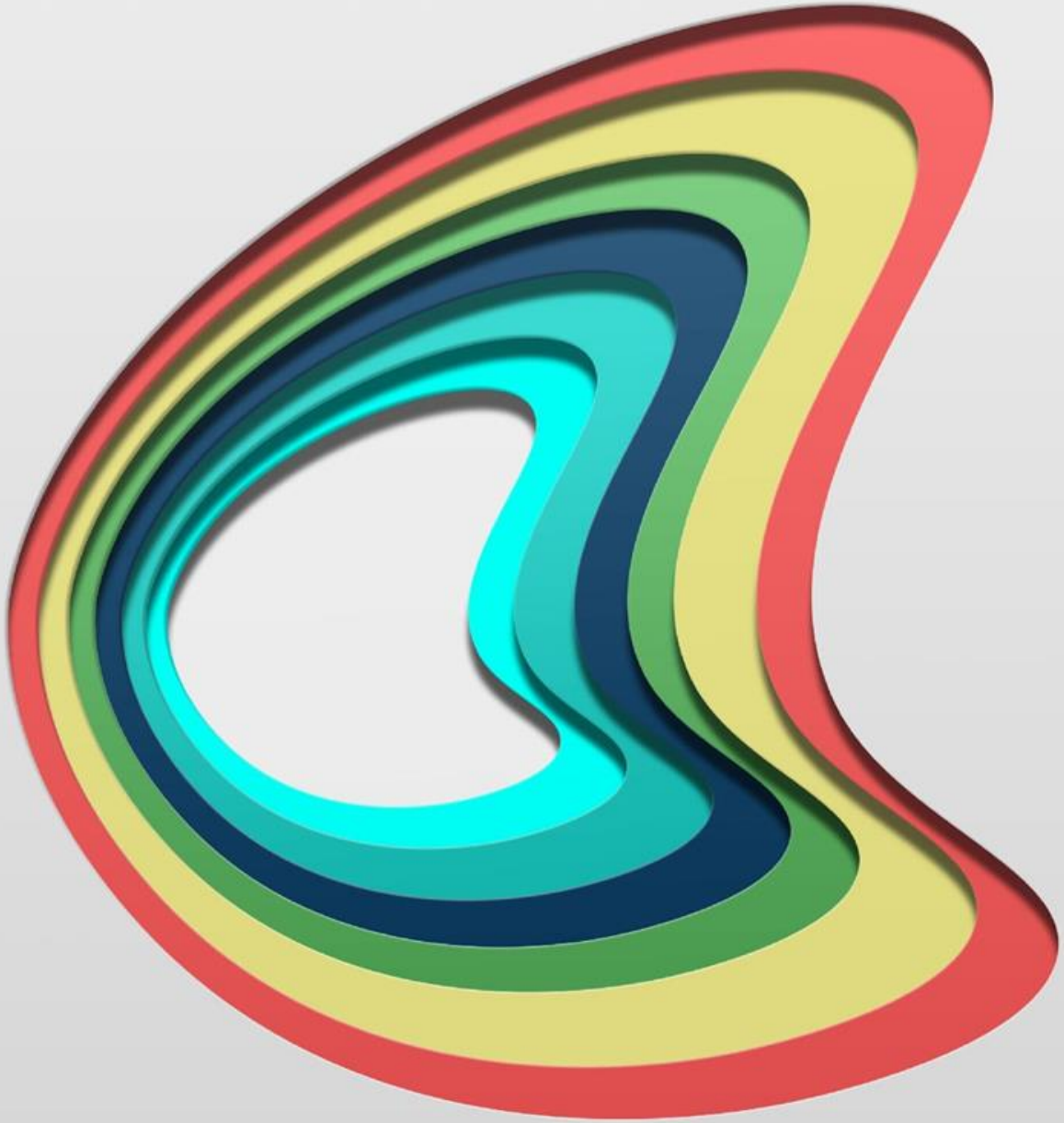


Progetto di Digital and Image Management

Febbraio 2022

Confalonieri Riccardo, 830404

Mariani Ginevra, 829506

Mora Lorenzo, 825393

Obiettivi del progetto

- **Audio Recognition**, lo scopo è quello di implementare dei modelli di ML e DL che siano in grado di predire correttamente la voce dello speaker;
- **Face Recognition**, consiste nel definire dei modelli in grado di identificare il volto dei componenti del gruppo;
- **Image Retrieval**, data un'immagine di query lo scopo è quello di trovare le 10 immagini, a partire da un dataset esterno, più similia quella data;

Audio Recognition - Dataset

- Il dataset, realizzato ad hoc, inizialmente conteneva 300 audio della durata di 5 secondi registrati leggendo paragrafi diversi tratti da due libri;
- Gli audio sono stati successivamente tagliati in segmenti da due secondi e *aumentati* attraverso due approcci (*pitch shifting* e *time stretching*);
- Sono stati testati molteplici *feature extractors* tra cui MFCC, ZCR, RMSE e Spectral Centroids;
- Spesso una sola features potrebbe non bastare, per cui sono state testate delle combinazioni dei feature extractors.

Support Vector Machine

Il primi approccio sperimentato per la classificazione è SVM con kernel radiale:

- L'algoritmo è stato addestrato con le features estratte con MFCC;
- L'addestramento è stato eseguito sia sui dati segmentati sia sui dati con aumentati considerando anche la normalizzazione degli stessi per capire quali fossero i dati più adatti;
- La configurazione dei dati migliore dei dati è quella data dai dati aumentati, ma non normalizzati;
- Sono state testate diversi configurazioni dei parametri *C* e *gamma*, concludendo che quella migliore fosse data da *C* = 18 e *gamma* = 10^{-9}.

Random Forest

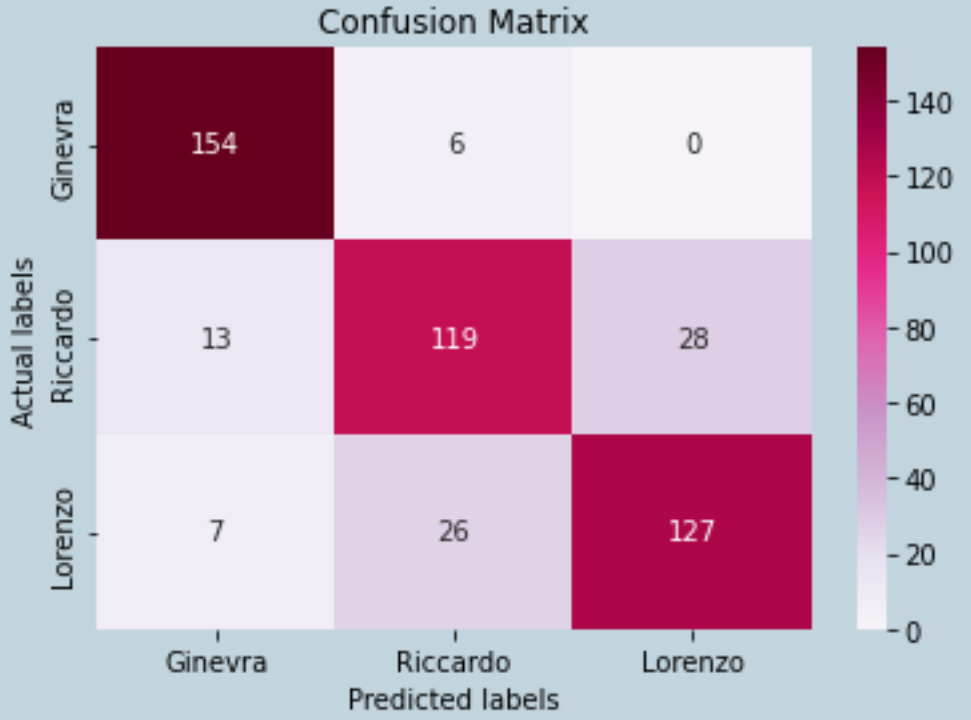
Il secondo approccio sperimentato è stato Random Forest, un algoritmo molto usato per classificazione.

- RF è stato addestrato con le features estratte con la funzione MFCC;
- Il numero ottimale di stimatori è risultato essere 100 con profondità 12;
- Gli audio utilizzati sono quelli segmentati e aumentati senza normalizzazione;
- Il tempo di esecuzione è nettamente superiore rispetto a SVM;

RF risulta incline ad un adattamento eccessivo, infatti con audio nuovi l'algoritmo ha commesso diversi errori. Per tale ragione non è usato nella demo live.

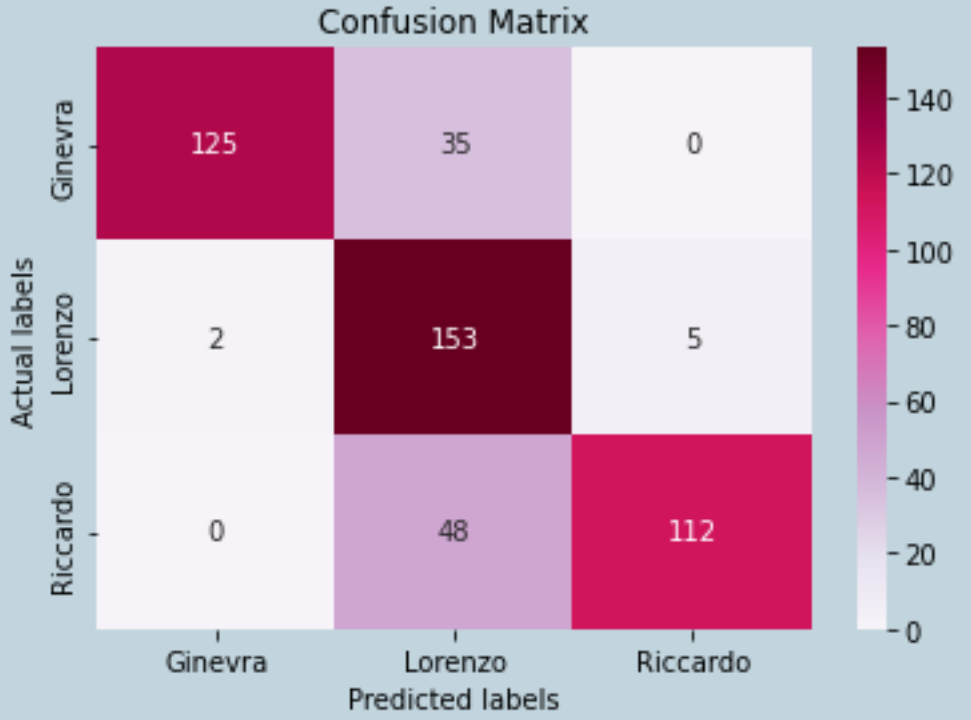
Risultati a confronto

SVM



Accuratezza sul test set: 0.83

RANDOM FOREST

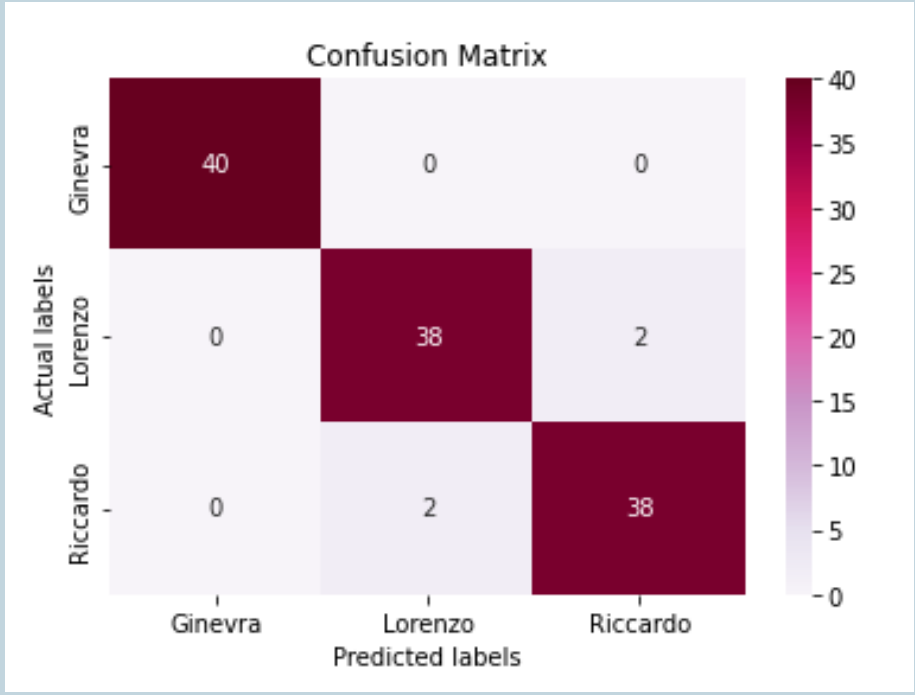


Accuratezza sul test set: 0.81

Rete Neurale Convoluzionale

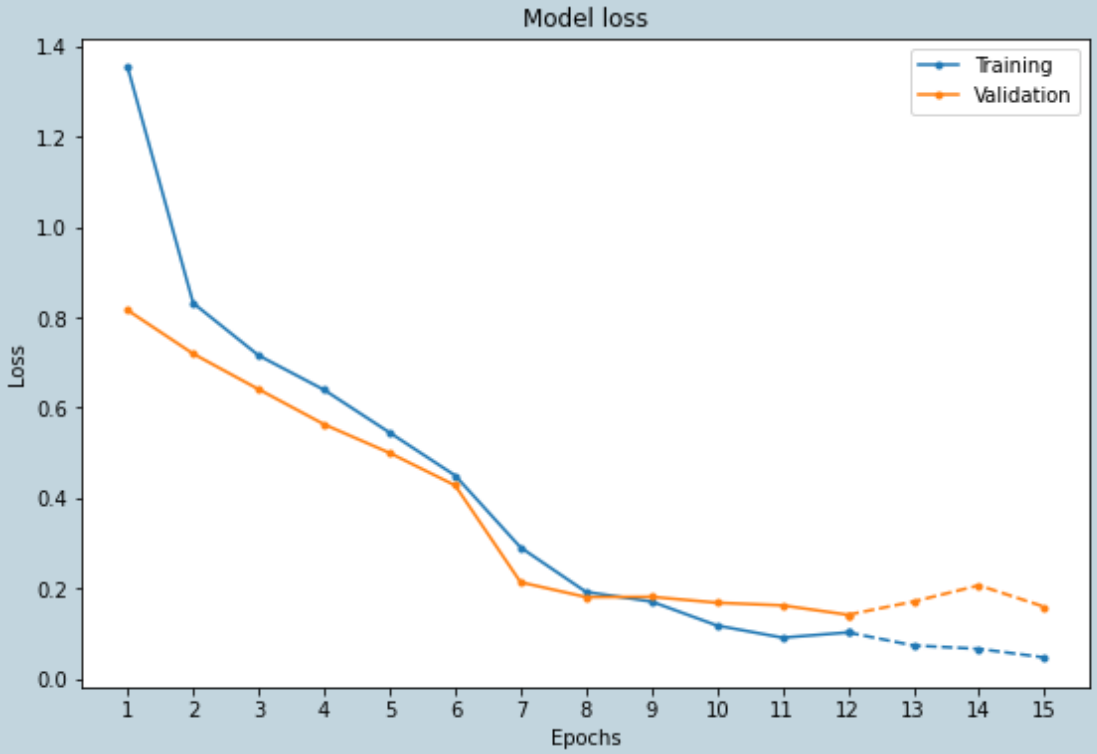
La rete prende in input le features generate con MFCC sui dati segmentati ed è costituita dai seguenti layer:

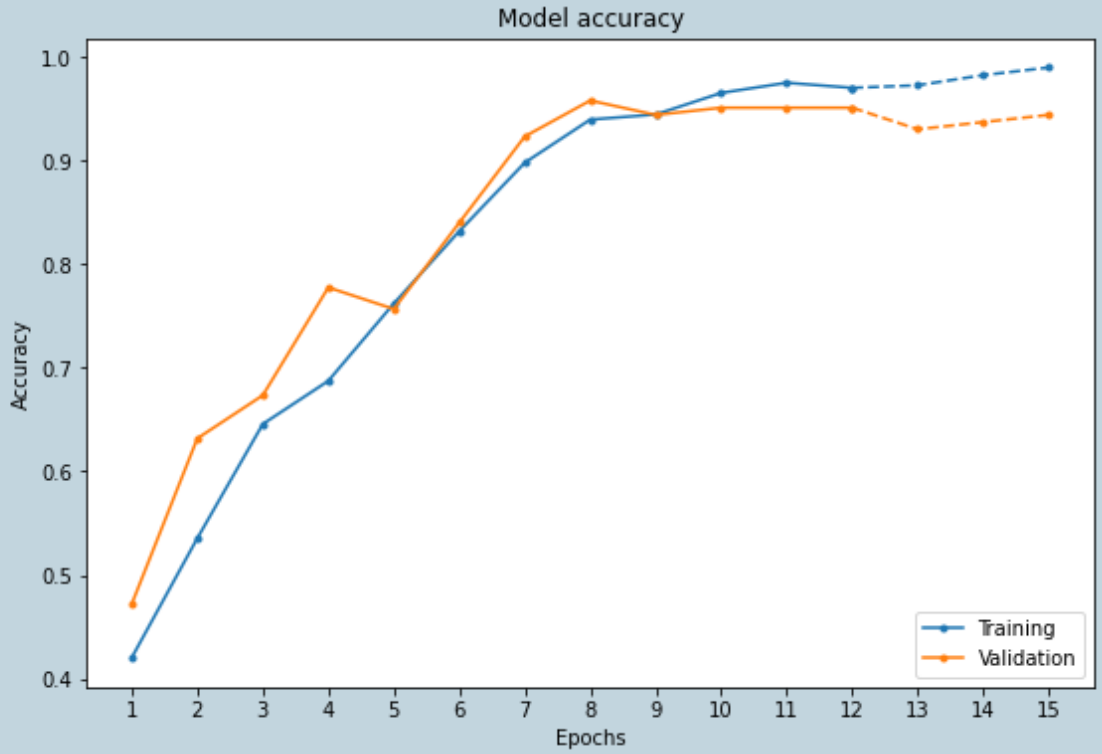
- Input((None, 13, 173, 1)
- Conv2D(32, (3,3), 'relu')
- Conv2D(32, (3,3), 'relu')
- Dropout(0.3)
- MaxPooling2D(pool_size = (2,2))
- Flatten()
- Dense(16, activation='relu')
- Dense(16, activation='relu')
- Dense(3, activation='softmax')



Accuratezza sul test set: 0.97

Training della rete

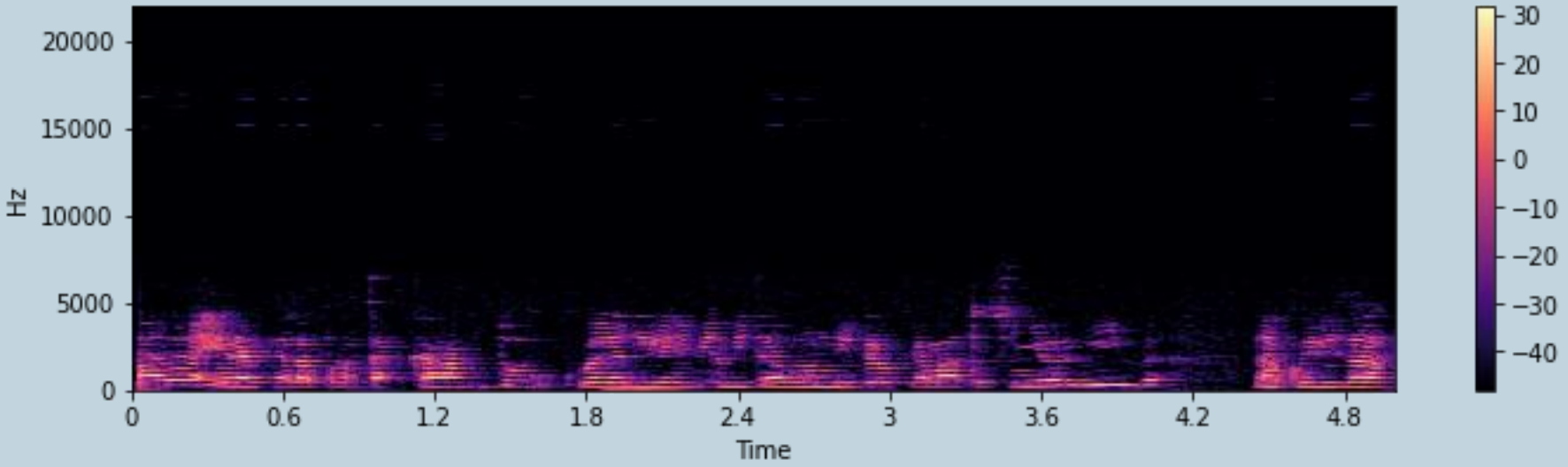




Osservando il grafico della loss il modello sembra aver sofferto nella prima fase di underfitting e di overfitting nelle ultime epoche, globalmente però l'addestramento è andato abbastanza bene.

Spectrogram Classification

Uno **spettrogramma** è una rappresentazione visiva di come lo spettro delle frequenze di un segnale audio vari nel tempo.

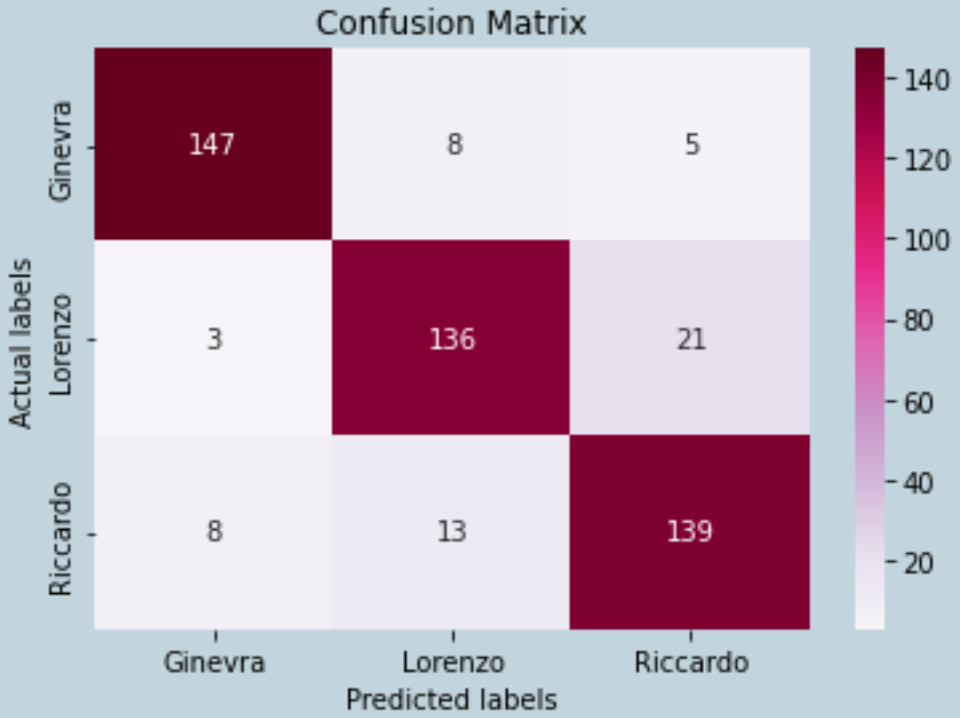


Una volta generati gli spettrogrammi per gli audio tagliati e *aumentati* sono stati convertiti in scala di grigi e sono stati passati alla rete convoluzionale *from scratch* che abbiamo implementato.

La rete implementata

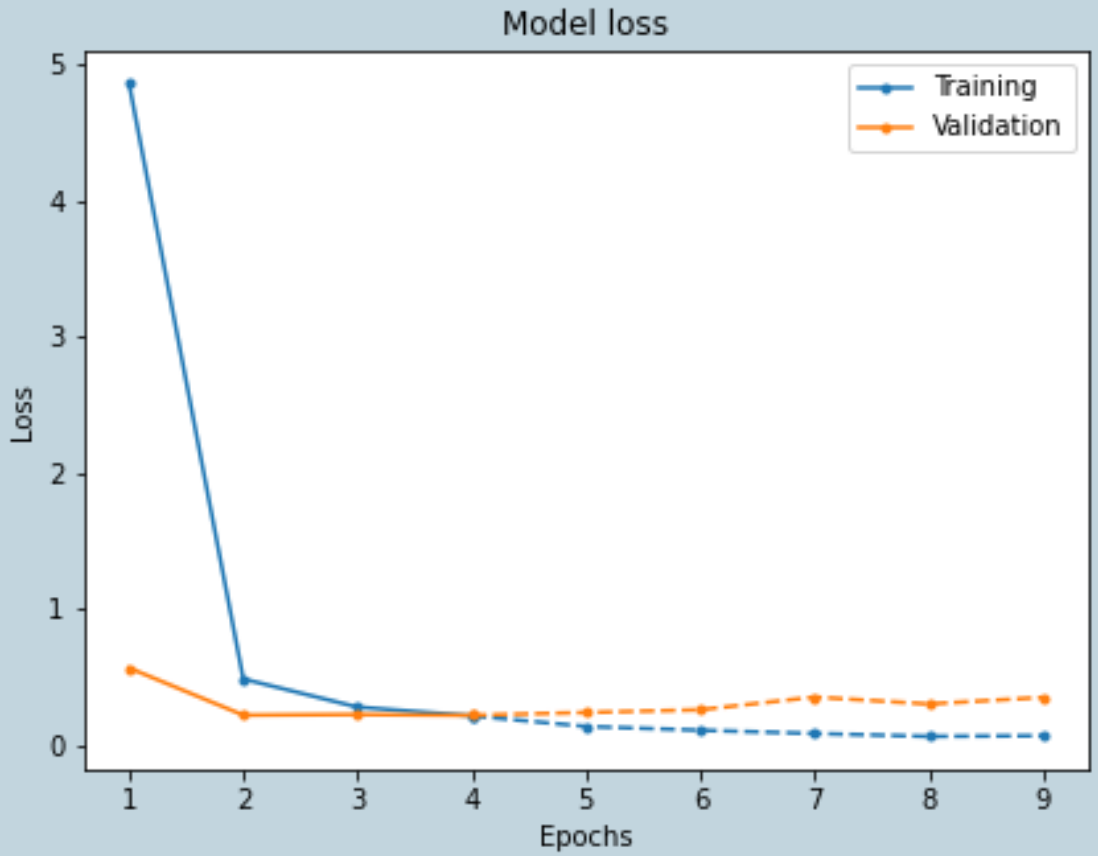
Il modello oltre al layer di input è stato costruito nel seguente modo:

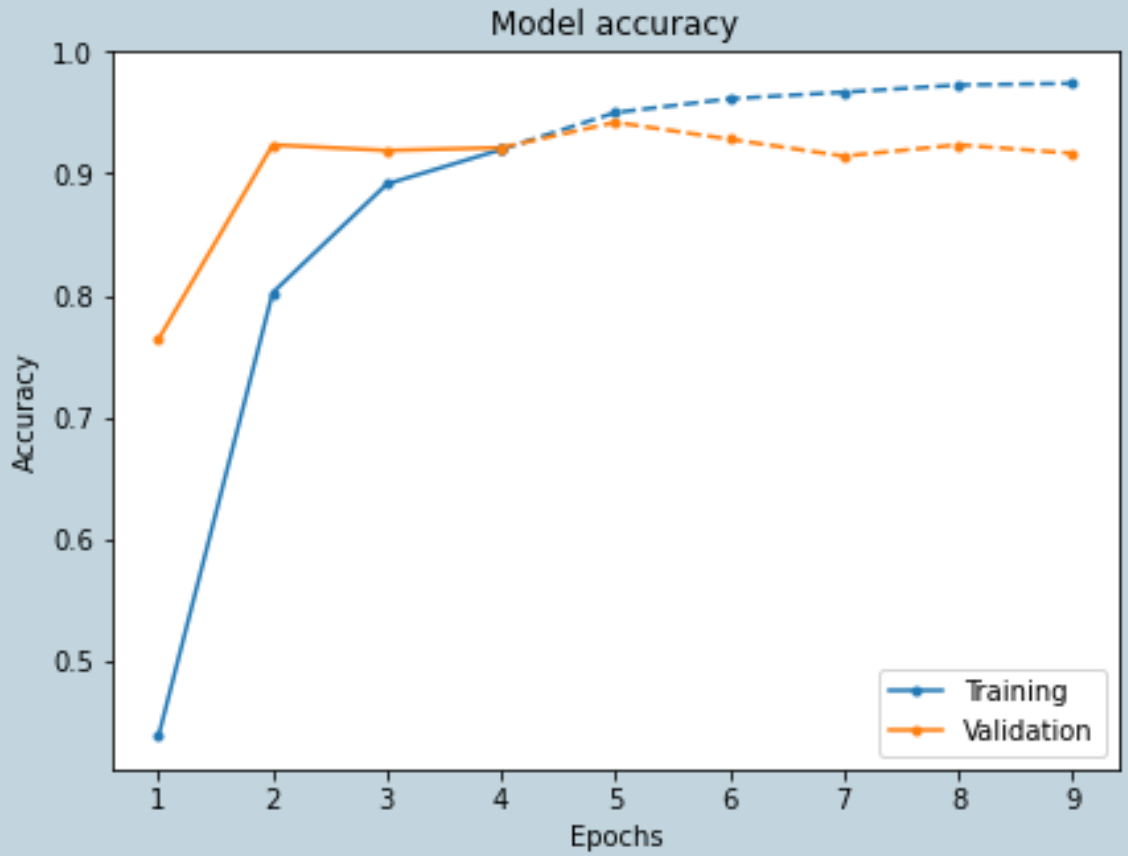
- Normalization()
- Conv2D(32, (3,3), 'relu')
- Conv2D(64, (3,3), 'relu')
- MaxPooling2D(pool_size = (2,2))
- Dropout(0.1)
- Flatten()
- Dense(128, activation='relu')
- Dropout(0.2)
- Dense(64, activation='relu')
- Dropout(0.3)
- Dense(3, activation='softmax')



Accuratezza sul test set: 0.88

Training della rete





Osservando il grafico della loss il modello sembra aver sofferto nella prima fase di underfitting e di overfitting nelle ultime epoche, globalmente però l'addestramento è andato abbastanza bene.

Conclusioni

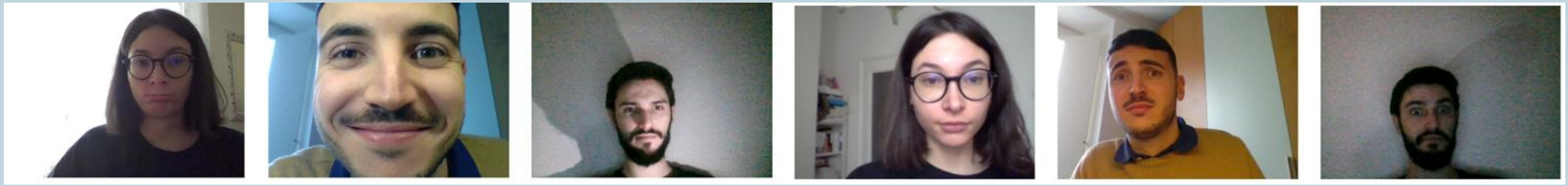
I migliori modelli ottenuti durante l'implementazione sono:

- SVM addestrato sui dati segmentati e aumentati;
- Rete convoluzionale per mfcc *from scratch* addestrata su dati segmentati;
- Rete convoluzionale per spettrogrammi addestrata su dati segmentati e aumentati.

Si sono rivelati dei buoni classificatori anche per audio registrati in un contesto diverso da quello della lettura. Perciò sono stati scelti per la realizzazione della demo.


FACE
RECOGNITION

Il dataset



Le foto acquisite per generare il dataset rispettano le seguenti caratteristiche:

- Scattate in ambiente casalingo;
- Variazioni di espressione e di luce;

Sono state acquisite in totale 300 foto, di cui il 90% è stato usato come training e validation e le rimanenti come test. A tutte queste foto è stato applicato un **face detector**, *dlib*, per estrarre il crop del solo volto.

In una seconda fase di testing si sono introdotte anche nuove immagini dei tre componenti del gruppo scattate in situazioni “reali”, così da permettere una verifica più concreta dei risultati ottenuti dai diversi modelli.

Approccio metodologico

Date le poche immagini a disposizione si è deciso di utilizzare approcci di **fine tuning**. Sono state fissate 100 epoche di addestramento, e utilizzato l'**EarlyStopping**. Inoltre sono stati considerati i seguenti approcci di **data augmentation**:

- **Random flip, c**apovolge casualmente l'immagine in orizzontale;
- **Random rotation,** ruota l'immagine in un certo range di gradi;
- **Random Zoom,** effettua uno zoom (sia zoom-in che zoom-out) dell'immagine sia in altezza che in larghezza;
- **Random contrast:** regola il contrasto in modo indipendente per ciascun canale di un'immagine in base a un fattore casuale.

Questi hanno permesso di ampliare il dataset a disposizione senza effettivamente raccogliere nuovi elementi e di addestrare modelli più robusti e flessibili a possibili cambiamenti di luce o scene di acquisizione. Tutti questi layer sono attivi soltanto durante la fase di training e non durante l'inferenza!

Fine tuning

Per l'approccio di fine tuning si è scelto di utilizzare le seguenti tre architetture:

VGG16

- ImageNet
- Stack convoluzionali 3x3
- +16M parametri
- Accuracy ~0.73

MOBILENET-V2

- ImageNet
- Low resources (per mobile)
- Leggera (14MB)
- +2M parametri
- Accuracy ~0.73

VGGFace

- Face dataset
- Basata su VGG (o Resnet50)
- +145M parametri
- Accuracy ~0.92

Tutte le architetture sono state tagliate al termine dei blocchi convoluzionali e i pesi *freezati*.
Inoltre sono stati aggiunti nuovi layer FC finali differenti per le tre architetture.

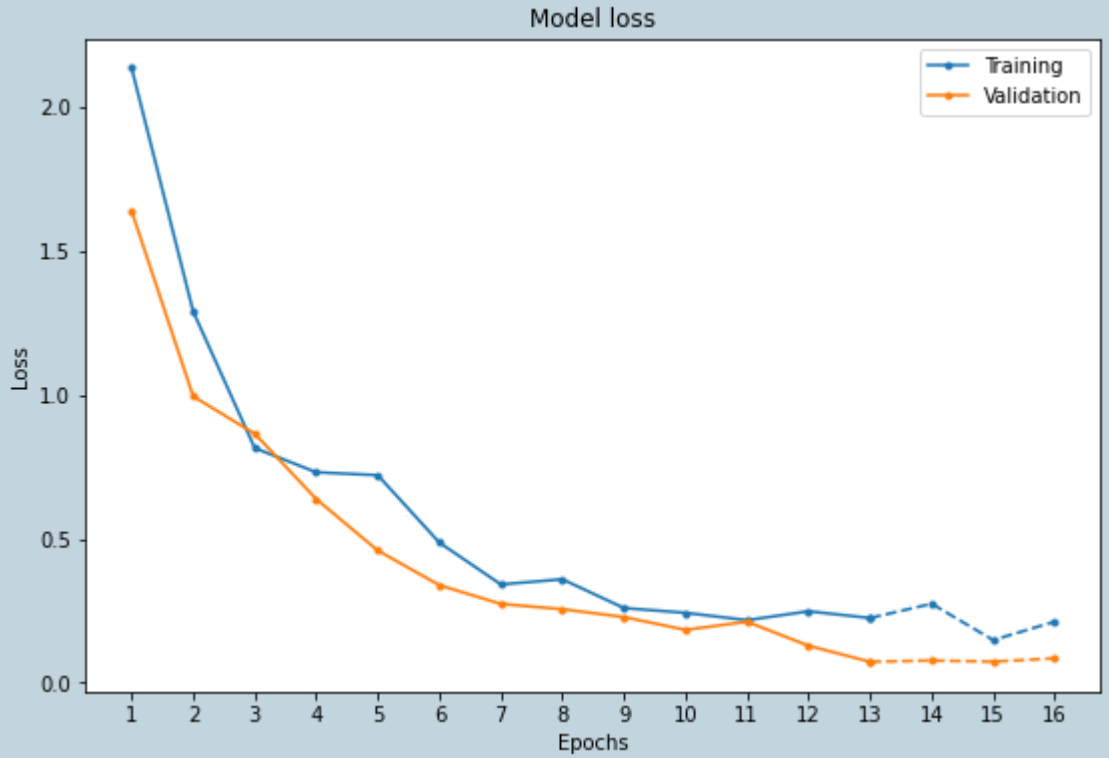
Configurazioni individuate

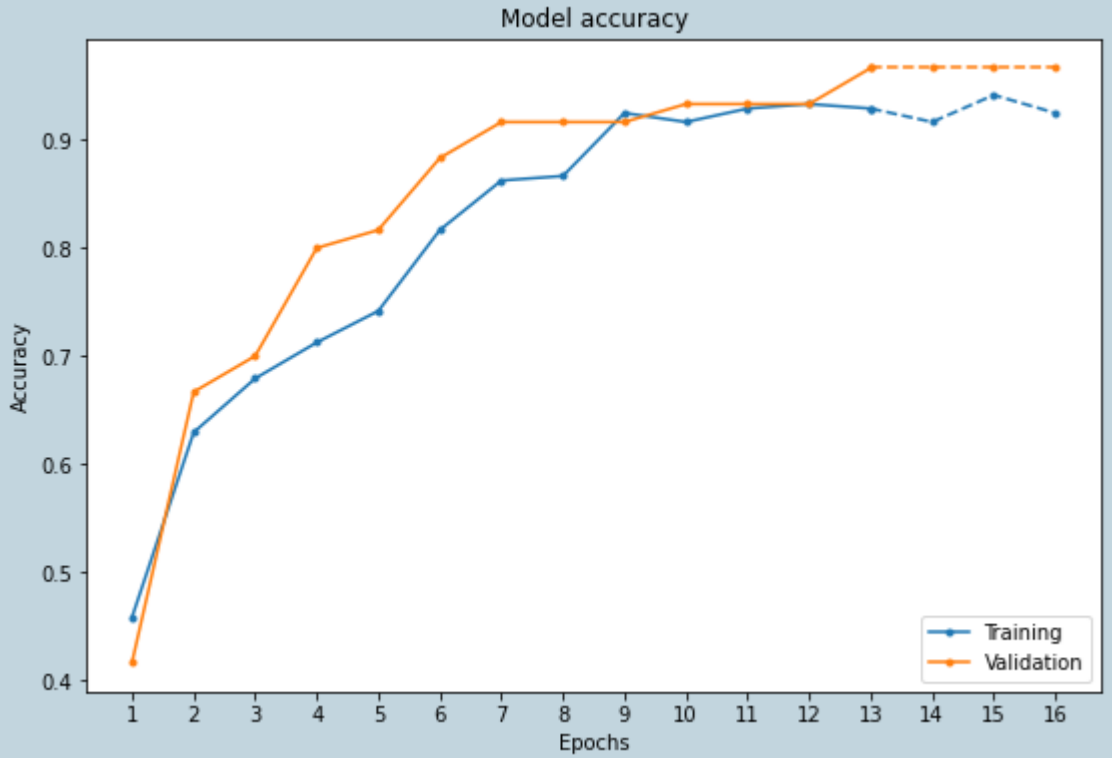
Per tutte e tre le architetture si è provato ad utilizzare differenti iperparametri per cercare di migliorare il più possibile i risultati. Di seguito vengono riportati i migliori parametri individuati:

	VGG16	MobileNet-V2	VGGFace
Configurazione layer FC	[32, 16, 3]	[-, -, 3]	[64, 32, 3]
Probabilità di dropout FC1	0.2	-	-
Random rotation	[-0.35, 0.35]	[-0.4, 0.4]	Non considerato
Random contrast	[-0.2, 1.8]	[-0.2, 1.8]	Non considerato
Random zoom in altezza	[-0.6, 0.6]	[-0.5, 0.5]	Non considerato
Random zoom in larghezza	[-, -]	[-0.5, 0.5]	Non considerato
Ottimizzatore	Adam(1e-04)	RMSprop(1e-03)	Adam(1e-04)
Accuracy	0.966	0.983	**1**

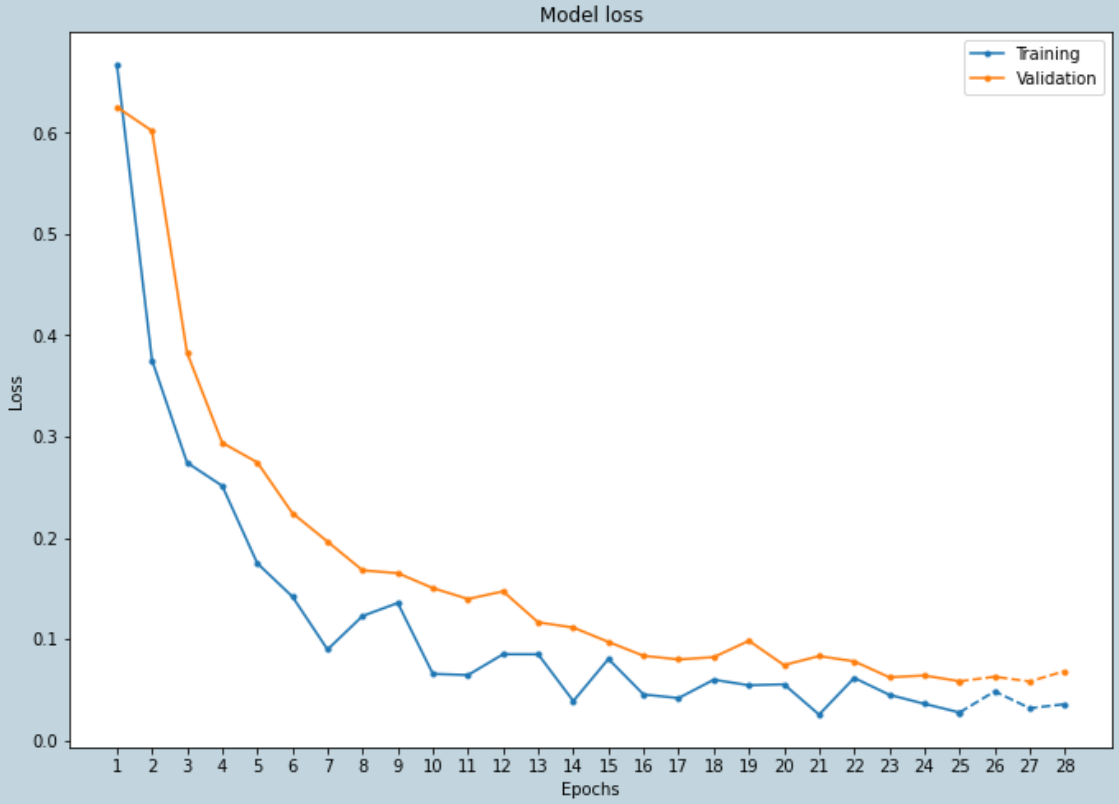
Si ricorda che per tutte si è utilizzato anche il random flip orizzontale.

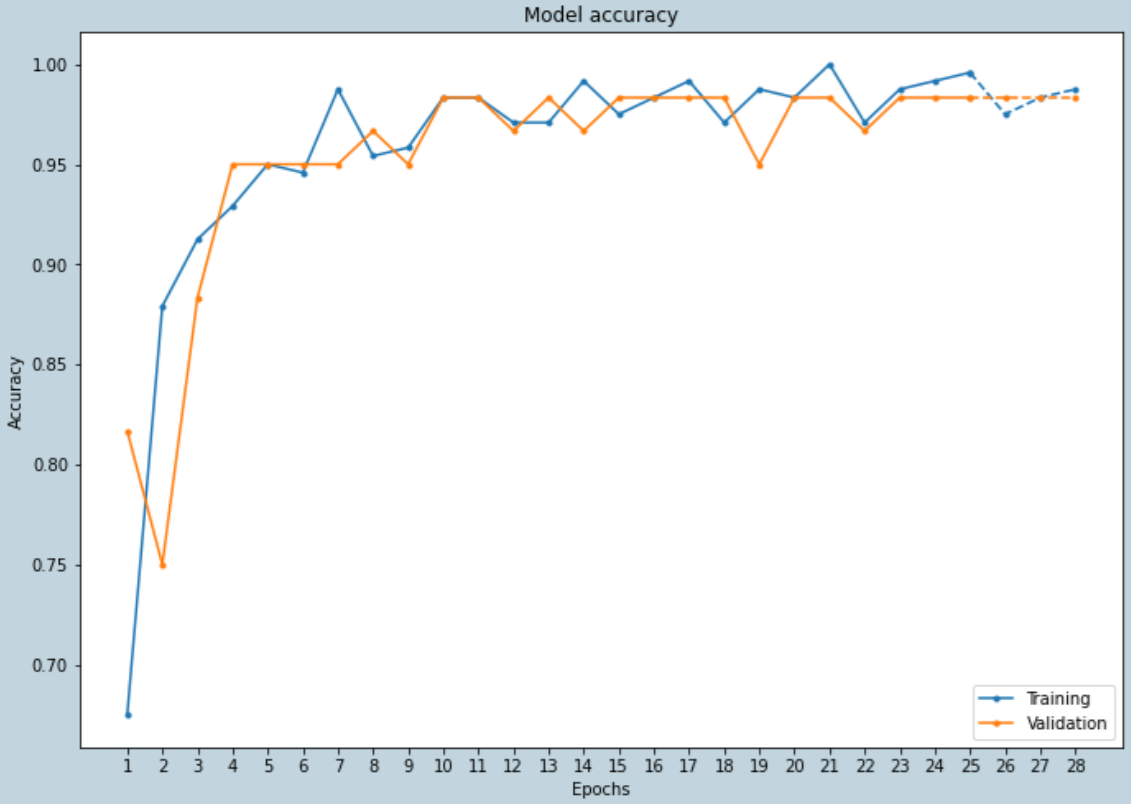
VGG | Training



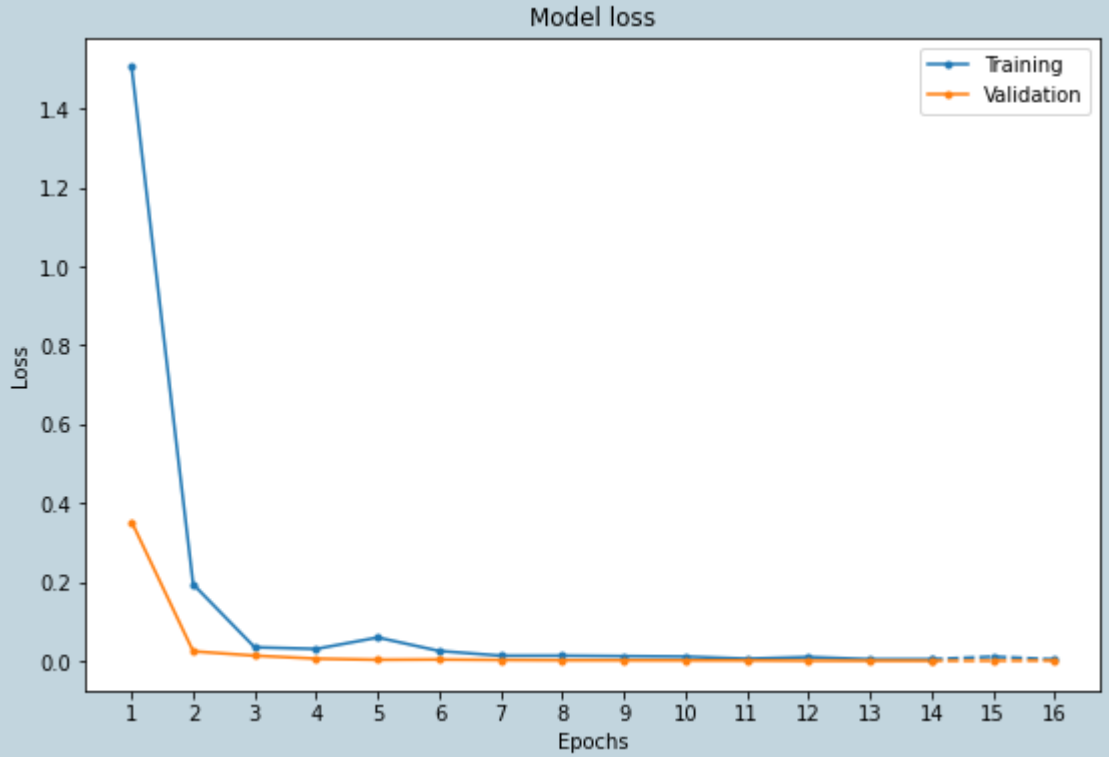


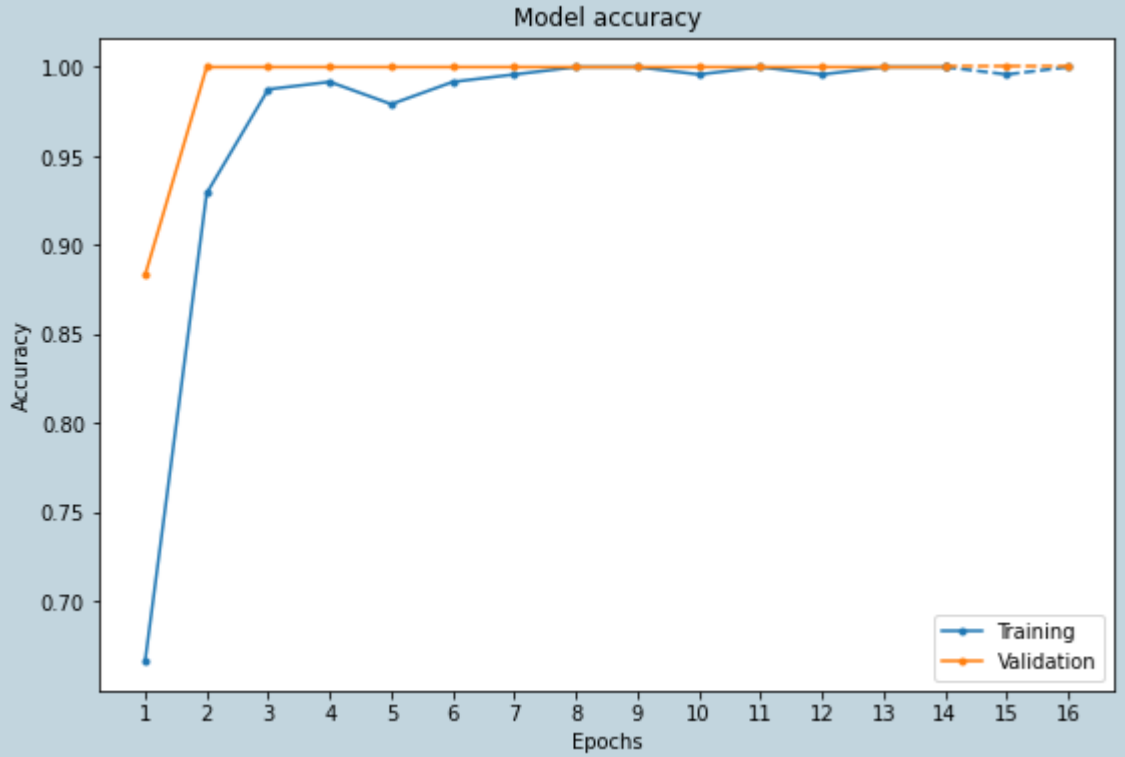
MobileNet-v2 | Training





VGGFace | Training



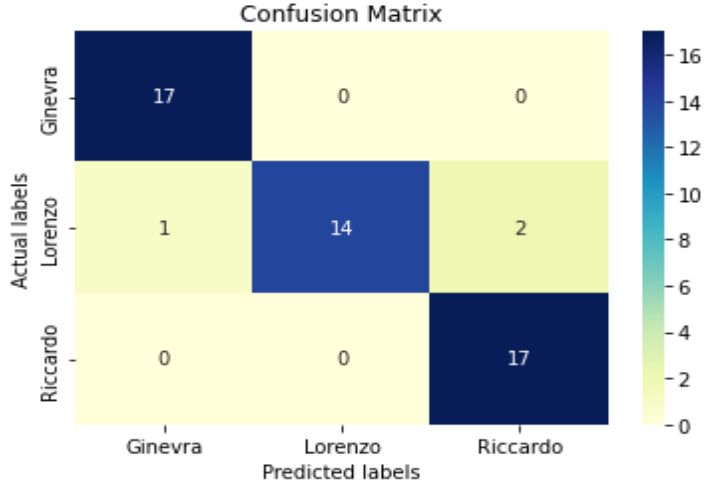


Risultati

VGG16

Classification Report

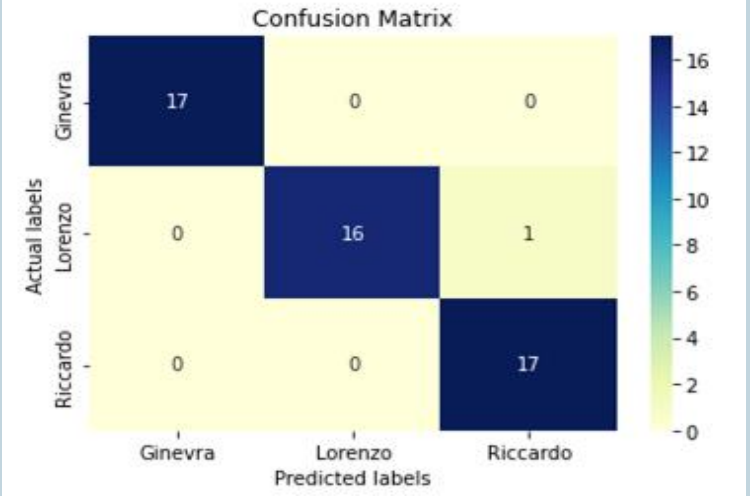
	precision	recall	f1-score
Ginevra	0.94	1.00	0.97
Lorenzo	1.00	0.82	0.90
Riccardo	0.89	1.00	0.94
accuracy			0.94
macro avg	0.95	0.94	0.94
weighted avg	0.95	0.94	0.94



MOBILENET-V2

Classification Report

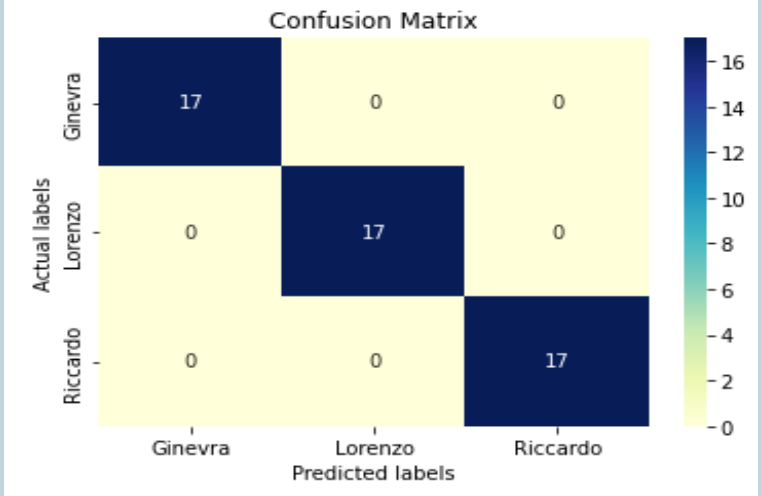
	precision	recall	f1-score
Ginevra	1.00	1.00	1.00
Lorenzo	1.00	0.94	0.97
Riccardo	0.94	1.00	0.97
accuracy			0.98
macro avg	0.98	0.98	0.98
weighted avg	0.98	0.98	0.98



VGGFace

Classification Report

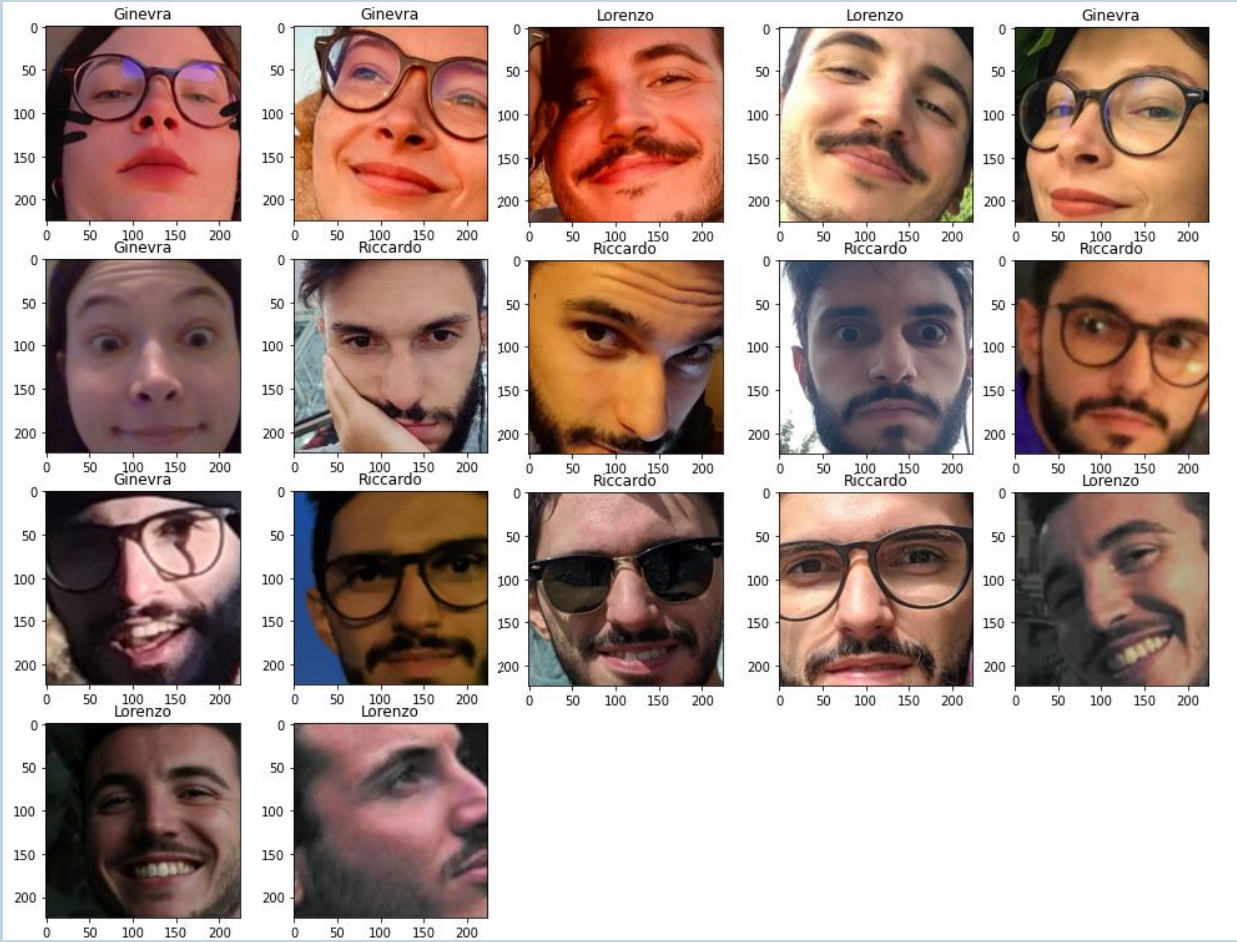
	precision	recall	f1-score
Ginevra	1.00	1.00	1.00
Lorenzo	1.00	1.00	1.00
Riccardo	1.00	1.00	1.00
accuracy			1.00
macro avg	1.00	1.00	1.00
weighted avg	1.00	1.00	1.00



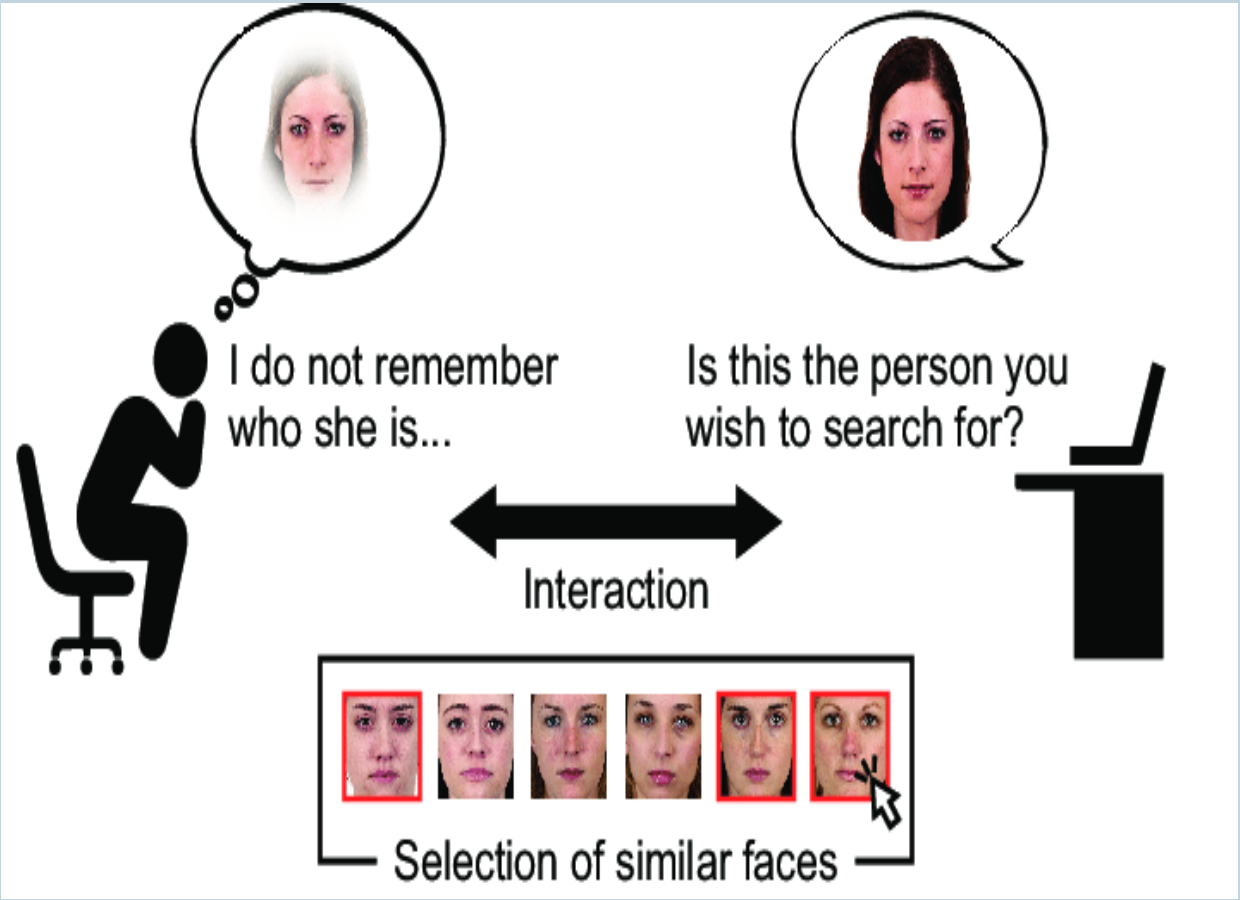
Risultati Wild

Soffermandosi sulle tempistiche di inferenza MobileNet-V2 risulta essere la più veloce con circa *0.125s* per immagine, VGG e VGGFace hanno tempistiche simili con *1.69s* e *1.86s* rispettivamente per immagine. Questa metrica può essere importante in base al particolare contesto applicativo finale.

Analizzando l'inferenza sul test set contenente immagini "reali" i risultati sembrano essere in linea con quanto già osservato. Ovvero MobileNet-V2 e VGG16 ottengono circa le stesse performance classificando in modo errato rispettivamente 6 e 5 immagini sulle 17 a disposizione. Mentre VGGFace commette *un solo errore*, l'immagine in questo caso riporta però metà volto in ombra.



Face Retrieval



Approccio metodologico

Per questo task si è utilizzato il dataset consigliato per le immagini dei volti vip. A partire da questo è stato applicato il face detector *dlib* per estrarre i soli volti sui quali poi si sono estratte le features. Questo perché volevamo evitare di ottenere immagini simili dovute al contesto (sfondo, ambiente, etc.).

Inoltre per **problemi computazionali** e di tempo si è deciso di utilizzare solo un subset rispetto alle oltre 200 mila immagini di volti vip a disposizione. Si è quindi scelto di considerare al più **20 immagini** per ogni vip.

Approccio metodologico (2)

In totale sono state estratte le features di **31.496 volti** di **tutti i 1500 vip** a disposizione. Si è scelto di non fissare un numero di immagini globali per non escludere a priori dei vip. Inoltre, in alcuni casi, non si hanno tutte e 20 le immagini in quanto i dati a disposizione erano inferiori o non si è riusciti ad estrarre i volti col face detector scelto per via della ridotta dimensione dell'input.

Per l'estrazione delle features sono stati considerati due modelli neurali preaddestrati **MobileNet-V2** e **VGGFace**. Tutte le features sono state poi rappresentate tramite la struttura ad albero **KDTree**.

MobileNet-V2

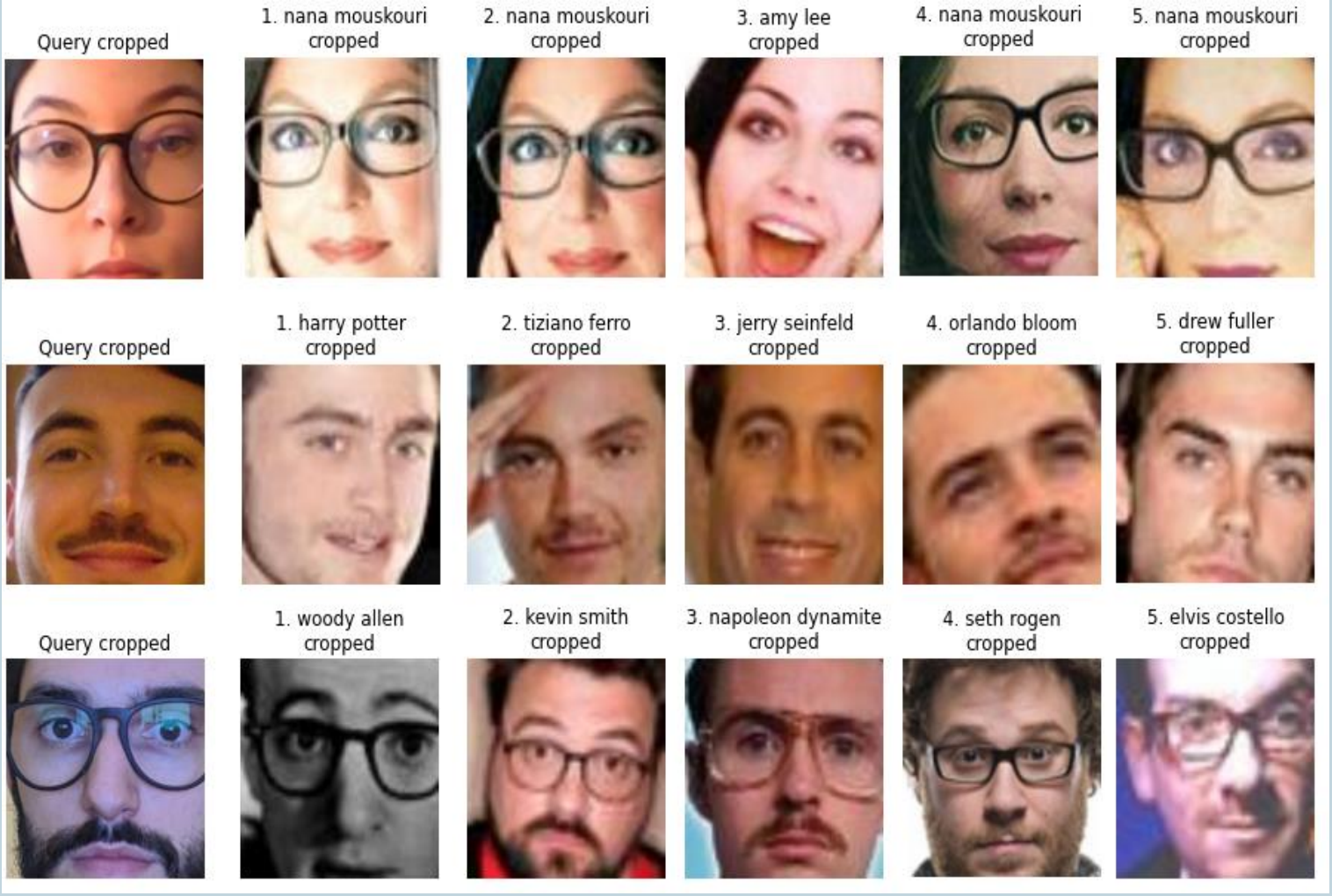
Questa rete ha permesso di estrarre **1024 features** di riferimento per ogni singolo volto.

Successivamente la stessa rete è stata applicata alle immagini di query per ottenere la medesima rappresentazione. Di seguito è possibile osservare i cinque volti più simili ottenuti per le tre query:

VGGFace

Questa rete, basata su VGG, ha permesso di estrarre **516 features** per ogni singolo volto.

Come nel caso precedente la stessa rete è stata poi applicata alle immagini di query. Di seguito è possibile osservare i cinque volti più simili ottenuti per le tre query:



Considerazioni finali

Valutando i risultati ottenuti ci si ritiene abbastanza soddisfatti dal retrieving ottenuti con VGGFace. Mentre MobileNet-V2 ha mostrato alcuni limiti dovuti ai pesi pretrainati su altre tipologie di immagini rispetto ai volti.

Inoltre è importante notare che per VGGFace nel caso di 'Ginevra' si ottiene *più volte la stessa persona*. Questo può essere un risultato non voluto e può quindi essere oggetto di miglioramenti futuri.

Infine si è notato come la qualità delle immagini dei vip e della query in input incida molto sui risultati.

DOMANDE?

Grazie per l'attenzione!