphone

- designed to pick up vocals and high frequencies

- capacitor: is made up of
  - two suspended, lightweight metal plates (diaphragm and back plate)
  - a condenser capsule that sits between them

- thin diaphragm
  - increased sensitivity
  - more fragile

- When acoustic sound waves reach the diaphragm, the sound pressure causes it to vibrate against the back plate. This in turn causes the voltage between them to fluctuate.

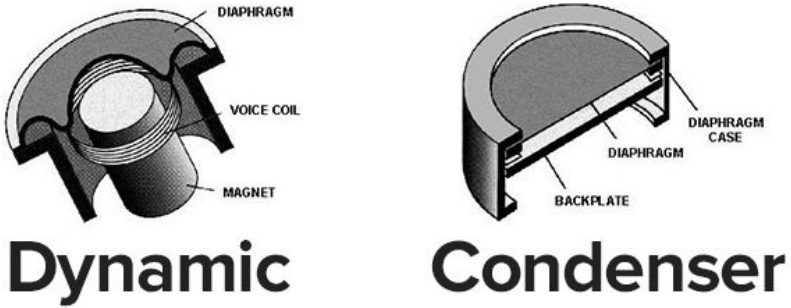


Image source: ledgernote.com

Ribbon Microphone

- extended rectangular diaphragm mede of thin aluminium with magnets at either end
- when sound waves hit it, it vibrates to create an electrical charge
- most ribbon microphones are only bi-directional

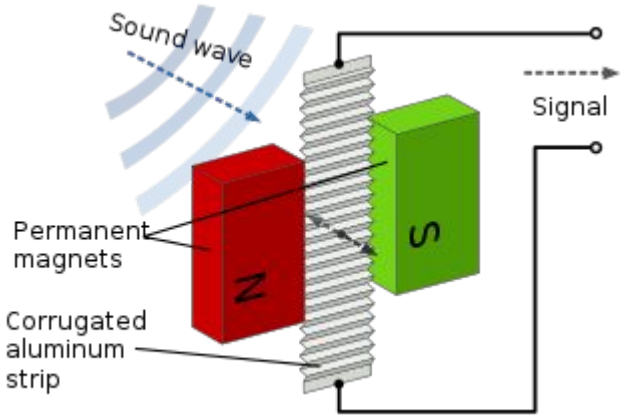
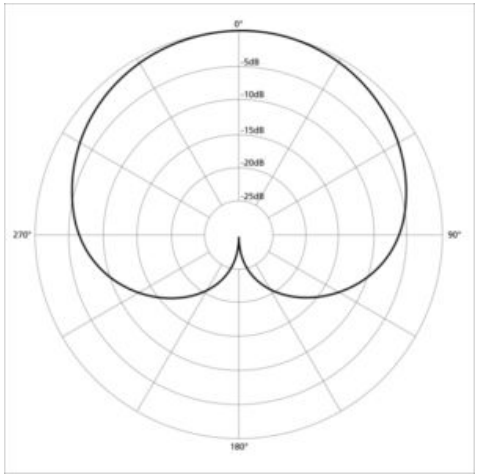


Image source: ledgernote.com

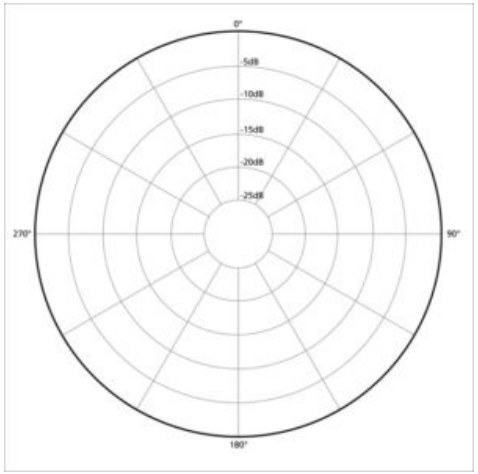
The polar pattern

- most crucial spec of any microphone
- it is the direction from which the microphone is able to pick the sound

unidirectional



omnidirectional



bi-directional

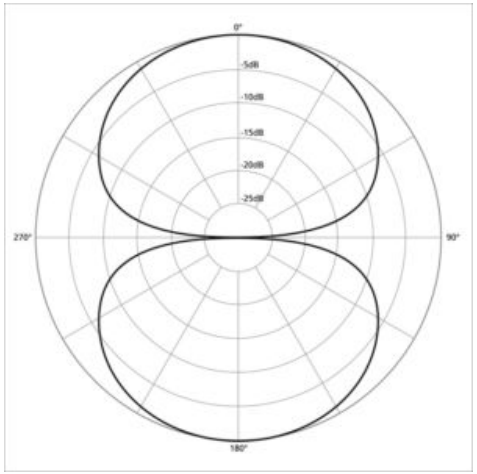


Image source: ledgernote.com

The proximity effect

- it is not the title of a movie
- it affects all mics except those with omnidirectional patterns
- increase in low-frequency response the closer the mic gets to the source
- voice seems deeper when microphone is close to the mouth

Image source: ledgernote.com

Digitalization of a signal

Two operations are needed:

- **sampling**: acquire the value of the signal every n milliseconds (i.e. at a sampling frequency)
- **quantization**: adjust the level of each sample to one of the possible values of amplitude

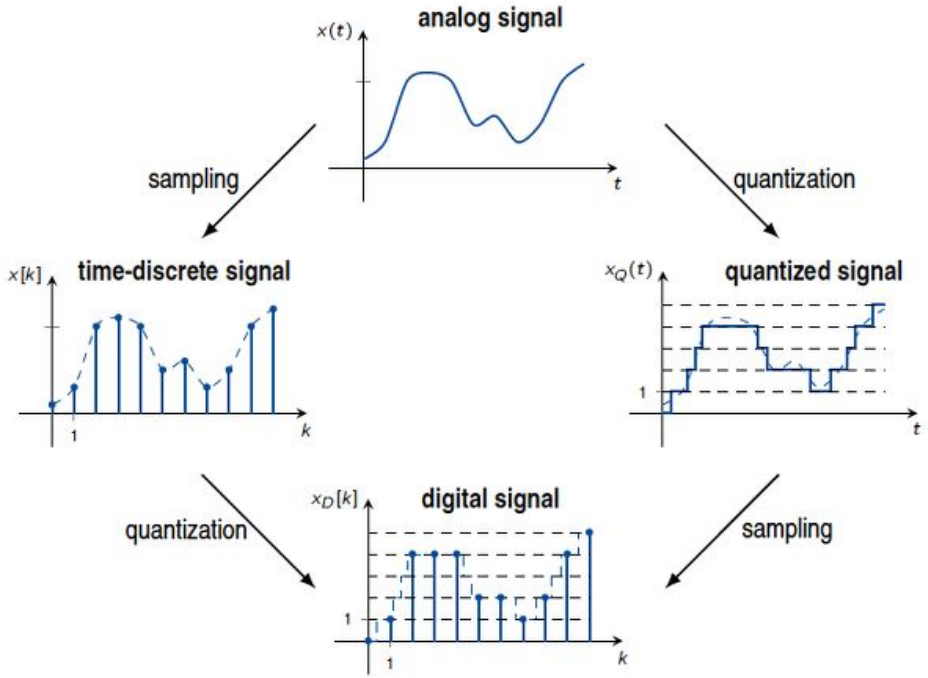


Image credits: https://www.javatpoint.com/difference-between-analog-signals-and-digital-signals

Numerical representation of the samples

- Each possible level of quantization is represented by a set of bits
- Each sample is then represented by the corresponding level represented in binary code

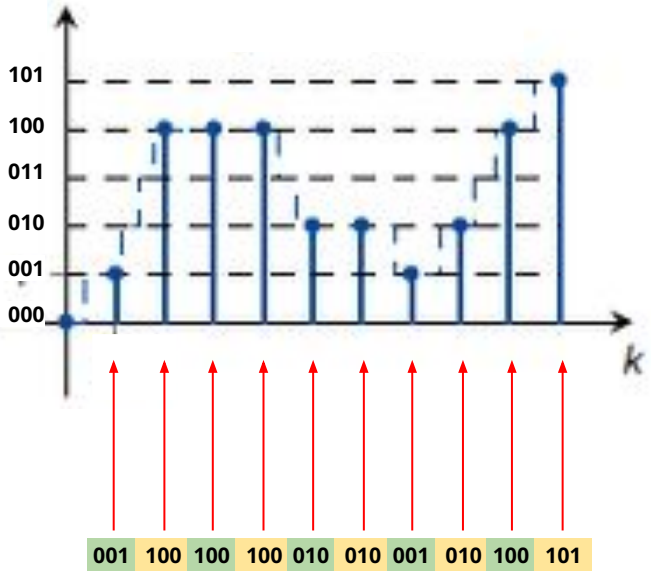


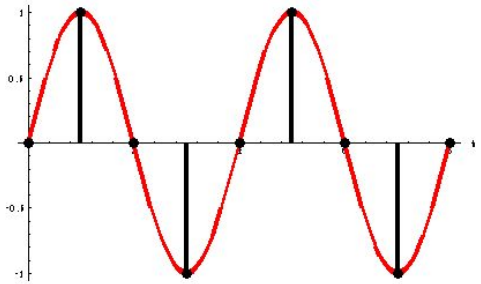
Image credits: https://www.javatpoint.com/difference-between-analog-signals-and-digital-signals

Nyquist theorem (a.k.a. the sampling theorem)

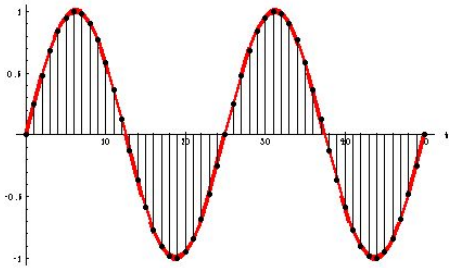
- The sampling frequency f_s must be at least twice the highest frequency f_{max} present in the signal

$$f_s >= 2 f_{max}$$

Incorrect sampling frequency



Correct sampling frequency



Dimension of the audio

The dimension of an audio file is directly proportional to:

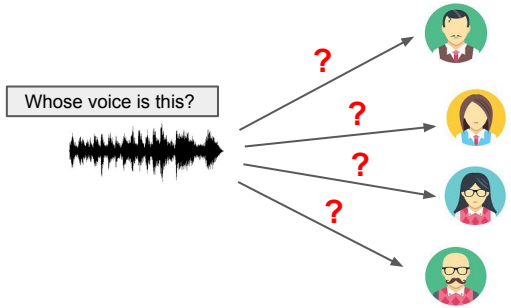
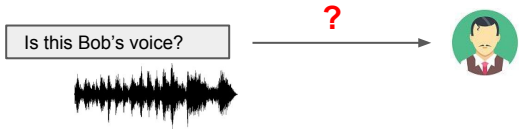
- its duration
  - longer audio → more samples
- the sampling frequency
  - higher freq. → more samples
- the levels of quantization
  - more levels → more bits needed for repr. a sample

$$d = t \times f_c \times q$$

$$byte = s \times \frac{samples}{s} \times \frac{byte}{sample}$$

$d = dimension$
$t = time$
$f_s = sampling\ \ freq.$
$q = bits\ for\ quantization$

Speaker identification vs speaker verification

	Speaker identification	Speaker verification
Schema		
Goal	Determine who is talking from a set of known speakers	Determines whether the person is who he/she claims to be
Identity claim	No	Yes
Source of the voice	From a known speaker (closed-set identification)	From a unknown speaker (open-set identification)

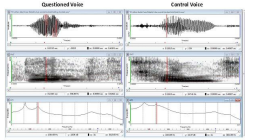
Applications

- **Authentication**
  - Confidential information protection

- **Surveillance**
  - Recognizing the target speakers who are of interest to the security agency

- **Personalization**
  - Limit the services of a digital assistant depending on the user's voice

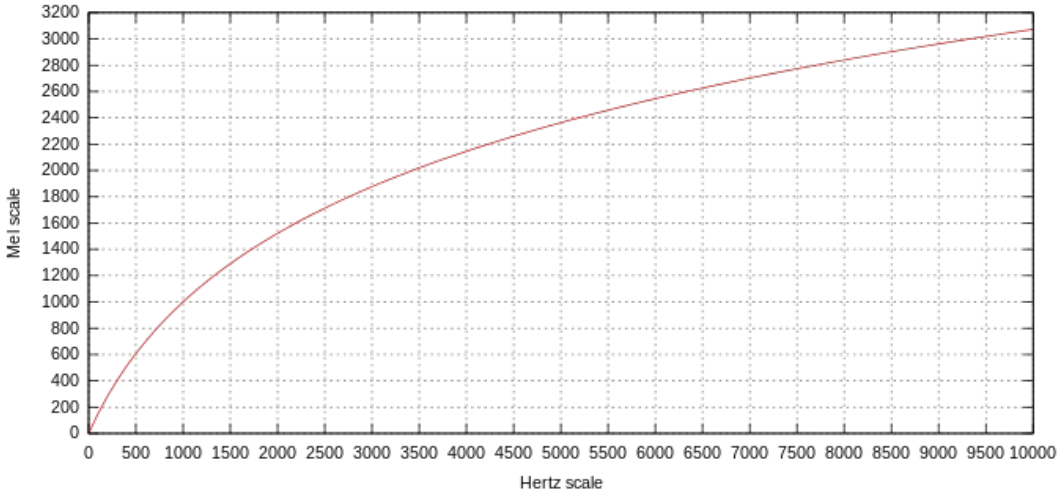
- **Forensics**
  - Comparison between a voice sample recorded during a crime and the voice of the suspect



- **Multi-speaker tracking**
  - Locate and track a person of interest's voice to silence other sources

Mel Scale

- it is a perceptual scale of pitches judged by listeners to be equal in distance from one another
- we are more sensitive to lower frequencies than higher frequencies



$$mel(f) = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right)$$

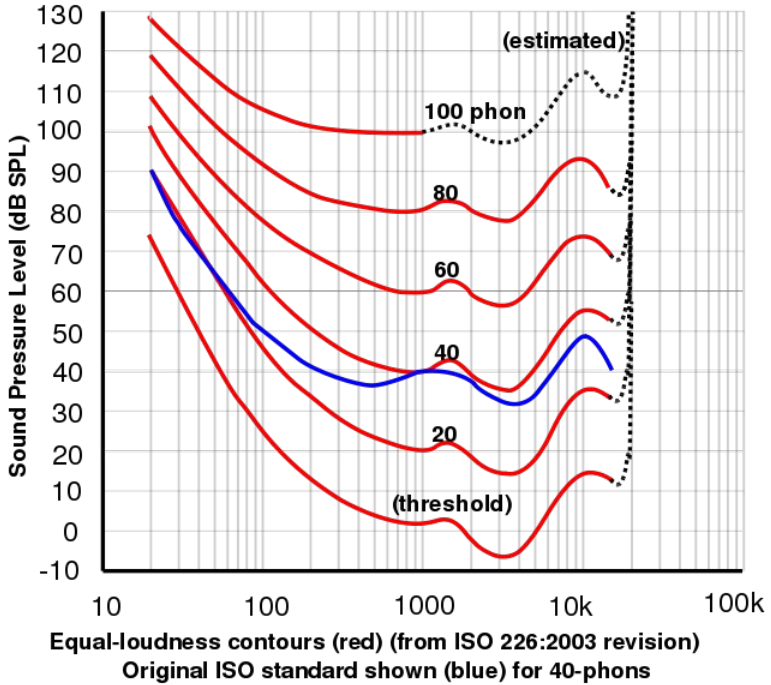
Human perception

Description

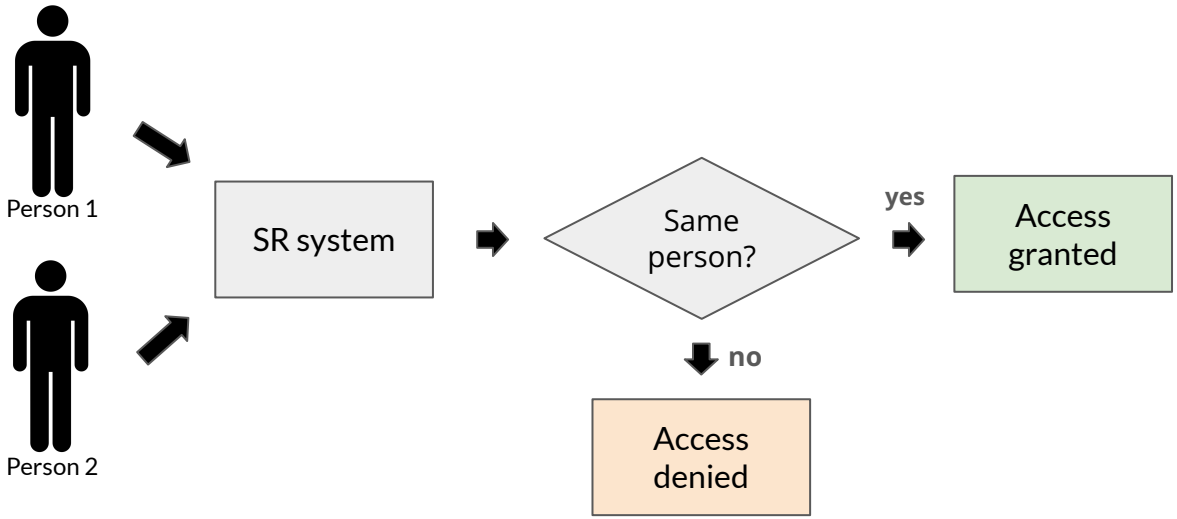
- each line represents the same level of loundness perceived by humans
- loudness is expressed in phon
- each line is in function of sound pressure level over the frequency spectrum

Deductions

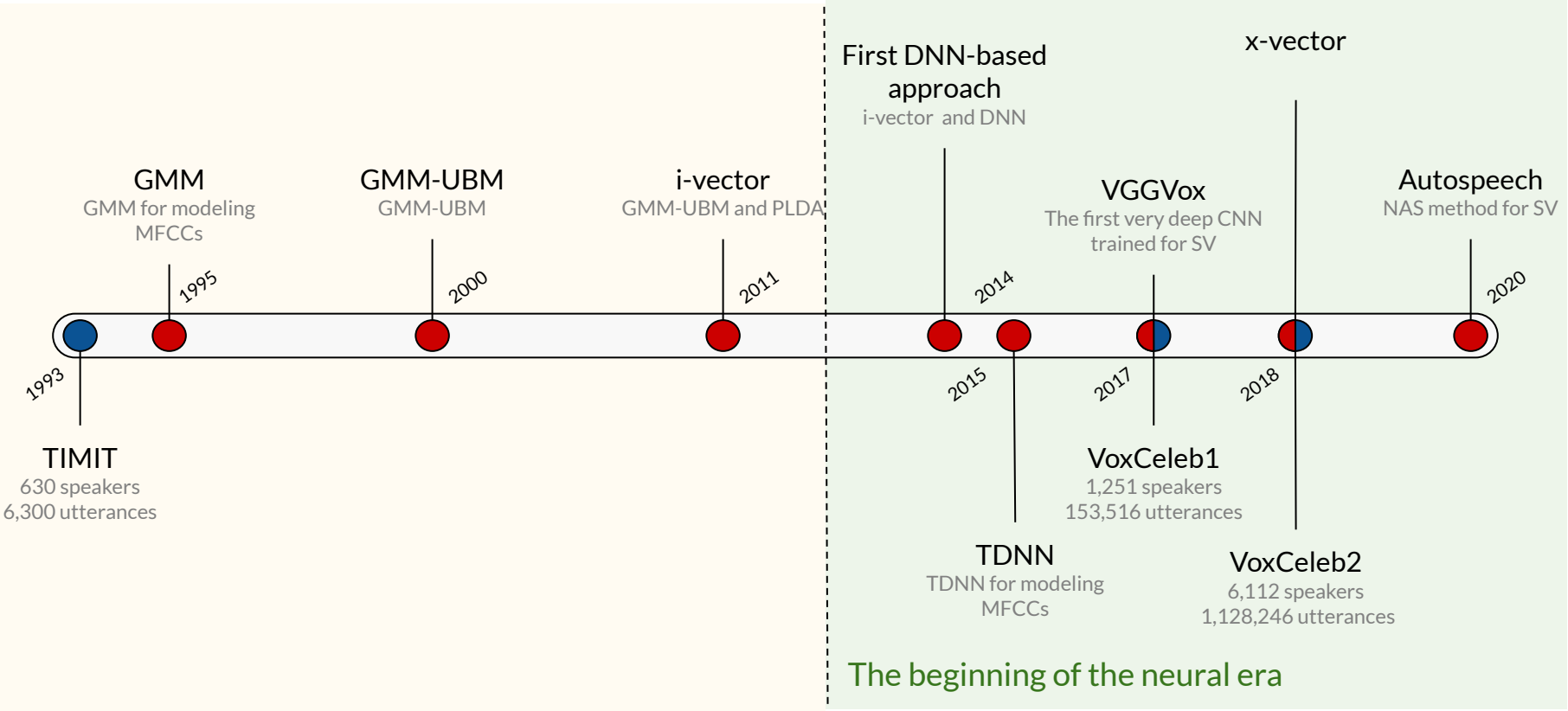
- for equal loudness, lower frequencies require more sound pressure
- the trend is not constant
- there is a sweet spot between 3k and 4k



Speaker recognition (SR)

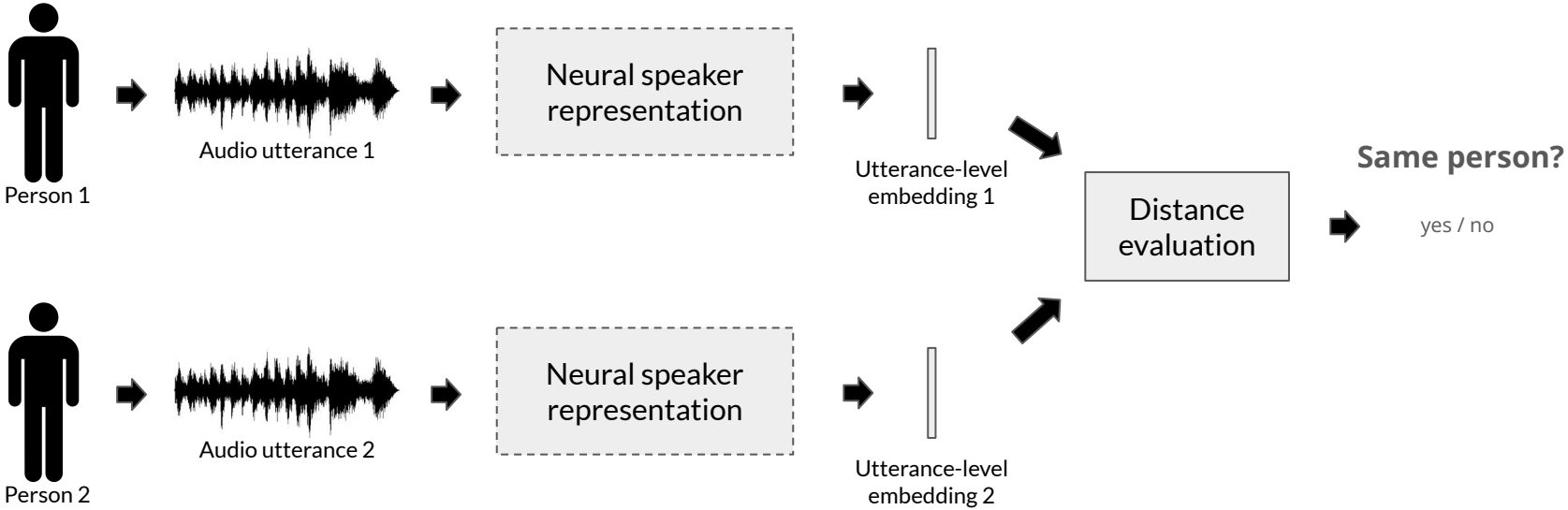


A brief history of speaker recognition

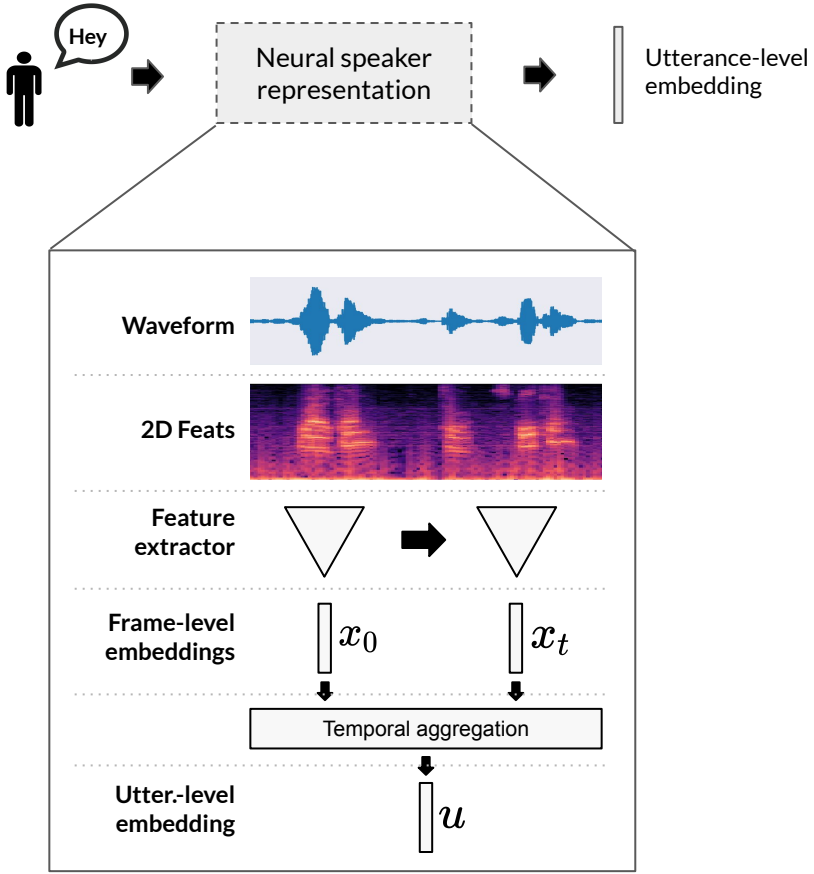


Neural speaker verification

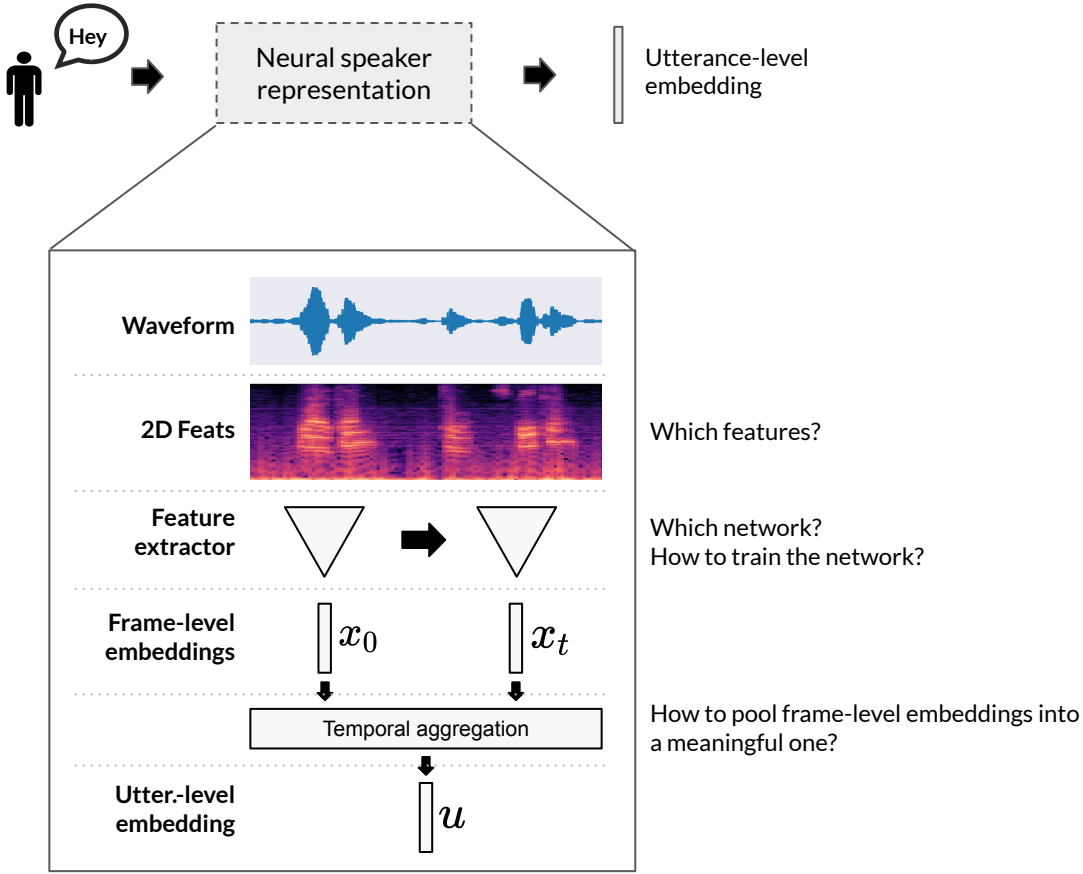
Are these two voices belonging to the same person?



Neural speaker representation

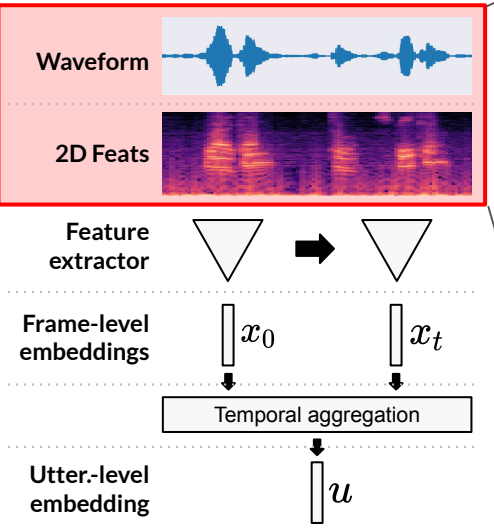


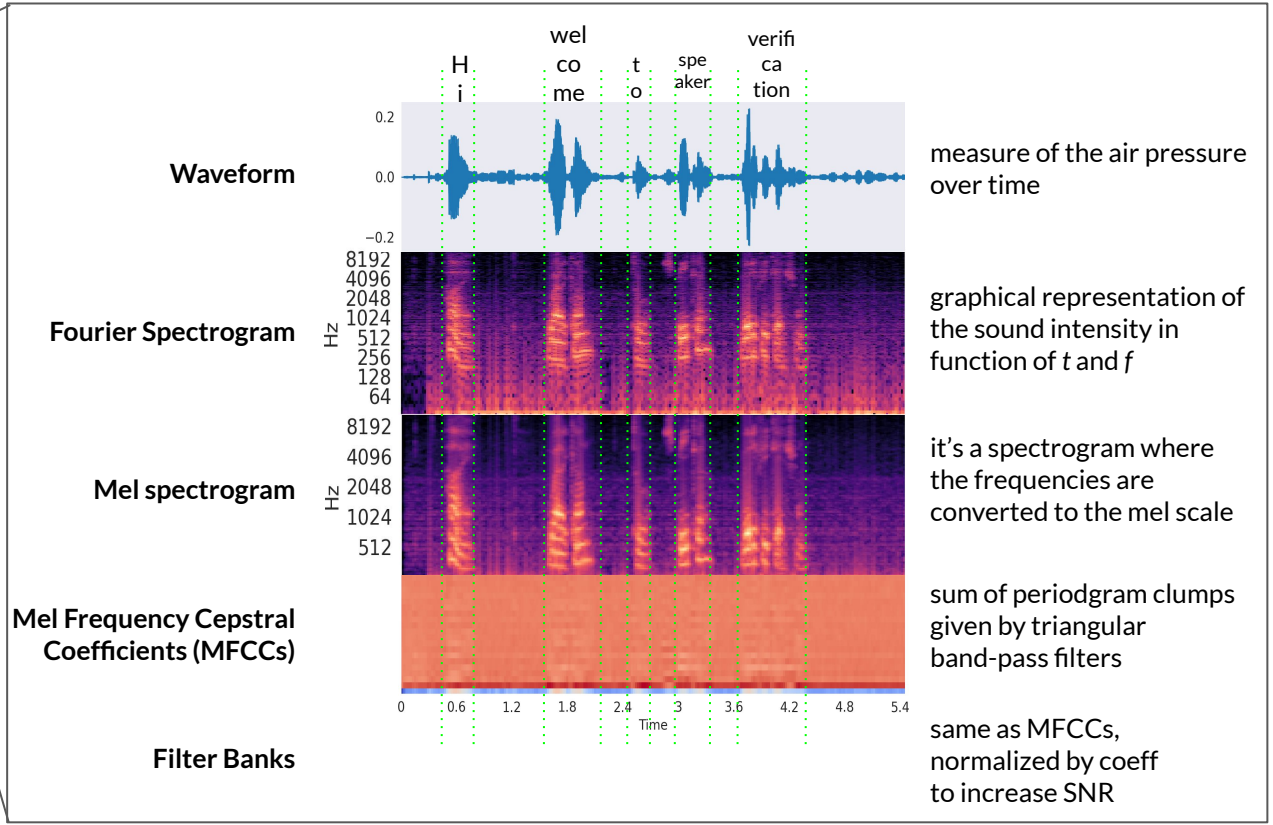
Neural speaker representation



Acoustic features

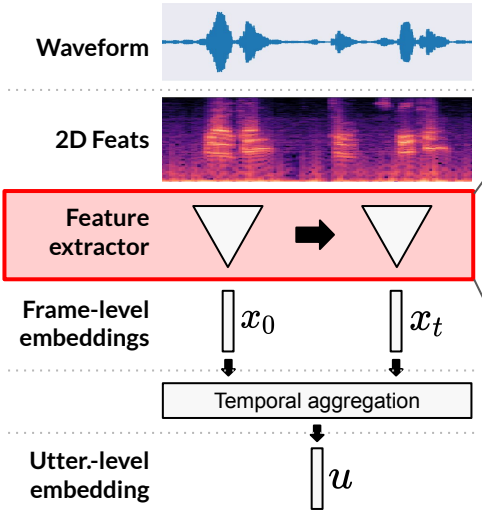
Acoustic features are a low-level representation of the input signal



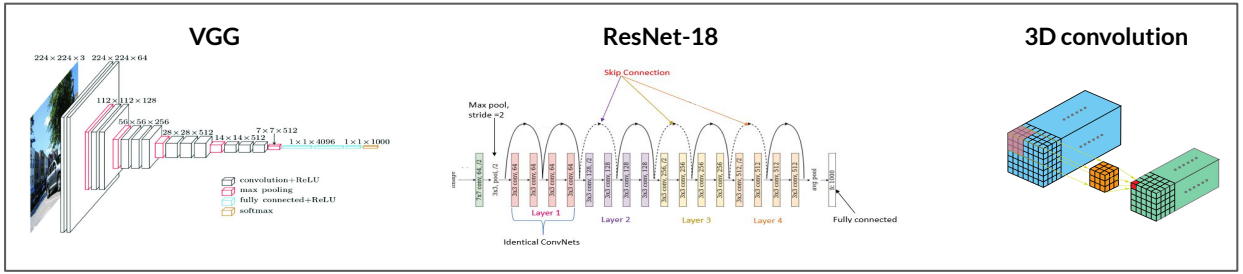


Backbone

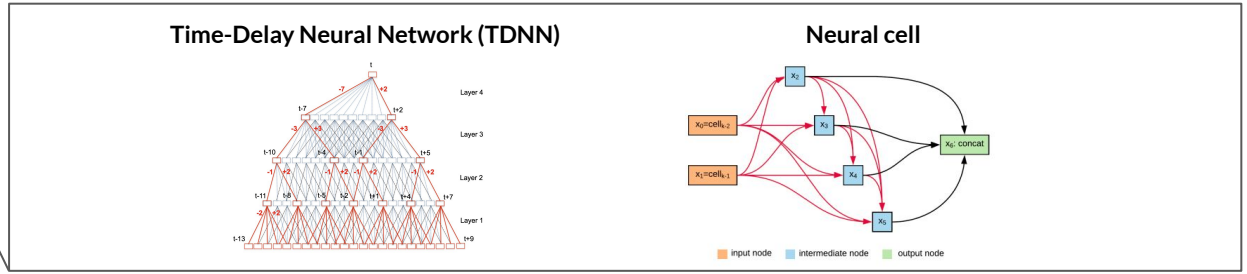
Features are extracted through a neural network, here called **backbone**



Architectures inherited from image recognition

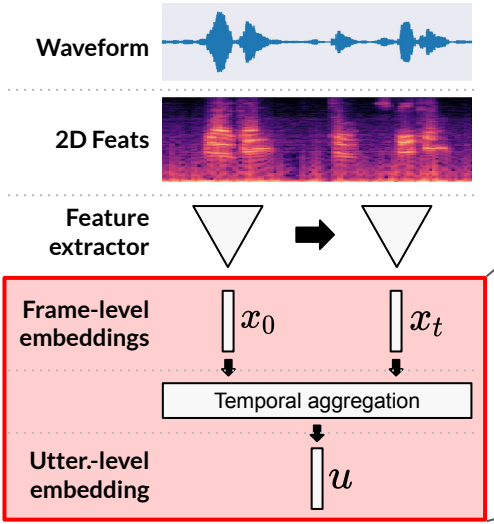


Architectures designed for speaker recognition



Temporal aggregation

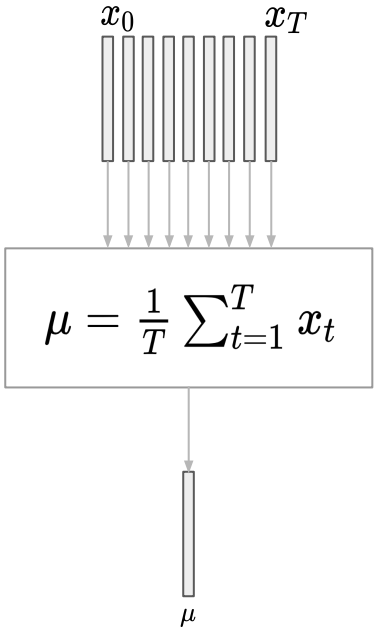
It is the strategy for merging frame-level descriptors into a single utterance-level descriptor



- Parameter-free layers
  - Temporal Average Pooling (TAP)
  - Global Statistical Pooling (GSP)
- Learnable layers
  - Self-attentive Pooling Layer (SAP)
  - Attentive Statistical Pooling (ASP)
  - Vector of Locally Aggregated Descriptors (VLAD)

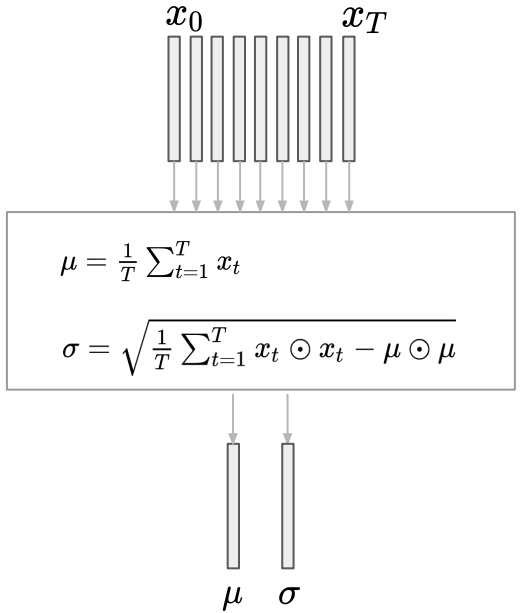
Parameters-free temporal aggregators

Temporal Average Pooling (TAP)



Exploring the encoding layer and loss function in end-to-end speaker and language recognition system.
Cai, W., Chen, J. and Li, M., 2018.
arXiv preprint arXiv:1804.05160.

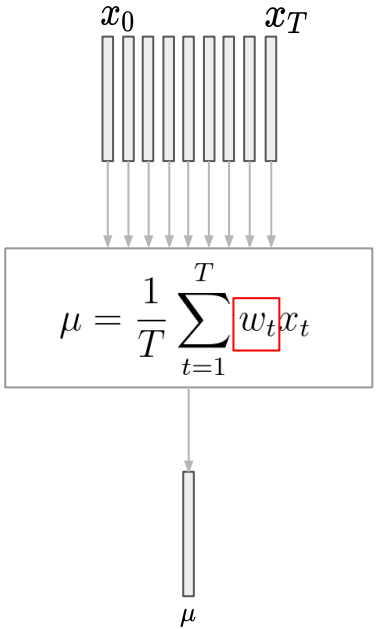
Global Statistical Pooling (GSP)



Deep neural network embeddings for text-independent speaker verification.
Snyder, D., Garcia-Romero, D., Povey, D. and Khudanpur, S., 2017, August.
In Interspeech (Vol. 2017, pp. 999-1003).

Learnable temporal aggregators (attention-based)

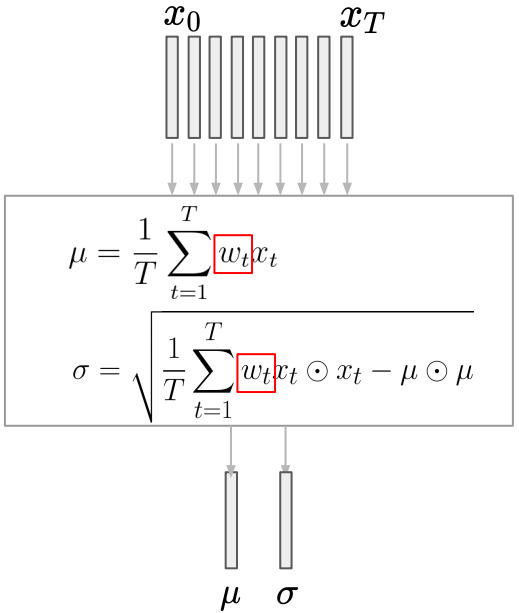
Self-attentive Pooling Layer (SAP)



w_t is the attention

Exploring the encoding layer and loss function in end-to-end speaker and language recognition system
Cai, W., Chen, J. and Li, M., 2018
arXiv preprint arXiv:1804.05160

Attentive Statistical Pooling (ASP)

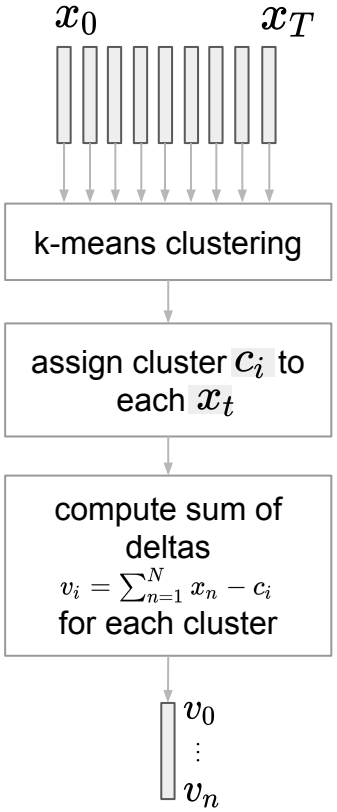


Attentive statistics pooling for deep speaker embedding
Okabe, K., Koshinaka, T. and Shinoda, K., 2018
arXiv preprint arXiv:1803.10963

Learnable temporal aggregators (dictionary-based)

Vector of Locally Aggregated Descriptors (VLAD)

1. Clusters frame-level embeddings
2. descriptor composed by average distance of cluster samples w.r.t. cluster center



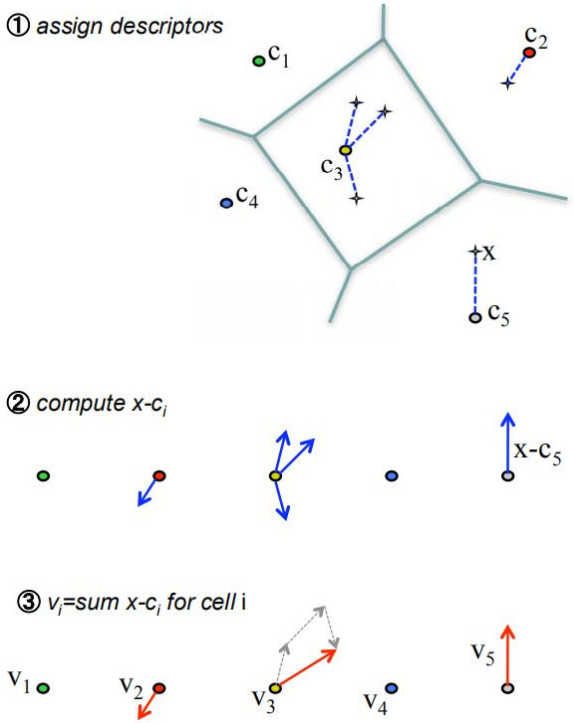


Image source: https://hal.inria.fr/inria-00548637/file/jegou_compactimagerepresentation_slides.pdf

Parameters of temporal aggregation strategies

- Let n be the length of a frame-level embedding
- TAP / GSP are parameter-less
- SAP and ASP have $n\,(n+2)$ parameters
  - $n\,(n+1)$ for the projection in the latent space
  - n for the context vector (self attention)
- VLAD has c parameters, where c is the number of clusters chosen

Loss functions

- **Classification objectives:** the discriminative power of the features is the result of classification
  - Cross-entropy

- **Classification objectives that constrain the embedding space:** the learning of the embedding space is constrained to make the features more discriminating and improve the classification
  - Angular softmax (A-Softmax *a.k.a.* SphereFace)
  - Additive margin softmax (AM-Softmax *a.k.a.* CosFace)
  - Additive angular margin softmax (AAM-Softmax *a.k.a.* ArcFace)

- **Metric learning objectives:** the embedding space is formed taking into account the relationships between the samples
  - Contrastive
  - Triplet
  - (Angular) Prototypical

Loss functions

- **Classification objectives:** the discriminative power of the features is the result of classification
  - Cross-entropy

- **Classification objectives that constrain the embedding space:** the learning of the embedding space is constrained to make the features more discriminating and improve the classification
  - Angular softmax (A-Softmax *a.k.a.* SphereFace)
  - Additive margin softmax (AM-Softmax *a.k.a.* CosFace)
  - Additive angular margin softmax (AAM-Softmax *a.k.a.* ArcFace)

- **Metric learning objectives:** the embedding space is formed taking into account the relationships between the samples
  - Contrastive
  - Triplet
  - (Angular) Prototypical

Loss functions constraining the embedding space

- Classification objectives that constrain the embedding space
  - Angular softmax (A-Softmax *a.k.a.* SphereFace)
  - Additive margin softmax (AM-Softmax *a.k.a.* CosFace)
  - Additive angular margin softmax (AAM-Softmax *a.k.a.* ArcFace)

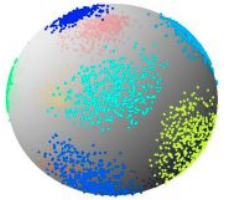
$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

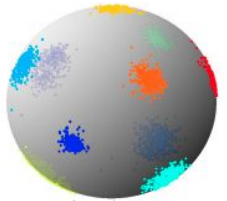
	s	m_1	m_2	m_3
A-Softmax	1	m	0	0
AAM-Softmax	s	0	m	0
AM-Softmax	s	0	0	m

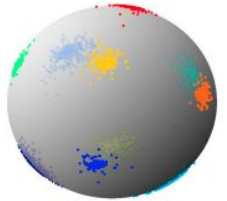
with m > 0 and s > 1

s → scale factor
θ_j → is the results of the dot product of normalised vector **W** and **x**
m_1 → multiplicative angular margin
m_2 → additive angular margin
m_3 → additive cosine margin

NOTE: W is learnable


cross-entropy


A-Softmax


AM-Softmax

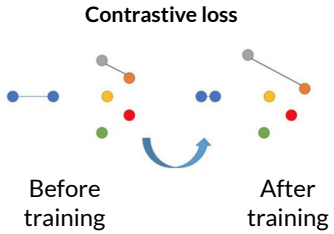
Loss functions using metric learning

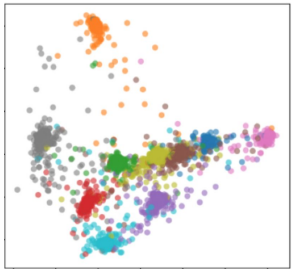
- Metric learning objectives
  Given a mini-batch of M utterances from each of N different speakers, whose embeddings are $\mathbf{x}_{j,i}$, where $1 \leq j \leq N$ and $1 \leq i \leq M$

  - **Contrastive**

$$L_{\mathrm{C}} = \frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{K} \mathbb{1}_{j=k} \left\| \mathbf{x}_{j,0} - \mathbf{x}_{k,1} \right\|_2^2 +$$

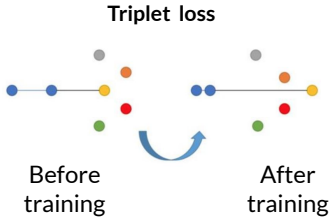
$$\mathbb{1}_{j \neq k} \max(0, m - \left\| \mathbf{x}_{j,0} - \mathbf{x}_{k,1} \right\|_2)^2$$

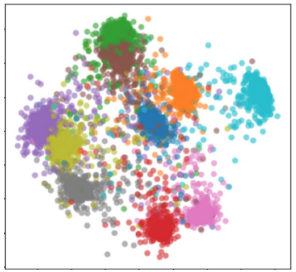




  - **Triplet**

$$L_{\mathrm{T}} = \frac{1}{N} \sum_{j=1}^{N} \max(0, \left\| \mathbf{x}_{j,0} - \mathbf{x}_{j,1} \right\|_2^2 - \left\| \mathbf{x}_{j,0} - \mathbf{x}_{k \neq j,1} \right\|_2^2 + m)$$



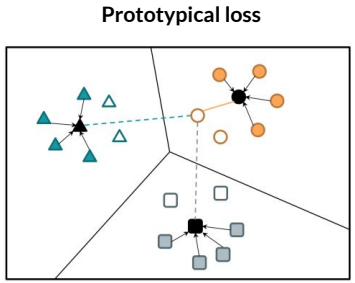


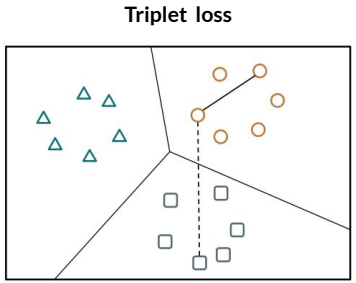
Loss functions using metric learning

- **(Angular) Prototypical**

$$L_{\mathrm{P}} = -\frac{1}{N}\sum_{j=1}^{N}\log\frac{e^{\mathbf{S}_{j,j}}}{\sum_{k=1}^{N}e^{\mathbf{S}_{j,k}}}$$

$$\mathbf{S}_{j,k} = \|\mathbf{x}_{j,M} - \mathbf{c}_k\|_2^2 \qquad \mathbf{c}_j = \frac{1}{M-1}\sum_{m=1}^{M-1}\mathbf{x}_{j,m}$$





Benchmarking

Evaluation of a speaker verification system

We measure and compare neural methods for SV in terms of:

- **Effectiveness**
  - Equal Error Rate (EER)
  - Minimum Detection Cost Function (MinDCF)

- **Efficiency**
  - Model complexity
  - memory usage
  - computational cost
  - inference time

Minimum Detection Cost Function (MinDCF)

- limitation: it has parameters that imply a particular application of the speaker detection technology

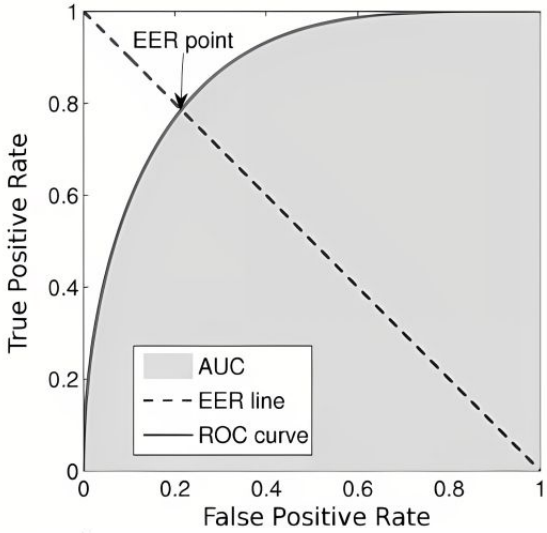
$$MinDCF = C_m \times P_m \times P_{tar} + C_{fa} \times P_{fa} \times (1 - P_{tar})$$

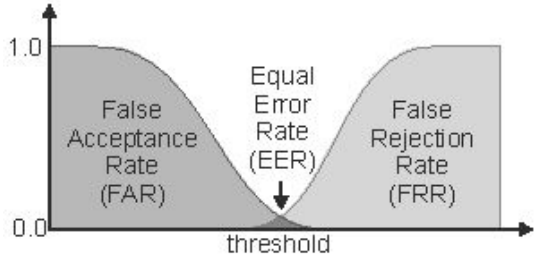
Parameters:
P_{tar} = 0.05 ← probability that a trial should be classified as true
C_m = 1 ← cost of misses
C_{fa} = 1 ← cost of false alarms

Inputs:
P_m ← probability of misses
P_{fa} ← probability of false alarms

EER

- metric used in biometric security systems to measure the effectiveness of the system in identifying individuals correctly
- As the number of false acceptances (FAR) goes down, the number of false rejections (FRR) will go up and vice versa





1. For a given threshold, compute TP, TN, FP, FN:

$$TP \iff e_i = g_i - 1$$
$$TN \iff e_i = g_i - 0$$
$$FP \iff e_i = 1 \wedge g_i - 0$$
$$FN \iff e_i = 0 \wedge g_i = 1$$

2. Compute TPR, FPR, FNR

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \qquad FNR = 1 - TPR$$

3. Compute accuracy (ACC) and equal error rate (EER)

$$ACC = \frac{TP + TN}{P + N}$$

$$EER = \frac{FAR_i + FRR_j}{2} \qquad i, j = min_{i,j} \left| FAR_i - FRR_j \right|$$

Training datasets

	AMI [8]	TIMIT [9]	VoxCeleb1 [10]	VoxCeleb2 [3]
# of POIs	24	630	1,251	6,112
# of utterances	N/A	6,300	153,516	1,128,246
# of hours	100	5.4	352	2,442

- All the audio samples have been resampled at 16 kHz for the experiments
- No Voice Activity Detection (VAD)
- Methods trained on the VoxCeleb1 dataset consider only 1,211 speakers of the **development set**
- Methods trained on VoxCeleb2 consider the **entire dataset** as there is no overlap with the testing data

Methods

	Audio encoding	Backbone	Aggregation	Emb. dim	Loss (+ → 2step)
ResNet-18 [1]	Spectrogram	ResNet-18	TAP	512	CE
ResNet-34 [1]	Spectrogram	ResNet-34	TAP	512	CE
SincNet [2]	Raw Waveform	SincNet	TAP	2048	CE
VGGVox [3]	Spectrogram	VGG-M	TAP	1024	CE+Contrastive
CNN-3D [4]	MFEC	LCN	TAP	128	CE
AutoSpeech [1]	Spectrogram	NAS-derived	-	2048	CE
MobileNet-v2 [5]	Raw Waveform	MobileNet-v2	ASP	320	AM-Softmax
ResNet-SE-34L [6]	Mel Spectrogram	ResNet-SE-34	SAP	512	Ang. Prototypical
ResNet-SE-34-v2 [7]	Mel Spectrogram	ResNet-SE-34	ASP	512	CE+Ang. Proto.

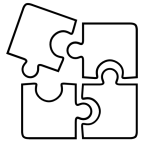
Results

- The table shows eleven rows since the *SincNet* method was trained on three datasets (i.e. AMI, TIMIT and VoxCeleb2)
- The methods trained on the VoxCeleb2 achieves the best results
- *MobileNet-v2* achieves an EER 38% lower than the best method, which is *ResNet-SE-34-v2*
- *AutoSpeech* which consists of a shallow backbone learned by NAS achieves results comparable to very deep backbones

Modello	EER (%)	Min. DCF (%)
MobileNet	39.92	96.29
SincNet-TIMIT	32.73	94.24
SincNet-AMI	30.37	93.70
CNN 3D	21.88	87.08
VGGVox	14.14	72.33
ResNet-18	10.21	61.11
ResNet-34	6.68	47.22
SincNet-Vox2	4.03	26.61
AutoSpeech	2.48	15.92
ResNet-SE-34-L	2.21	17.06
ResNet-SE-34-V2	1.18	8.65

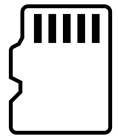
Method Efficiency

- The **method complexity** is measured in terms of the total amount of learnable parameters
  - This metric is expressed in Megabytes (MB)
  - This information is useful for understanding the *minimum amount of GPU memory required* for each method



- The **computational cost** of each method considered is estimated using the floating-point operations (FLOPs) in the number of multiply-adds

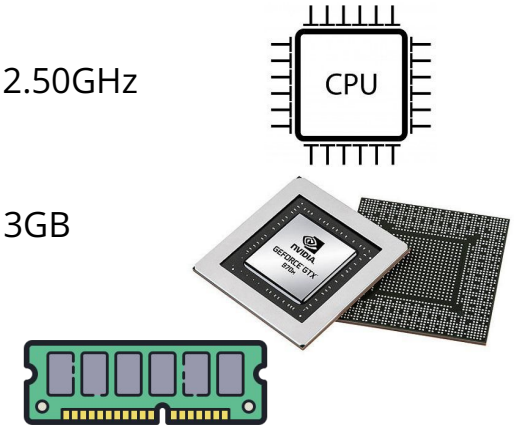
- The **memory usage** estimates the memory allocated by *the method and the memory required while processing the batch*
  - This metric is reported in Gigabytes (GB)



- The **inference time** required for processing a batch in CPU is computed
  - This measure is expressed in seconds
  - For statistical validation the reported time corresponds to the average over 10 runs

Method Efficiency

- *Computational cost*, *memory usage* and *inference time* are computed by processing 3-seconds utterances
- All the experiments were performed on a PC with the following characteristics
  - **CPU:** Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz
  - **GPU:** NVIDIA GeForce GTX 970M Maxwell 3GB
  - **RAM:** 2 × 8GB DDR4 RAM 2400 MHz



EER vs. Computational cost vs. Method complexity

Timeslot analysis

Researched questions

1. How do the temporal aggregators behave in function of the utterances lengths?

2. Is there a post-training trick that can improve the performance?

Dataset & configuration of the experiments

- It has been used the VoxCeleb2 dataset:

Identities	6112
Males	3761
Spoken hours	2442
Samples	1’128’246
Average num. of samples for identity	185
Average length of samples (s)	7.8

- Starting from [1], we replace the temporal aggregator with the t.a. under analysis, and we retrain the system.

[1] In defence of metric learning for speaker recognition
Chung et al.
Interspeech

Overall performance

- SAP scored the best EER and ACC on the VoxCeleb2
- Attention-based temporal aggregators perform better than the others

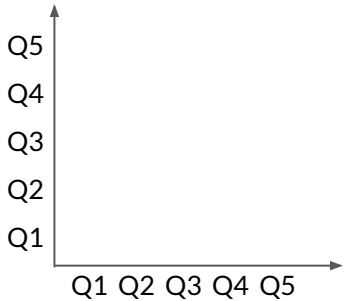
	EER ↓	**ACC (%) ↑**	**FPR (%) ↓**	**FNR (%) ↓**
TAP	3.41	96.34	4.99	0.45
GSP	3.42	95.60	5.00	0.45
SAP	2.76	97.08	5.01	0.27
ASP	2.90	96.06	5.03	0.29
VLAD	4.63	95.31	4.99	0.80

Timeslot analysis

- Each sample is composed of 2 utterances (declared and under-inspectio)
- Lengths have been divided into 5 quantiles on the basis of their duration:

$$\underset{Q1}{[4.0, 4.6)}, \underset{Q2}{[4.6, 5.7)}, \underset{Q3}{[5.7, 7.4)}, \underset{Q4}{[7.4, 10.6)}, \underset{Q5}{[10.6, 69.04]}s$$

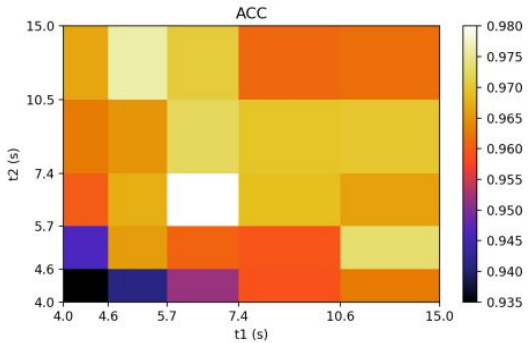
- This results in 25 groups. Each group contains about 1450 samples:



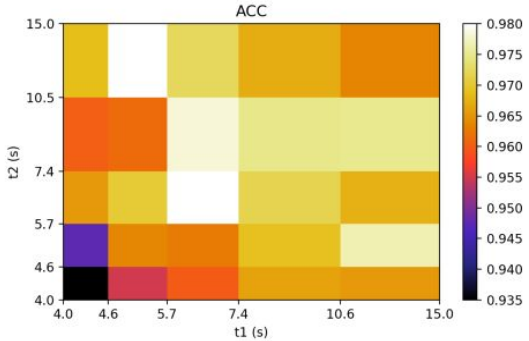
Timeslot analysis (accuracy)

Parameter-free

TAP

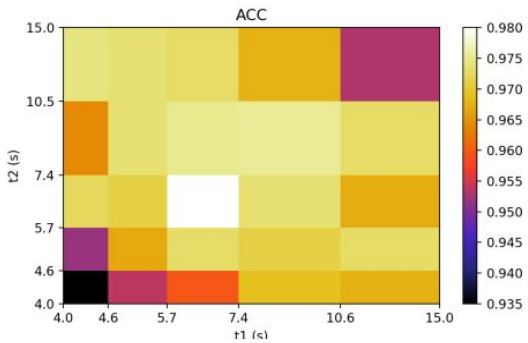


GSP

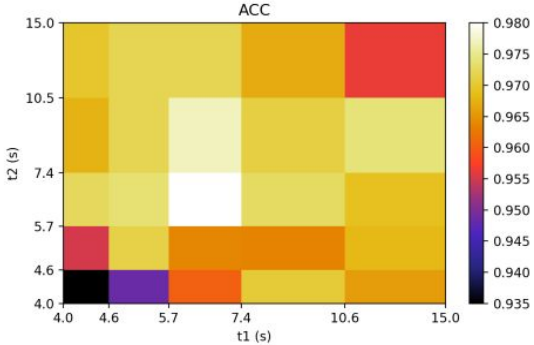


Attention-based

ASP

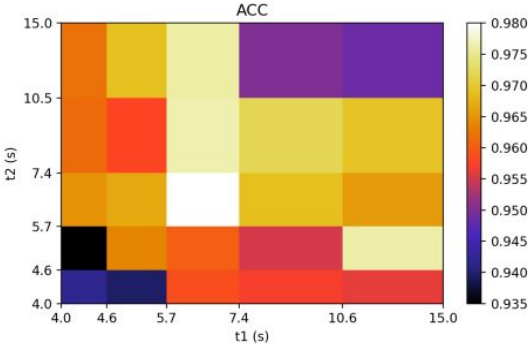


SAP



Dictionary-based

VLAD



Post-training improvement

- All aggregators present the maximum accuracy in the interval [5.7, 7.4)
- We cropped utterances longer than 7.4s to 7s to make them fall in the sweet-spot
- Results show that accuracy improves, especially for ASP and VLAD:

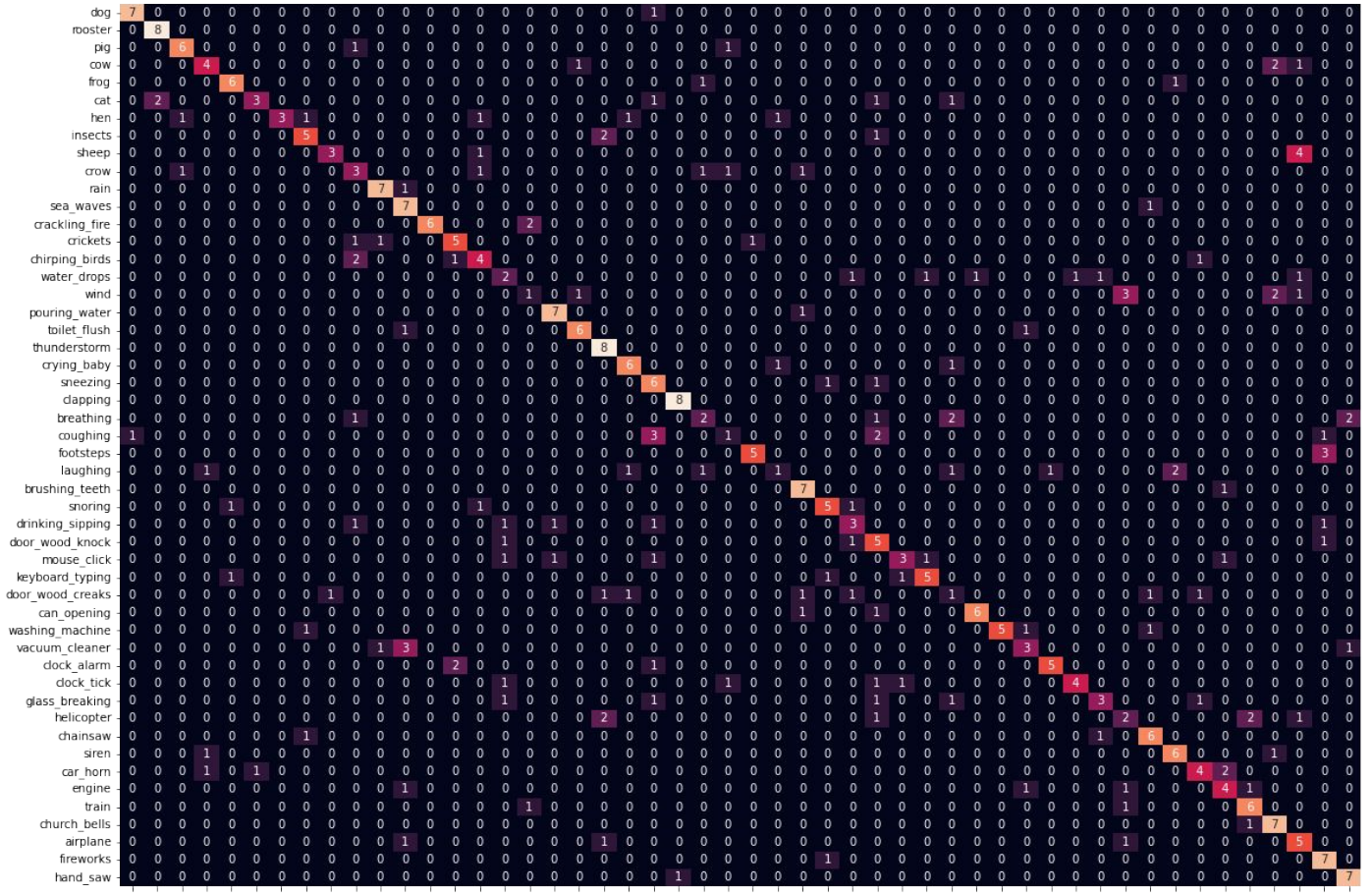
TAP		GSP		ASP		SAP		VLAD	
orig.	crop	orig.	crop	orig.	crop	orig.	crop	orig.	crop
97%	97%	96%	97%	96%	98%	97%	97%	94%	97%
			+1%		+2%				+3%

Exercise

ESC-50: Dataset for Environmental Sound Classification

50 classes.

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw


dog
rooster
pig
cow
frog
cat
hen
insects
sheep
crow
rain
sea_waves
crackling_fire
crickets
chirping_birds
water_drops
wind
pouring_water
toilet_flush
thunderstorm
crying_baby
sneezing
clapping
breathing
coughing
footsteps
laughing
brushing_teeth
snoring
drinking_sipping
door_wood_knock
mouse_click
keyboard_typing
door_wood_creaks
can_opening
washing_machine
vacuum_cleaner
clock_alarm
clock_tick
glass_breaking
helicopter
chainsaw
siren
car_horn
engine
train
church_bells
airplane
fireworks
hand_saw
0
1
2
3
4
5
6
7
8

acc=60%