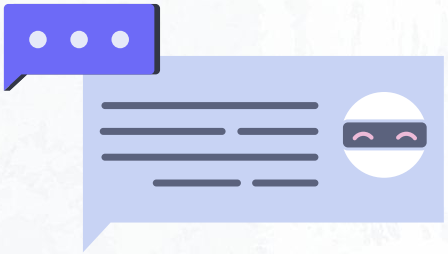
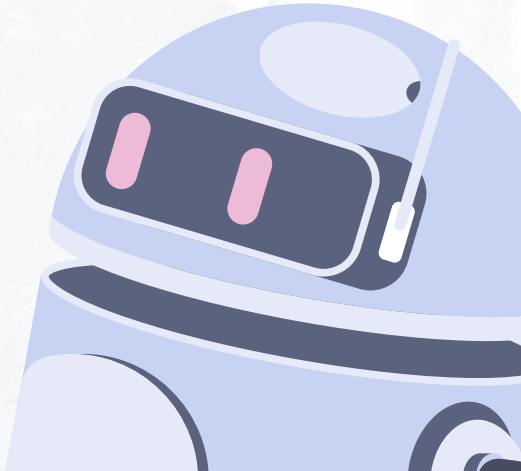
# Progetto FVAB

## **Lips**-based
## **Visual Speaker Recognition**
## using a **PNN**

Dario Trinchese 0522501493
Antonio Gravino 0522501502
Carmine Napolitano 0522501538

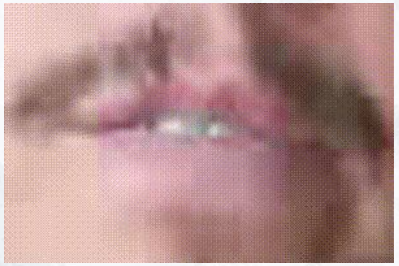




Questa presentazione è stata creata senza l'uso di AI

# Obiettivi

- **Rappresentare le informazioni della zona labiale** in maniera chiara e matematicamente rigorosa.
- Effettuare il **riconoscimento biometrico** mediante l’utilizzo di una **Probabilistic Neural Network**, utilizzando i dati ottenuti sia per l'**identificazione** che per la **verifica** di un **soggetto**, potendo così sia scoprire l’identità di un soggetto sconosciuto che confermare quella di un soggetto noto.
- Massimizzare le prestazioni della PNN.

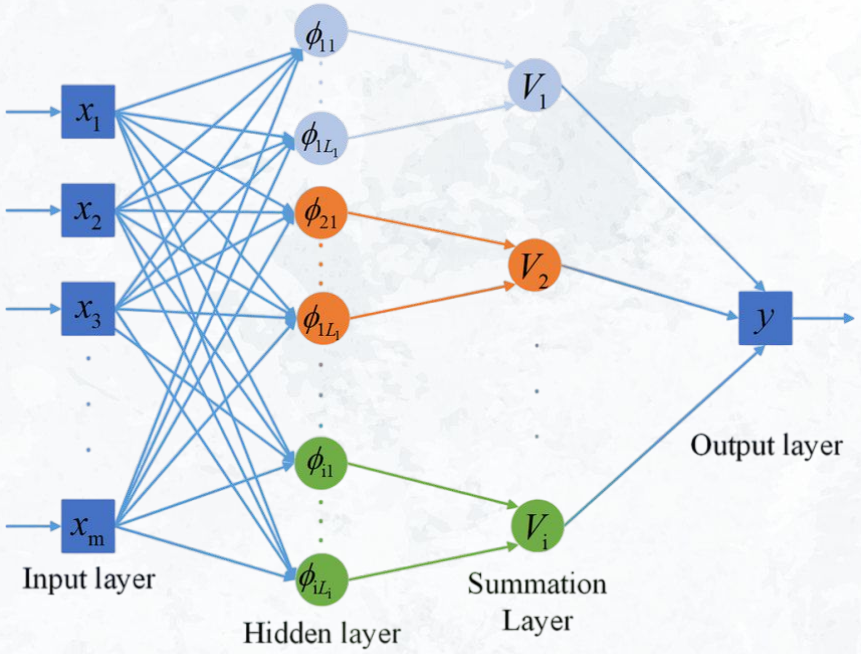


Soggetto **1_1_1_7** (Nicolò Balini)

Soggetto **1_1_2_9** (Marco Montemagno)

# Probabilistic Neural Network (PNN)

- Le PNN utilizzano un'architettura a più strati, con uno strato di input, uno o più hidden layer ed uno strato di output che produce le previsioni.
- La caratteristica distintiva delle PNN, rispetto alle classiche reti neurali, è che grazie al **summation layer** hanno capacità di fornire stime probabilistiche per l'appartenenza ad una determinata classe di un dato in input.
- Il **summation layer** è lo strato in cui le unità di base vengono sommate. Le **unità di base** sono dei vettori di addestramento che producono un valore di attivazione che è proporzionale alla similarità con il dato in input.

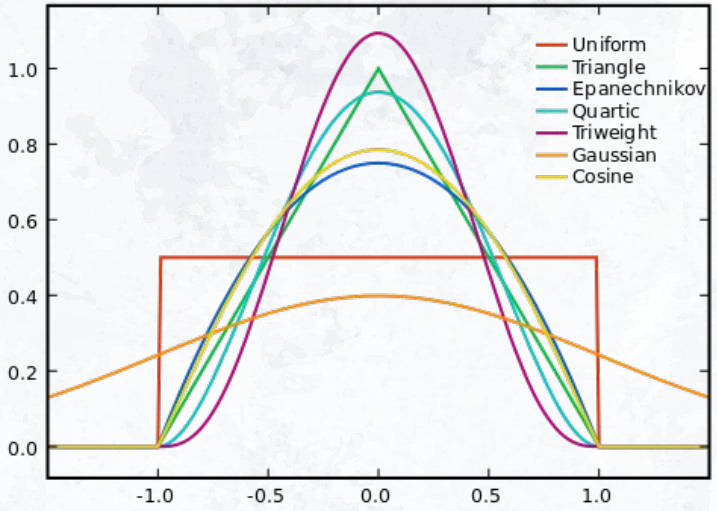


# Kernel

Il concetto di kernel nell'ambito delle reti neurali è molto importante, poiché un kernel stabilisce come viene valutata la somiglianza tra un dato in input ed i dati di addestramento.

Per la PNN sono stati implementati diversi kernel:

- **RBF, Laplaciano, Triangolare, Epanechnikov:** assegnano un peso più alto alle istanze più vicine all'unità di base, cioè più il dato in input è simile all'unità di base maggiore sarà il peso assegnato.
- **Uniforme:** assegna un peso costante a tutte le istanze all'interno di una determinata regione intorno all'unità di base.



# Dataset: Babele

**BABELE:**

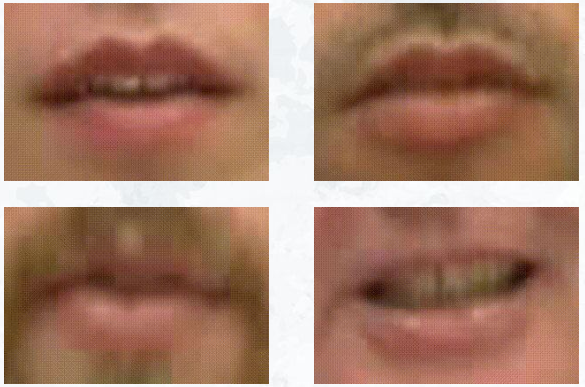
- Per ogni individuo, i suoi video di test e train sono clip di uno **stesso video originario**.
- Il suddetto dataset contiene video di labbra della durata di 10 secondi dove:
  - Per il **train** si hanno **4 video** per ogni soggetto.
  - Per il **test** si ha **1 solo video** per ogni soggetto.

**256** individui totali

# Dataset: VidTimit

**VIDTIMIT**:

- Sessioni di registrazioni distinte, per video di test e train.
- Individui registrati in occasioni diverse, ma sempre con lo **stesso dispositivo di acquisizione**, **posa** ed **illuminazione**.
- Circa *4 volte meno* individui distinti rispetto a BABELE.
- Di ogni individuo sono presenti 10 video; i primi **8***→ **video di train**, gli ultimi **2** → **video di test**

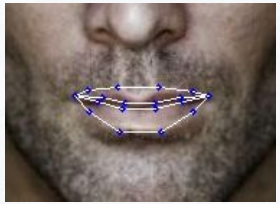


**43** individui totali

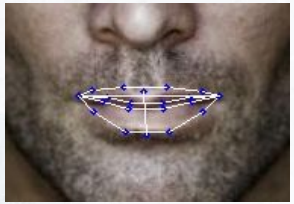
***Sperimentazioni future:***
*Quanto impatta avere più materiale per il training?*

# Due differenti approcci

- **Landmarks-based (Babele)**: si utilizza una parte dei landmarks forniti da MediaPipe relativi alle labbra, creando dei frozenset che ospitano tali landmarks (Lip Landmarks (LL), Lip Landmarks Dynamic (LLD), Lip Landmarks Full Mesh). Sfruttando tali frozenset si estraggono i landmarks da video:
  - Sono previste tre differenti configurazioni:



**Base**
Features: **20**
Frozenset: LL



**Dinamica**
Features: **22**
Frozenset: LLD

**Full-Mesh**
Features: **153**
Frozenset: LLFM

- **Video-based (Babele, VidTimit)**: si va ad utilizzare l'algoritmo **sparse-coding** per andare ad estrarre informazioni relative alle caratteristiche rilevanti nei video della labbra di diversi soggetti.

# Metriche

Definendo il termine “**metrica**” come il modo in cui si misura la distanza tra due punti, sono state utilizzate le seguenti metriche:

- Distanza euclidea:
  - Vengono considerate esclusivamente le coordinate x e y

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Distanza euclidea normalizzata:
  - Le distanza ottenute sono scalate in un intervallo [0,1]

- Distanza CityBlock3D:
  - Vengono considerate le coordinate x, y, z

$$d(P_1, P_2) = |(x_1 - x_2)| + |(y_1 - y_2)| + |(z_1 - z_2)|$$

Nella slide precedente viene fatta una distinzione tra metrica euclidea ed euclidea normalizzata, che potrebbe risultare poco chiara, siccome i landmarks estratti tramite MediaPipe sono già normalizzati.

Il motivo di questa distinzione è legato a come vengono, a livello di codice, calcolati i punti:

- Con la **metrica euclidea**, per le coordinate di **x** e **y** di ogni landmarks già normalizzato fornito da MediaPipe , si effettua una moltiplicazione di **x** e **y** rispettivamente per **larghezza** ed **altezza** dell'immagine, e successivamente si calcola la distanza fra i due punti.
- Con la metrica euclidea normalizzata, invece, la distanza viene calcolata direttamente sui punti che MediaPipe restituisce, e quindi la distanza risulta essere normalizzata.

La scelta, nel primo caso,  di alterare i punti forniti da MediaPipe mediante la **shape** dell'immagine è dovuta alla volontà di voler effettuare il **confronto tra due distanze che utilizzano la stessa metrica**, dove però una non è normalizzate e l'altra si.

*Slide approfondimento,*
*non presentare*

# Metodo di Sampling

Vogliamo prelevare **n** frame da un video. Quali frame prelevare?
**(t = numero totale di frame del video = 300)**

**Sampling sequenziale**: prelievo degli **ultimi n frame** dal video

$$FRAMES = \{F_i \mid t - n \leq i \leq t\}$$

**Sampling spaced**: prelievo di n frame dal video ad **intervalli regolari di t/n**, approssimato per difetto all'unità intera più vicina

$$FRAMES = \{F_{iz} \mid z = \lfloor t/n \rfloor,\ 0 \leq i \leq n - 1\}$$

# Biometria statica e biometria comportamentale

Le metodologie di sampling appena descritte, sono strettamente correlate l'una alla biometria statica, l'altra alla biometria comportamentale, vediamo il perché e cosa significano:

- **Biometria statica**: caratteristiche osservate in un determinato istante. Features prese dagli ultimi 20 frame del video (**sampling sequenziale**). Non si esamina il parlato.
- **Biometria comportamentale**: caratteristiche osservate nel tempo. Features prese da frame non contigui (**sampling spaced**). Si esamina il parlato.

**Intuizione alla base**: due o più soggetti differenti possono avere labbra delle stesse dimensioni, ma la probabilità che abbiamo anche lo stesso modo di muovere le labbra durante il parlato risulta essere bassa.


Biometria statica

Biometria comportamentale

# Biometria statica e biometria comportamentale

Ora che abbiamo introdotto cosa s'intende per "**sampling**", è possibile spiegare qual è l'effettiva **differenza** fra **biometria statica** e **biometria comportamentale** nel contesto delle labbra, che permette poi di andare a distinguere le sperimentazioni effettuate che vedremo nel dettaglio.

Quando parliamo di **biometria statica** intendiamo caratteristiche osservate in un determinato istante. Infatti, per le sperimentazioni che hanno questa direzione, quello che si fa è prendere le features dagli ultimi 20 frame del video e quindi si utilizza un **sampling sequenziale**. Prendendo tali frame, le features ottenute non permetteranno di esaminare quello che è il parlato di una persona.

Per **biometria comportamentale** intendiamo invece caratteristiche osservate nel tempo, in questo caso si utilizza un **sampling spaced** quindi le features verranno prese da frame non contigui, di conseguenza le features permetteranno di analizzare il parlato di un soggetto, che vedremo essere una caratteristica molto discriminante.

L'intuizione alla base che ci ha portato a fare questa distinzione, è che magari **due o più soggetti differenti possono avere labbra delle stesse dimensioni, ma la probabilità che abbiamo anche lo stesso modo di muovere le labbra durante il parlato risulta essere bassa**. Come vedremo in seguito il nostro modello andrà a confermare tale intuizione.

*Slide approfondimento,*
*non presentare*

# Sperimentazioni effettuate

È chiaro che il numero di sperimentazioni possibili è molto alto!
Le sperimentazioni effettuate sono state le seguenti:

***Base (20 segmenti)***

- Biometria statica
  - **20** frame per video
    - (sampling **sequenziale**, **ogni metrica**, **ogni kernel**)
- Biometria comportamentale
  - **75**, **100**, **150**, **300** frame per video
    - (sampling **spaced**, distanza **euclidea**, **default** kernel (laplaciano))

***Dinamica (22 segmenti)***

- Biometria comportamentale
  - **75**, **100**, **150**, **300** frame per video
    - (sampling **spaced**, distanza **euclidea**, **default** kernel (laplaciano))

***Full-Mesh (153 segmenti)***

- Biometria comportamentale
  - **75**, **100**, **150**, **300** frame per video
    - (sampling **spaced**, distanza **euclidea**, **default** kernel (laplaciano))

# Sperimentazioni biometria statica: perché?

Tali sperimentazioni hanno rappresentato il punto di partenza del nostro lavoro, ed avevano come obiettivo principale quello di capire, con **quale kernel** e con **quale metrica** la PNN conducesse a risultati migliori.

*Perché un semplice punto di partenza e non di arrivo?*

Il motivo è che risultava evidente che ciò che era alla base del lavoro, cioè dataset di video, non venissero del tutto sfruttati, ciò che veniva fatto era riproducibile utilizzando delle semplici immagini.

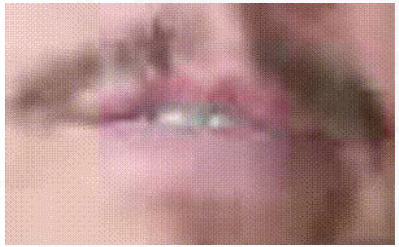
Ricordando che tali sperimentazioni prendevano le n_feats dagli ultimi n_frames, risulta evidente che non si utilizzassero **le informazioni legate al movimento delle labbra!**

*Cosa si perdeva?*

# Sperimentazioni biometria comportamentale: perché?

Tali sperimentazioni permettono di sfruttare a pieno i dataset utilizzati, permettendo l'acquisizione di informazioni semanticamente rilevanti.

**Perché semanticamente rilevanti?** Analizziamo questi tre soggetti



Soggetto **1_1_1_7**

Soggetto **1_1_2_9**

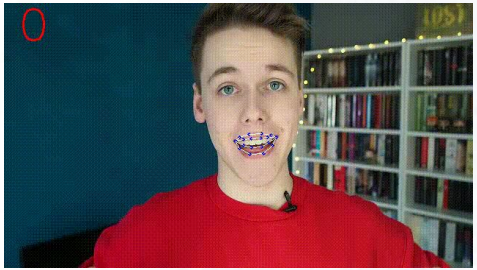
Soggetto **2_2_2_1**

Notate **differenze comportamentali**?

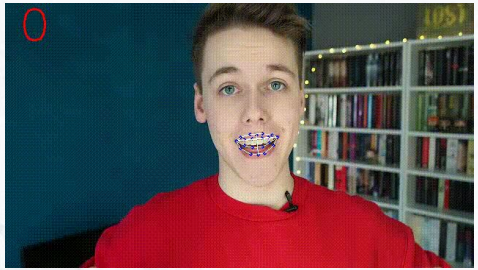
Indizi: velocità del parlato, ridimensionamento spaziale delle labbra, durata delle espressioni, etc..

# Sperimentazioni biometria comportamentale: un’ulteriore considerazione

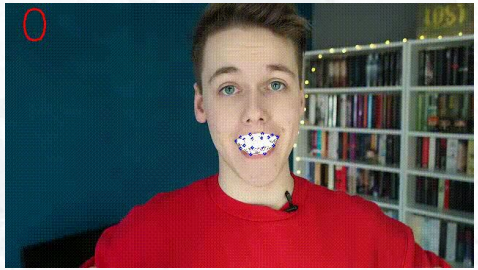
Come mostrato in precedenza per tutte le configurazioni, cioè “**Base**”, “**Dinamica**” e “**Full-Mesh**” sono state condotte delle sperimentazioni che tengono conto della **componente comportamentale** delle labbra durante il parlato.



Base



Dinamica



Full-Mesh

Ciò che sappiamo è che analizzare il **movimento è cruciale per la biometria comportamentale**, ciò che si vuol capire è se le informazioni relative alla **distanza tra gli estremi delle labbra** durante il movimento risultano essere particolarmente discriminanti.

Il confronto tra le configurazioni, in particolare tra quella “**Base**” e quella “**Dinamica**”, *poiché possiedono un numero simile di features e perché l’una possiede e l’altra no le distanze fra gli estremi*, ci permette di far emergere questo aspetto appena evidenziato.

# Biometria statica e comportamentale: strategia di memorizzazione

**Biometria statica**:


Definiamo **N_FEAT** come il numero di features (segmenti) prelevate per singolo frame, e **N_FRAMES** come il numero di frame esaminati dal video.

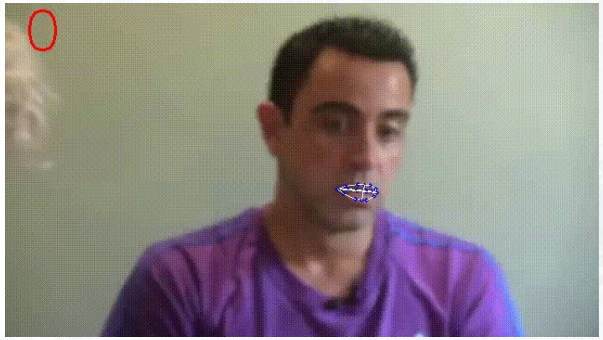
Con la strategia statica la riga i-esima contiene le n features prelevate dal singolo frame:

$$r_i = (f_1, f_2, \ldots, f_{n\_feat})$$
$$0 \leq i \leq n\_frames - 1$$

In questo caso il dataset avrà, **per il singolo video**, tanti samples Ri (**righe**) quanti sono i frame scelti nel video (n_frames) ed ogni sample Ri avrà il numero di features stabilite (n_feat)

# Biometria statica e comportamentale: strategia di memorizzazione (2)

**Biometria comportamentale**:




Nella strategia dinamica, invece, la riga i-esima corrisponde alle informazioni prelevate da un **intero video**, **concatenando** le informazioni prelevate dai singoli frames:

$$r = (r_0, r_1, r_2, \ldots, r_{n\_frames-1})$$

$$0 \leq i \leq n\_frames - 1$$

$$r_i = (f_1, f_2, \ldots, f_{n\_feat})$$

In questo caso il dataset avrà, **per il singolo video**, **un unico sample R**, che consiste nella concatenazione del tanti sample Ri quanti sono i frame analizzati dal video (n_frames). Ogni sample Ri avrà n_feat elementi.

# Datasets: alcuni dati

| Numero Frames | Configurazione | Train/Test | X SHAPE | Y SHAPE |
|---|---|---|---|---|
| 20 | Base | Train | 557x400 | 557x256 |
| 20 | Base | Test | 142x400 | 142x256 |
| 30 | Base | Train | 545x600 | 545x256 |
| 30 | Base | Test | 138x600 | 138x256 |
| 50 | Base | Train | 852x1000 | 852x256 |
| 50 | Base | Test | 217x1000 | 217x256 |
| 75 | Base | Train | 471x1500 | 471x256 |
| 75 | Base | Test | 118x1500 | 118x256 |
| 100 | Base | Train | 471x2000 | 471x256 |
| 100 | Base | Test | 119x2000 | 119x256 |
| 150 | Base | Train | 467x3000 | 467x256 |
| 150 | Base | Test | 118x3000 | 118x256 |
| 300 | Base | Train | 468x6000 | 468x256 |
| 300 | Base | Test | 118x6000 | 118x256 |

*Slide approfondimento,*
*non presentare*

# Datasets: alcuni dati (2)

| Numero Frames | Configurazione | Train/Test | X SHAPE | Y SHAPE |
|---|---|---|---|---|
| 20 | Dinamica | Train | 557x440 | 557x256 |
| 20 | Dinamica | Test | 142x440 | 142x256 |
| 30 | Dinamica | Train | 545x660 | 545x256 |
| 30 | Dinamica | Test | 138x660 | 138x256 |
| 50 | Dinamica | Train | 852x1100 | 852x256 |
| 50 | Dinamica | Test | 217x1100 | 217x256 |
| 75 | Dinamica | Train | 471x1650 | 471x256 |
| 75 | Dinamica | Test | 118x1650 | 118x256 |
| 100 | Dinamica | Train | 471x2200 | 471x256 |
| 100 | Dinamica | Test | 119x2200 | 119x256 |
| 150 | Dinamica | Train | 467x3300 | 467x256 |
| 150 | Dinamica | Test | 118x3300 | 118x256 |
| 300 | Dinamica | Train | 468x6600 | 468x256 |
| 300 | Dinamica | Test | 118x6600 | 118x256 |

*Slide approfondimento,*
*non presentare*

# Datasets: alcuni dati (3)

| Numero Frames | Configurazione | Train/Test | X SHAPE | Y SHAPE |
|---|---|---|---|---|
| 20 | Dinamica | Train | 557x3060 | 557x256 |
| 20 | Dinamica | Test | 142x3060 | 142x256 |
| 30 | Dinamica | Train | 545x4590 | 545x256 |
| 30 | Dinamica | Test | 138x4590 | 138x256 |
| 50 | Dinamica | Train | 852x7650 | 852x256 |
| 50 | Dinamica | Test | 217x7650 | 217x256 |
| 75 | Dinamica | Train | 471x11475 | 471x256 |
| 75 | Dinamica | Test | 118x11475 | 118x256 |
| 100 | Dinamica | Train | 471x15300 | 471x256 |
| 100 | Dinamica | Test | 119x15300 | 119x256 |
| 150 | Dinamica | Train | 467x22950 | 467x256 |
| 150 | Dinamica | Test | 118x22950 | 118x256 |
| 300 | Dinamica | Train | 468x45900 | 468x256 |
| 300 | Dinamica | Test | 118x45900 | 118x256 |

*Slide approfondimento,*
*non presentare*

# RISULTATI

*I risultati che ora verranno mostrati sono tutti relativi all'**identificazione** fatta su **256 distinti soggetti***

# Risultati sperimentazioni biometria statica

Riassumendo, come esplicitato nei lucidi precedenti, l'unica configurazione per la quale si è considerata la biometria statica è quella "Base":

- Tutte le sperimentazioni in tale senso:
  - Hanno come **costanti**:
    - **Numero di frame per video**: 20
    - **Sampling**: sequenziale
  - Hanno come **variabili**:
    - **Metrica:** (Euclidea, Euclidea Normalizzata, CityBlock3D)
    - **Kernel:** (RBF, Laplaciano, Triangolare, Epanechnikov, Uniforme)

La possibilità di diversificare le sperimentazioni con biometria statica mediante 5 kernel e 3 differenti metriche, con una singola configurazione ci porta ad avere 15 differenti risultati

| Kernel | Metrica | Accuracy % | Precision % | Recall % |
|---|---|---|---|---|
| **RBF** | Euclidea | 14,84 | 65,44 | 15,50 |
| **RBF** | Euclidea Norm. | 6,89 | 80,52 | 6,89 |
| **RBF** | CityBlock3D | 10,79 | 67,22 | 11,49 |
| **Laplaciano** | **Euclidea** | **15,71** | **61,89** | **16,37** |
| **Laplaciano** | Euclidea Norm. | 7,47 | 77,10 | 7,47 |
| **Laplaciano** | CityBlock3D | 13,10 | 65,07 | 13,78 |
| **Uniforme** | Euclidea | 4,42 | 72,56 | 4,80 |
| **Uniforme** | **Euclidea Norm.** | **0,39** | **99,6** | **0,39** |
| **Uniforme** | **CityBlock3D** | **0,39** | **99,6** | **0,39** |
| **Epanechnikov** | Euclidea | 14,86 | 61,28 | 15,52 |
| **Epanechnikov** | Euclidea Norm. | 6,89 | 80,52 | 6,89 |
| **Epanechnikov** | CityBlock3D | 10,79 | 67,22 | 11,49 |
| **Triangolare** | Euclidea | 15,84 | 56,42 | 16,50 |
| **Triangolare** | Euclidea Norm. | 7,47 | 77,16 | 7,47 |
| **Triangolare** | CityBlock3D | 13,10 | 65,07 | 13,78 |

# Risultati sperimentazioni biometria statica:considerazioni

1. Si può osservare una significativa differenza tra i **kernel** che attribuiscono un peso in base alla similarità dell'input rispetto all'unità di base e il **kernel uniforme**, che invece assegna un peso costante all'interno di un determinato intervallo attorno all'unità di base.
2. Approfondendo il risultato migliore ottenuto nelle sperimentazioni con la biometria statica, l'effetto di smorzamento del **kernel laplaciano** potrebbe risultare vantaggioso nel raggiungere risultati superiori quando si lavora con dati rumorosi. È possibile che la presenza di elementi di rumore abbia favorito l'efficacia del kernel laplaciano rispetto agli altri metodi.

| Kernel | Metrica | Accuracy % | Precision % | Recall % |
|---|---|---|---|---|
| RBF | Euclidea | 14,84 | 65,44 | 15,50 |
| RBF | Euclidea Norm. | 6,89 | 80,52 | 6,89 |
| RBF | CityBlock3D | 10,79 | 67,22 | 11,49 |
| Laplaciano | Euclidea | 15,71 | 61,89 | 16,37 |
| Laplaciano | Euclidea Norm. | 7,47 | 77,10 | 7,47 |
| Laplaciano | CityBlock3D | 13,10 | 65,07 | 13,78 |
| Uniforme | Euclidea | 4,42 | 72,56 | 4,80 |
| Uniforme | Euclidea Norm. | 0,39 | 99,6 | 0,39 |
| Uniforme | CityBlock3D | 0,39 | 99,6 | 0,39 |
| Epanechnikov | Euclidea | 14,86 | 61,28 | 15,52 |
| Epanechnikov | Euclidea Norm. | 6,89 | 80,52 | 6,89 |
| Epanechnikov | CityBlock3D | 10,79 | 67,22 | 11,49 |
| Triangolare | Euclidea | 15,84 | 56,42 | 16,50 |
| Triangolare | Euclidea Norm. | 7,47 | 77,16 | 7,47 |
| Triangolare | CityBlock3D | 13,10 | 65,07 | 13,78 |

# Risultati sperimentazione biometria comportamentale

Tutte le configurazioni prevedono invece la considerazione della biometria comportamentale, dove per le diverse sperimentazioni **le costanti** sono:

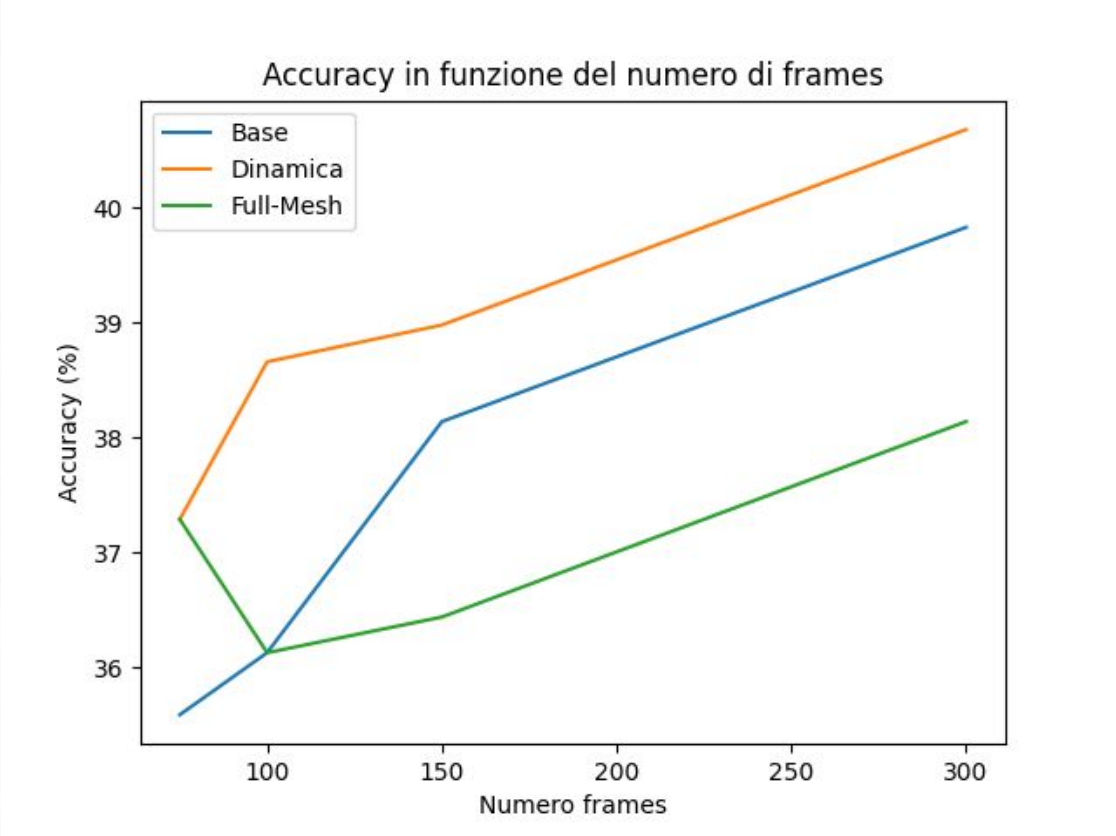
- **Kernel** (Laplaciano)
- **Metrica** (Euclidea)

Le variabili sono la **configurazione** (Base, Dinamica, Full-mesh) e il **numero di frame per video** (nel caso di 300 frame, si prendono tutti i frame del video)

La possibilità di diversificare le sperimentazioni con biometria dinamica con 3 differenti configurazioni e 4 differenti numeri di frame prelevati per video, ci porta ad avere 12 risultati differenti. Il **miglior risultato** lo si raggiunge con configurazione **Dinamica** e **300 frame**.

| Configurazione | Numero frames | Accuracy % | Precision % | Recall % |
|---|---|---|---|---|
| **Base** | **75** | 35,59 | 78,09 | 36,66 |
| **Base** | **100** | **36,13** | **77,99** | **37,70** |
| **Base** | **150** | 38,13 | 75,34 | 40,65 |
| **Base** | **300** | 39,83 | 75,60 | 42,27 |
| **Dinamica** | **75** | 37,28 | 77,25 | 39,83 |
| **Dinamica** | **100** | 38,65 | 77,23 | 41,12 |
| **Dinamica** | **150** | 38,98 | 77,16 | 41,46 |
| **Dinamica** | **300** | **40,67** | **75,36** | **43,08** |
| **Full-mesh** | **75** | 37,28 | 75,12 | 39,83 |
| **Full-mesh** | **100** | 36,13 | 72,86 | 38,70 |
| **Full-mesh** | **150** | 36,44 | 73,98 | 39,02 |
| **Full-mesh** | **300** | 38,13 | 74,34 | 40,65 |

# Risultati sperimentazione biometria comportamentale



# Risultati sperimentazione biometria comportamentale: considerazioni

1. I risultati delle sperimentazioni con biometria comportamentale, sono in media molto migliori rispetto a quelli ottenuti mediante le sperimentazioni con biometria statica. Una possibile interpretazione è che, nel momento in cui si vuole effettuare l'identificazione di un soggetto, mediante labbra, **il parlato risulta essere molto più discriminante rispetto alla forma delle stesse**.
2. La configurazione “Dinamica” porta ad un risultato di poco migliore rispetto alla configurazione “Base”, **le informazioni relative alle distanze fra gli estremi delle labbra risultano essere positive anche se non fondamentali, per il nostro modello.**
3. L’ultima considerazione, è relativa al numero di frame. L’**accuracy più elevata** la si ottiene con la “Dinamica” e 300 frames, ma in realtà anche per le altre configurazioni il picco di prestazioni si ha con 300 frames, cioè prendendo ogni singolo frame del video. È possibile pensare, ricollegandoci alla prima considerazione, che il motivo sia legato al fatto che andando a prendere tutti i possibili frame, il modello vada ad **addestrarsi sul parlato**, **ottenendo così un incremento di efficacia.**

| Configurazione | Numero frames | Accuracy % | Precision % | Recall % |
|---|---|---|---|---|
| **Base** | **75** | 35,59 | 78,09 | 36,66 |
| **Base** | **100** | **36,13** | **77,99** | **37,70** |
| **Base** | **150** | 38,13 | 75,34 | 40,65 |
| **Base** | **300** | 39,83 | 75,60 | 42,27 |
| **Dinamica** | **75** | 37,28 | 77,25 | 39,83 |
| **Dinamica** | **100** | 38,65 | 77,23 | 41,12 |
| **Dinamica** | **150** | 38,98 | 77,16 | 41,46 |
| **Dinamica** | **300** | **40,67** | **75,36** | **43,08** |
| **Full-mesh** | **75** | 37,28 | 75,12 | 39,83 |
| **Full-mesh** | **100** | 36,13 | 72,86 | 38,70 |
| **Full-mesh** | **150** | 36,44 | 73,98 | 39,02 |
| **Full-mesh** | **300** | 38,13 | 74,34 | 40,65 |

# Sparse Coding: una strategia differente

**Problema:** identificazione dei soggetti

**Input**: video (delle loro labbra)

Nell'approccio geometrico sono state ricercate delle tecniche che permettessero di utilizzare al meglio i **dati geometrici** prelevati dal video.

**Approccio video-based**: uso di tecniche **sparse coding** (tecnica di rappresentazione dei dati)

Trova una rappresentazione "**sparsa**" dei dati, catturando le **caratteristiche salienti** dei dati

$$v = (f_1, f_2, f_3, f_4, \ldots, f_s)$$

$$f_i \in \mathbb{R}_{\mathbb{0}}^{+}$$

$$s = video\_height \times video\_width$$

# Sparse Coding: un strategia differente (2)

**Approccio landmarks-based:** misurare distanze tra coppie strategiche di landmarks


MediaPipe


$$R_i = (f_1, f_2, \ldots, f_{n\_feat}) \quad 0 \leq i \leq n\_frames - 1$$

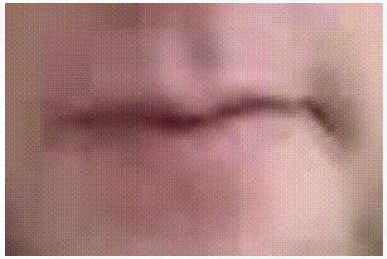
$$R = (R_0, R_1, \ldots, R_{n\_frames-1})$$

L'ipotesi alla base di questo approccio è che la **preziosità biometrica delle distanze** tra segmenti specifici dell'aria labiale possa **essere utile a distinguere individui**.

A seconda della visuale con la quale ci si pone (labbra come *biometria statica o comportamentale*) i dati entrano nel modello di AI in forme differenti, ma sono sempre e comunque **distanze**.

# Sparse Coding: un strategia differente (3)

**Approccio video-based:** convertire video delle labbra in vettori che “riassumono” il video



Sparse coding

$$v = (f_1, f_2, f_3, f_4, \ldots, f_s)$$

$$f_i \in \mathbb{R}_{\mathbb{0}}^{+}$$

$$s = video\_height \times video\_width$$

L’ipotesi alla base di questo approccio è diversa: la preziosità biometrica non risiede soltanto nelle distanze ma in **tutte le caratteristiche fenotipiche delle labbra.**

La rappresentazione vettoriale risultante cattura le **caratteristiche principali del video in un'unica struttura dati**, che può essere utilizzata per scopi come l’identificazione di individui.

# Sparse Coding: un strategia differente (4)

$$v = (f_1, f_2, f_3, f_4, \ldots, f_s)$$

$$f_i \in \mathbb{R}_{\mathbb{0}}^{+}$$

*Slide approfondimento, non presentare*

L'algoritmo usato per le sperimentazioni crea un numero di **dizionari** pari a **n_components**, parametro modificabile a piacere. Nel nostro caso n_components = 1.

La dimensione del dizionario è

$$s = video\_height \times video\_width$$

Dato che i frame dei video delle labbra sono di dimensione **300x200 pixel**, il dizionario conterrà 300x200 elementi cioè **60.000 elementi**.

Il numero s **non è condizionato dal framerate** del video perchè ad ogni frame, l'algoritmo esegue una "**partial-fit**" **sul dizionario**, modificandolo sulla base del frame correntemente prelevato ed analizzato.

# Sparse Coding: cosa c’è da sapere

Nel nostro caso, SC è utilizzato per **convertire video in un'informazione algebrica** (un vettore).

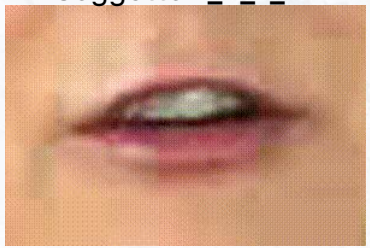
L’obiettivo è quello di usare tali informazioni algebriche per **addestrare e mettere alla prova il nostro modello di AI**.

In input all’algoritmo verranno forniti i video delle labbra.

Attenzione: i video trasportano informazioni riguardanti le labbra, **ma non solo**! *Tecnologia del dispositivo di acquisizione, rumore, illuminazione, posa dell’individuo, etc.*

Soggetto **7_2_1_17**

Soggetto **7_2_2_7**



Soggetto **2_2_2_21**

Soggetto **2_2_2_1**

# Approccio video-based: risultati

L'algoritmo permette di **modificare diversi parametri** (batch_size, alpha, n_iter, n_components, etc...).

Per entrambe le sperimentazioni, l'algoritmo è stato impostato in maniera tale da convertire ogni video in un vettore di **s = 60.000 features**.

$$v = (f_1, f_2, f_3, \ldots, f_{60000})$$

$$r = (v, label)$$

Ogni video verrà rappresentato nel dataset come **r = (v, label)** dove v è il vettore risultato dell'applicazione dell'algoritmo al video.

| Dataset | Accuracy % | Precision % | Recall % |
|---|---|---|---|
| BABELE | 87,5% | 94,01% | 87,5% |
| VidTimit | 96,51% | 98,06% | 96,51% |

Risultati molto promettenti... ma non possiamo esserne sicuri!

Perché VidTimit dà risultati migliori di BABELE?

# Approccio video-based: risultati (2)

**BABELE**:

- **Fotocamere** diverse, così come **illuminazione** e **posa** diverse (PIE Variations).
- I video di train e test sono **segmenti di uno stesso video originario**, quindi condividono tali caratteristiche.

| Dataset | Accuracy | Precision | Recall |
|---|---|---|---|
| **BABELE** | 87,5% | 94,01% | 87,5% |
| **VidTimit** | 96,51% | 98,06% | 96,51% |

Per fare un confronto e tirare delle somme, è stato necessario sperimentare anche **VidTimit**!

**VidTimit**:

- Video messi a fattor comune: **stessa** fonte, posa ed illuminazione
- **Sessioni** di registrazione **distinte** per ogni file video

Probabilmente il **vero riconoscimento** delle labbra è avvenuto in VidTimit!

La **natura "rumorosa"** dei video in BABELE può aver influenzato le performance

# Identificazione vs. Verifica

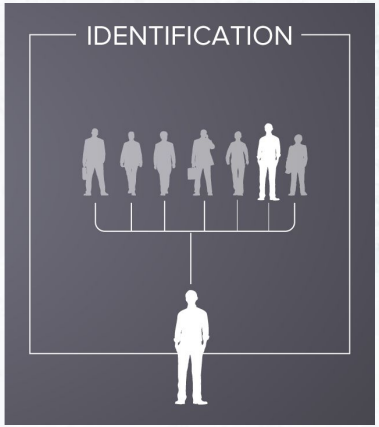
Vogliamo ora affrontare un nuovo problema, cioè **verificare gli utenti**.

Facciamo chiarezza:

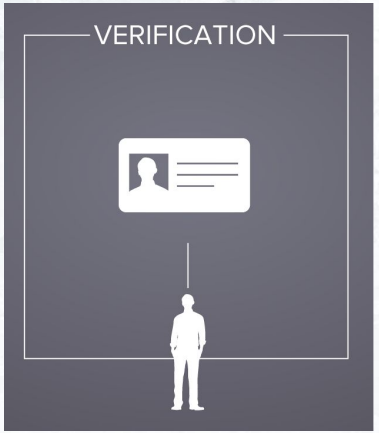
- **Identificazione**: assegnare un'identità univoca a un individuo
- **Verifica**: conferma dell'autenticità dell'identità dichiarata

Le sperimentazioni fino ad ora spaziano nel problema dell’identificazione.

Come **adattiamo i dati** per la verifica?




Assegnazione di un'identità unica




Conferma dell'identità

# Verificare gli utenti: i dati

Serve **adattare i dati** per mettere alla prova il modello sul problema della verifica.

È un lavoro pressoché immediato!

Nelle sperimentazioni precedenti:

$$r = (f_1, f_2, \ldots, f_n, label)$$

Dove ***n*** **dipende dalla configurazione** utilizzata (Full-Mesh, Dynamic, Base) mentre **label indica un codice identificativo del soggetto**.

Per il problema della verifica, sia **p** il **codice di un fissato soggetto,** la conversione:

$$r = (f_1, f_2, \ldots, f_n, label)$$

$$\downarrow$$

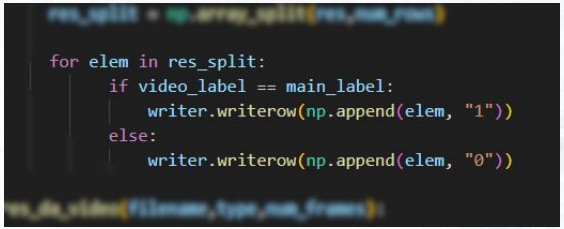
$$\begin{cases} (f_1, f_2, \ldots, f_n, 1) & \text{if } label = p \\ (f_1, f_2, \ldots, f_n, 0) & \text{if } label \neq p \end{cases}$$

crea un **dataset di verifica per p**

N.B.: questa conversione porta a dataset con **classi di taglie sbilanciate**, va ribilanciato!

# Verificare gli utenti: i dati (2)

La conversione descritta permette di avere un **dataset di verifica** per il **soggetto p**.

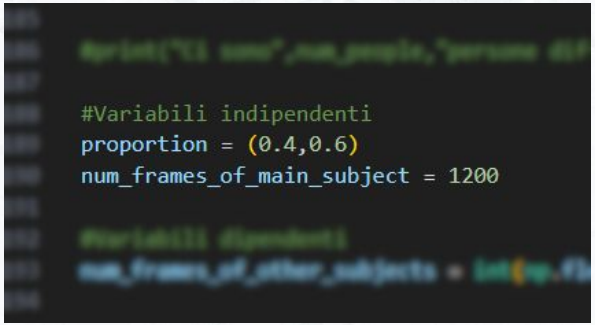


```
for elem in res_split:
        if video_label == main_label:
            writer.writerow(np.append(elem, "1"))
        else:
            writer.writerow(np.append(elem, "0"))
```

La conversione porta ad un dataset sbilanciato, in cui le righe di “p” occupano solo 1/256 ~ 0,3% del dataset...

Sono state effettuate **operazioni parametriche di bilanciamento dei datasets.**

Nei dataset per la sperimentazione, il soggetto “p” occupa il 40% del dataset, il restante 60% è occupato da campioni degli altri soggetti.



```
#Variabili indipendenti
proportion = (0.4,0.6)
num_frames_of_main_subject = 1200
```

Per le sperimentazioni è stata fissata come metrica il calcolo delle distanze euclidee, come kernel il Laplaciano.

# Verificare gli utenti: creazione del dataset

L'obiettivo è avere un dataset bilanciato e pronto ad essere utilizzato dal modello di intelligenza artificiale

I dataset ottenuti dalla conversione naive sono **fortemente sbilanciati**

Il **metodo programmatico di bilanciamento** permette di specificare la **percentuale di bilanciamento** (**p**, **1-p**) del dataset così come il **numero C di campioni** del soggetto

*Nel nostro caso:*

il soggetto p sarà presente, con 1200 campioni, al 40% nel dataset ($p = 0.4$, $C = 1200$).

Il dataset riempirà il restante 60% ($1-p = 0.6$) del dataset con 1800 campioni ($s = 0.6xP/0.4 = 1800$) degli altri 255 individui

Per ogni individuo, il metodo preleverà $\lfloor s/\#subjects \rfloor = \lfloor 1800/255 \rfloor = 7$ campioni ad individuo.

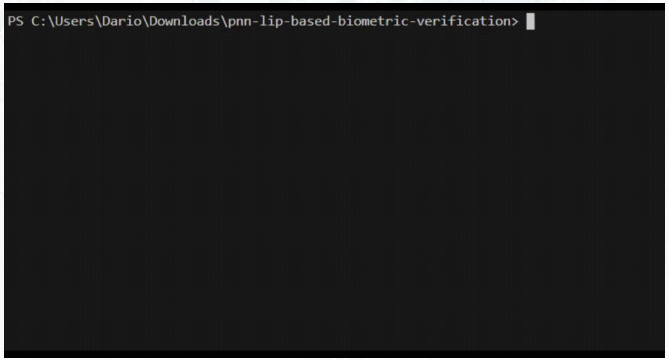
*Slide approfondimento,*
*non presentare*

# Verificare gli utenti: risultati

Configurazione utilizzata:

- **Metrica**: distanze euclidee
- **Kernel**: Laplacian
- **Sampling**: spaced
- **Configurazione**: Base
- **Strategia memorizzazione**: Statica

L'accuracy del modello, nei vari test su diversi soggetti, si attesta in media attorno al **76,05%**, con una precisione del circa **88,02%** e una recall attorno al **50,41%.**




Creazione del dataset per un soggetto, predizione e displaying dei risultati

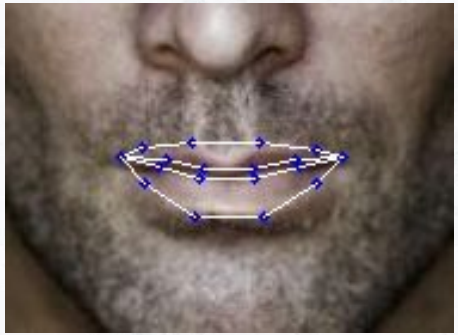
# Altre sperimentazioni (1)

Uno degli obiettivi del progetto è stato analizzare **quanti e quali componenti** geometriche meglio contribuiscano ad ottenere **risultati soddisfacenti.**

**Valori dei parametri della sperimentazione**

- **Kernel** (Laplacian)
- **Metrica** (Euclidea)
- **Configurazione** (base = 20 segmenti)
- **Sampling**: spaced
- **Frame**: 20
- **Dataset**: BABELE

L'obiettivo è quello di ridurre la dimensionalità del feature-set considerando solo le **features più "ferme"** e cercare di **identificare gli utenti**



FrozenSet LL (20 segmenti)
usato per la sperimentazione

Per features "ferme" intendiamo quelle features (segmenti) che, **in media** tra gli individui, hanno la **varianza** più **bassa.**

# Altre sperimentazioni (2)

Di ogni individuo, consideriamo soltanto le righe nelle quali è coinvolto nel dataset per intero:

$$DS = TRAIN \cup TEST$$

$$r_j = \{(f_1, \ldots, f_{20}, label) \in DS \mid label = j\}$$

Per ogni individuo, per ogni features, calcoliamo la varianza di tale feature:

$$Var_j(i) = \text{varianza della feature i nell'individuo j}$$

Calcoliamo, per ogni feature, il suo punteggio di instabilità, da 0 (molto stabile) a 1 (molto instabile):

$$Score(i) = \frac{\sum_{j=0}^{255} Var_j(i)}{256}$$

$$Scores = (Score(i) \mid 0 \leq i \leq 20)$$

Definito scores, creiamo un nuovo dataset prendendo come feature soltanto le “t” feature con score più basso e, ovviamente, la label.

$$TRAIN_t = \pi_{t,label}(TRAIN)$$

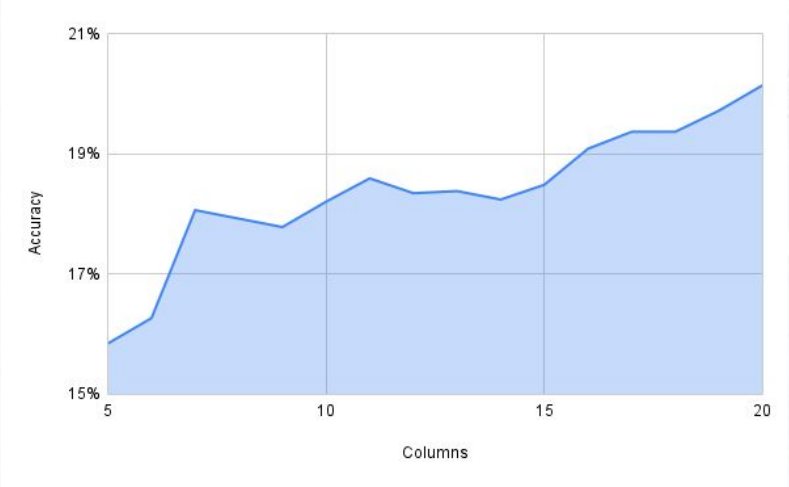
$$TEST_t = \pi_{t,label}(TEST)$$

# Altre sperimentazioni (3)

In questo grafico è mostrata l'accuracy in funzione del numero di colonne selezionate (da 5 a 20).

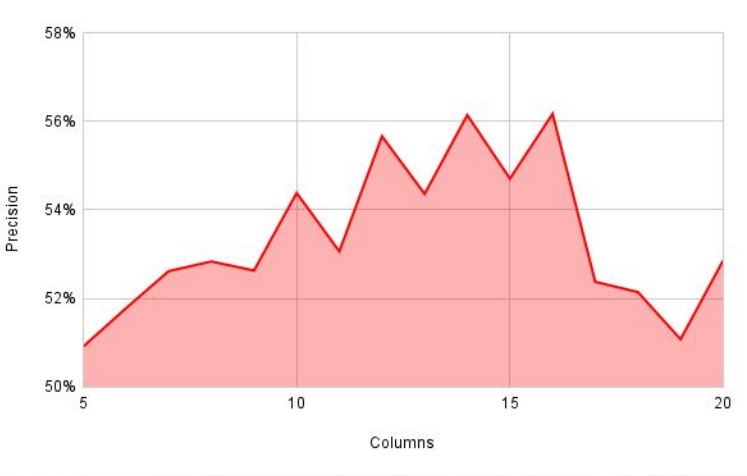
L'**accuracy migliore** (20,14%) è associata al prelievo di **tutte le colonne** (20 colonne).

Leggero **massimo locale** in corrispondenza del prelievo di 7 colonne (18,06%).

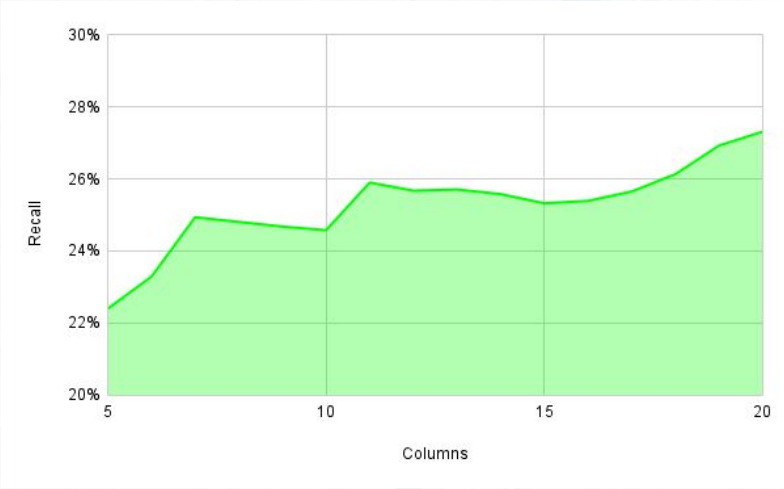



Accuratezza in funzione del numero di colonne

# Altre sperimentazioni (4)




Precisione in funzione del numero di colonne



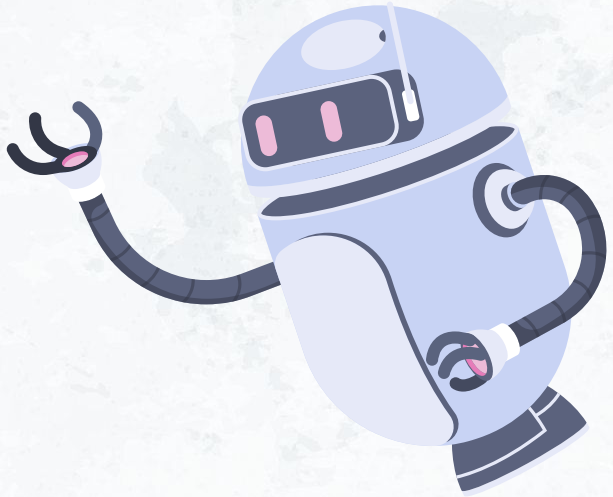

Recall in funzione del numero di colonne

# Considerazioni finali

Il progetto ha avuto come scopo quello di determinare l’efficacia biometrica dell’area labiale.

La biometria labiale può essere un elemento sufficiente di supporto al problema dell’identificazione e della verifica, sia in un sistema multi-biometrico che come singola biometria.

Le sperimentazioni effettuate possono essere riprodotte su *nuovi dataset, ad alta entropia, esenti da rumo*re e/o altri eventi.



Questa presentazione è stata creata senza l’uso di AI

Il codice è disponibile su **GitHub**

**Grazie dell’attenzione!**