

UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

DIGITAL SIGNAL AND IMAGE MANAGEMENT

Progetto Finale

Authors:

Mattia Boller - 873358 - m.boller@campus.unimib.it
Raffaele Moretti - 794537 - r.moretti8@campus.unimib.it
Silvia Ranieri - 878067 - s.ranieri7@campus.unimib.it



Indice

Sommario	1
1 Speaker Recognition	2
1.1 Preparazione dei dati	2
1.2 Costruzione del modello	2
1.3 Training del modello	2
1.4 Performance del modello	3
2 Face Recognition	4
2.1 Preparazione dei dati	4
2.2 Costruzione del modello	4
2.3 Training del modello	5
2.4 Performance del modello	5
3 Image Retrieval	7
3.1 Preparazione dei dati	7
3.2 Feature extractor	7
3.3 Costruzione struttura dati per CBIR	8
3.4 Test	8
Bibliografia	11

Sommario

Nel seguente progetto dedicato alla materia **Digital Signal and Image Management**, sono stati studiati diversi approcci per 3 tipi di problemi differenti, i quali verranno brevemente descritti di seguito insieme ad un veloce riassunto sulle soluzioni adottate:

- **Processing di segnali mono-dimensionali:** si tratta di un task di speaker recognition, i quali speaker da riconoscere sono i 3 membri del gruppo. Il problema è stato affrontato allenando una semplice rete neurale fully-connected con spezzoni da un secondo di diverse registrazioni appartenenti ai 3 speaker da riconoscere.
- **Processing di segnali bi-dimensionalni:** si tratta di un task di face recognition, i quali visi da riconoscere sono anche in questo caso quelli dei membri del gruppo. L'approccio al problema si è basato sulla libreria OpenCV per la face detection e il successivo allenamento di una rete neurale convoluzionale tramite transfer learning e fine tuning. La face recognition è stata successivamente implementata in real-time a video, la quale ha dimostrato robustezza anche in diverse condizioni di luce, espressioni e angolazioni diverse.
- **Content based image retrieval:** in questa ultima parte, l'obiettivo è stato quello di restituire 10 volti più simili ad un volto dato come input, estrapolandoli da un ampio dataset di foto di celebrità. Per raggiungere l'obiettivo si è fatto uso una seconda volta della libreria OpenCV per la face detection, i visi sono stati utilizzati per allenare un autoencoder, dal quale è stata estratta la parte di encoding da utilizzare come feature extraction per le foto. Per trovare i volti più simili quindi si è misurata la distanza tra le feature delle varie foto del dataset e quella di query.

1 Speaker Recognition

La prima parte del progetto è dedicata alla speaker recognition, quindi al riconoscimento dell'identità di un individuo a partire da una sua registrazione. Nella versione del problema affrontato in questo lavoro, non sono stati posti vincoli sulle frasi da dire o sulla qualità della registrazione.

1.1 Preparazione dei dati

Gli audio per allenare il modello di speaker recognition, sono stati raccolti attraverso uno script Python dedicato alla registrazione di tracce da 2 minuti, con una frequenza di campionamento pari a 44100Hz. Ogni membro del gruppo ha registrato 5 audio durante la lettura di testi diversi, in ambienti diversi. Inoltre è stato registrato un audio a testa da 10 minuti da utilizzare in fase di test. Le tracce audio sono state caricate, divise in chunk da 1 secondo e per ogni slice ne sono state estratte le feature tramite **MFCC** (Mel-frequency cepstrum) [1].

Al momento del caricamento delle registrazioni, è stata caricata anche l'identità dello speaker di una determinata traccia da utilizzare come label al momento di training e test del modello. Le label sono state convertite in formato one-hot encoding.

1.2 Costruzione del modello

Come classificatore per svolgere il riconoscimento è stata impiegata una rete neurale, la quale architettura è specificata di seguito:

- Dense layer: 128 neuroni, ReLU
- Dropout: 0.5
- Dense layer: 3 neuroni, Softmax

Come ottimizzatore la scelta è ricaduta su Adam, con una loss function di tipo categorical-crossentropy e l'accuratezza come metrica di misurazione delle performance.

1.3 Training del modello

Il modello è stato allenato per 10 epoche, utilizzando il 10% del training dataset come validation set, in modo da poter monitorare l'andamento della loss e dell'accuracy durante l'allenamento. Come batch size si è scelto un valore pari a 64.

Di seguito, in figura 1, sono presentati i grafici dell'andamento di loss e accuracy durante il training della rete.

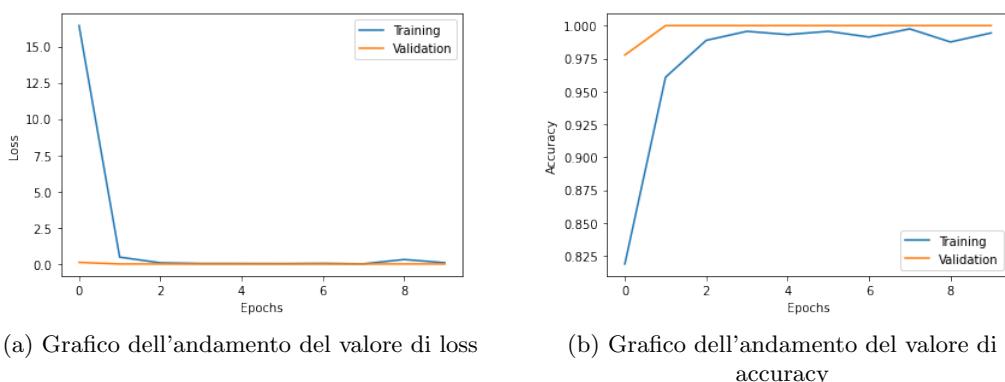


Figura 1: Grafici di monitoring della fase di training

Come è possibile vedere dai grafici, il modello ha raggiunto rapidamente il 100% di accuracy sul validation set.

1.4 Performance del modello

Il modello è stato testato sulle registrazioni da 10 minuti, divise in slice da 1 secondo, ottenendo il 100% di accuracy per ogni speaker.

Per permettere la classificazione di frasi intere di lunghezza variabile in una situazione più vicina alla realtà, si è deciso di testare il comportamento della rete attraverso una funzione che si occupa di registrare frasi di 5 secondi, le divide in slice da 1 secondo, ne calcola le feature attraverso MFCC, effettua la predizione dell'identità dello speaker per ogni chunk attraverso la rete e restituisce come predizione finale lo speaker che più volte viene riconosciuto dal modello in quelle 5 slice. Sono stati quindi fatti ascoltare attraverso lo stesso microfono alla funzione costruita 50 secondi degli audio da 10 minuti registrati in precedenza, dividendo i 50 secondi in 10 frasi da 5 secondi le quali sono state classificate con il metodo illustrato in precedenza. Anche in questo caso il modello ha dimostrato performance perfette ottenendo il 100% di accuracy per ogni speaker testato.

2 Face Recognition

La parte del progetto dedicata alla face recognition pone come problema quello di creare un modello capace di riconoscere, attraverso una foto o video, l'identità di una persona all'interno dei componenti del gruppo. Di seguito viene illustrato l'approccio seguito per risolvere il suddetto task.

2.1 Preparazione dei dati

I dati sono stati prima di tutto raccolti, quindi ogni membro del gruppo è stato incaricato di recuperare 30 foto di sé, con angolazioni del viso diverse e con condizioni di luce differenti.

Alle immagini raccolte sono stati estratti solamente i visi sfruttando la libreria OpenCV e l'Haar cascade classifier [2]. In alcuni casi, il rilevatore restituiva falsi positivi insieme alla reale faccia da estrarre, per risolvere il problema si è mantenuta solamente la rilevazione di dimensione maggiore, solitamente corrispondente al corretto viso da individuare.

Le nuove immagini sono state portate tutte alla dimensione, pari a 224x224, in modo che fossero in un formato adatto come input del modello. In figura 2 è possibile vedere un esempio di estrazione automatica del viso da una foto.

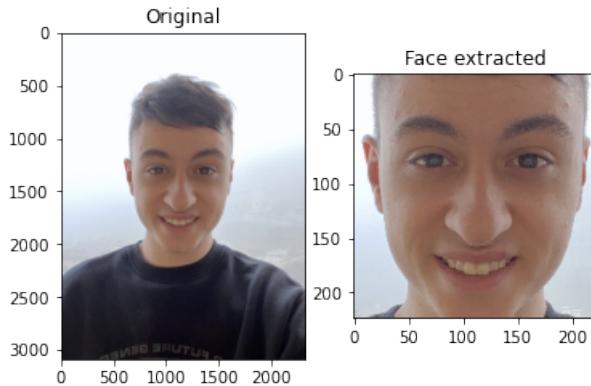


Figura 2: Esempio di detection del viso

Insieme ai visi, sono state estratte le label corrispondenti all'identità del membro del gruppo raffigurato nella foto e successivamente portate in formato one-hot encoding.

2.2 Costruzione del modello

Come modello scelto per risolvere il task di face detection, si è deciso di fare affidamento ad una **rete neurale convoluzionale**, classico tipo di rete neurale utilizzato per trattare immagini.

Data il numero molto limitato di foto a disposizione per il training del modello, si è scelto di sfruttare tramite transfer learning una rete pre-allenata, adattandola al problema corrente tramite fine tuning. La rete selezionata è stata la MobileNetV2, rete più compatta rispetto ad altre disponibili, scelta dettata dalla necessità di avere predizioni veloci, data l'intenzione di applicare la face recognition a video in real-time. La rete è stata quindi importata con i pesi derivanti dall'allenamento su ImageNet e rimuovendo gli ultimi layer.

Altra soluzione adottata per far fronte al problema della poca quantità di dati, è stata quella di applicare data augmentation alle immagini a tempo di training. Ad ogni epoca quindi le foto subiscono delle trasformazioni randomiche di tipo: riflessione orizzontale, rotazione, zoom e livello di contrasto. Di seguito viene presentata l'architettura del modello:

- Input: 224x224x3
- Data augmentation
- MobileNetV2
- Dense layer: 512 neuroni, ReLU
- Dense layer: 256 neuroni, ReLU
- Dense layer: 3 neuroni, Softmax

Come funzione di loss è stata adottata la categorical crossentropy, come algoritmo di ottimizzazione Adam e come metrica per misurare le performance del modello l'accuratezza.

2.3 Training del modello

Il modello è stato allenato per 20 epoch, usando batch di grandezza 16, utilizzando il 90% dei dati disponibili, di cui il 10% è stato tenuto come validation set per analizzare l'andamento della loss ed evitare casi di overfitting o underfitting. In figura 3 sono presentati i grafici di loss e accuracy del modello sui dati di training e di validation.

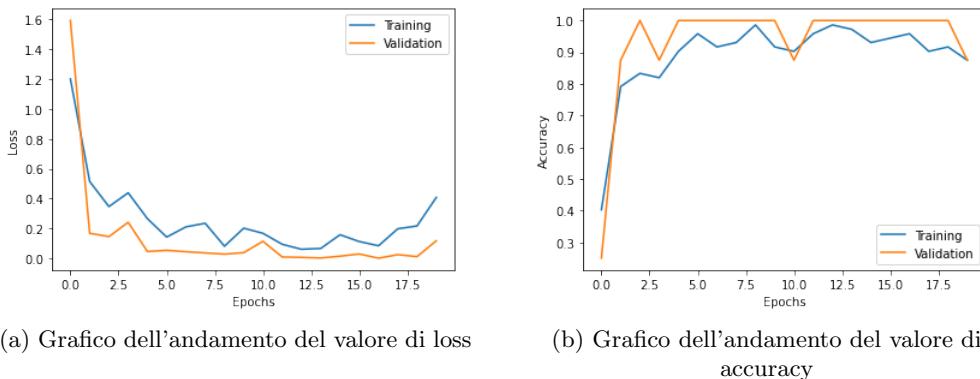


Figura 3: Grafici di monitoring della fase di training

2.4 Performance del modello

Il modello, in seguito al training, ha raggiunto un'accuracy pari al 100% sulle immagini di test, tuttavia non sono una quantità tale da permettere di prendere queste metriche come stime reali di performance della rete.

Per capire quali fossero le vere performance del modello, si è deciso di testarlo in un task di riconoscimento a video in real-time dell'identità del volto. Ogni frame del video viene prima processato in modo da estrarre i volti presenti, i quali vengono utilizzati come input per la rete che ne restituisce quindi la presunta identità. A video vengono mostrati i limiti della zona contenente il volto rilevato, l'identità presunta rilevata dalla rete e il livello di confidenza della predizione. Per livelli di confidenza al di sotto del 70%, il modello restituisce come identità 'Unknown'. Con questo metodo è stato possibile simulare un test set di dimensioni elevate, con immagini molto diverse tra loro per quanto riguarda angolazione del video, cambiamenti di luce, distanza dall'obiettivo ed espressioni. Il modello ha dimostrato ottime performance soprattutto con movimenti lenti del soggetto, meno preciso invece durante movimenti più veloci, a causa della pessima qualità dei frame estratti in situazioni di movimenti bruschi. Nella figura 4 sono presenti 2 frame di un video registrato durante un riconoscimento real-time del viso. Nel caso dei frame proposti, la rete ha riconosciuto correttamente l'identità

dell'individuo in entrambi i casi, con una confidenza del 97.8% in una situazione di buona illuminazione e con una confidenza del 87.0% in una situazione di bassa illuminazione.

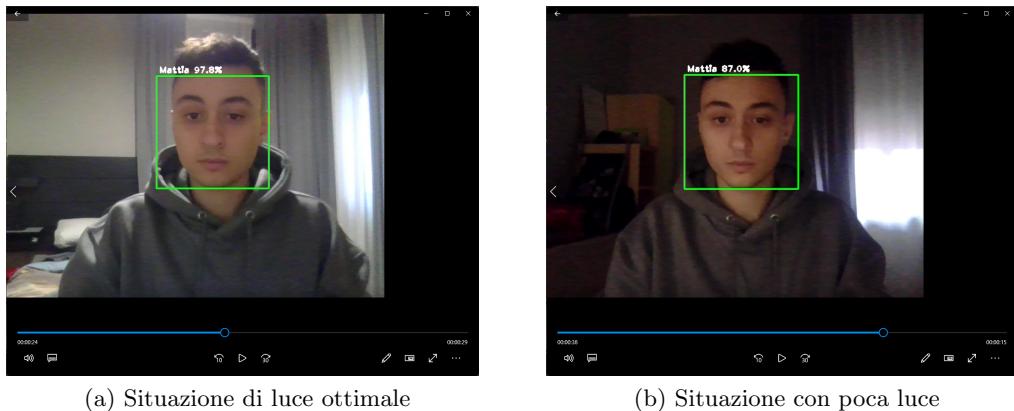


Figura 4: Frame di esempio di riconoscimento facciale in real-time

3 Image Retrieval

L'obiettivo della parte del progetto dedicata al Content Based Image Retrieval, è restituire le 10 immagini in cui la persona raffigurata presenta un viso il più simile rispetto alla persona ritratta in una foto in input. Nei paragrafi seguenti verranno presentate le tecniche usate per completare il task.

3.1 Preparazione dei dati

Le immagini da cui estrarre i visi più simili ad una faccia data in input, sono disponibile al seguente link: <https://skydrive.live.com/?cid=1e04f731c1dd71bcid=1E04F731C1DD71BC!105>. Il dataset contiene 202792 foto appartenenti a 1583 persone famose diverse.

Le immagini sono state utilizzate in due fasi:

- Training del modello di estrazione delle feature: massimo 3 foto per persona
- Costruzione dell'albero di ricerca: massimo 10 foto per persona

In entrambi i casi, dalla foto è stato estratto il viso della persona raffigurata utilizzando lo stesso metodo descritto nel paragrafo 2.1 dedicato alla face recognition, con la sola differenza che le immagini sono state portate ad una dimensione pari a 128x128. In seguito all'estrazione dei visi, i valori contenuti nelle matrici rappresentati le immagini sono stati portati tra 0 e 1.

3.2 Feature extractor

Si è deciso di sfruttare autoencoder per estrarre le feature da utilizzare come rappresentazione compressa delle immagini da cui poi calcolare la distanza con la foto di query in input. Un autoencoder è un particolare tipo di rete neurale, la cui architettura si compone di due parti: encoder e decoder. La prima parte si occupa di ridurre la dimensionalità del dato in input mentre la seconda, partendo dalla rappresentazione ridotta creata dall'encoder, cerca di ricostruire il dato originale. Attraverso il training quindi si ottiene che l'encoder imparerà a generare una rappresentazione compressa limitando al minimo la perdita di dati. Di seguito viene presentata l'architettura utilizzata:

Encoder:

- Conv2D layer: 32 filtri 3x3, ReLU
- MaxPooling: 2x2
- Conv2D layer: 16 filtri 3x3, ReLU
- MaxPooling: 2x2
- Conv2D layer: 8 filtri 3x3, ReLU
- MaxPooling: 2x2

Decoder:

- Conv2DTranspose: 8 filtri 3x3, ReLU
- Conv2DTranspose: 16 filtri 3x3, ReLU
- Conv2DTranspose: 32 filtri 3x3, ReLU
- Conv2D: 3 filtri 3x3, ReLU

Come ottimizzatore la scelta è ricaduta su Adam, mentre come funzione loss è stata scelta la Mean Squared Error. Il training è stato effettuato su un massimo di 3 foto per persona diversa presente nel dataset, allenando per 30 epoche con una batch size pari a 64.

Di seguito, in figura 5 è possibile vedere un esempio delle abilità di compressione e ricostruzione dell'autoencoder alla fine della fase di training.

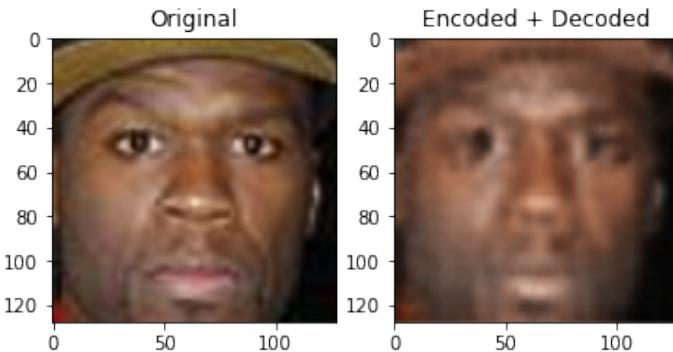


Figura 5: Esempio di compressione e ricostruzione di una foto

In seguito all'allenamento, dall'autoencoder è stata estratta solamente la parte di encoder, modello utilizzato per effettuare feature extraction dalle immagini. Le foto entrano nel modello con una grandezza pari a 128x128x3 (49152 valori) ed escono con una dimensione pari a 16x16x8 (2048), viene quindi applicato un rateo di compressione pari a 24.

3.3 Costruzione struttura dati per CBIR

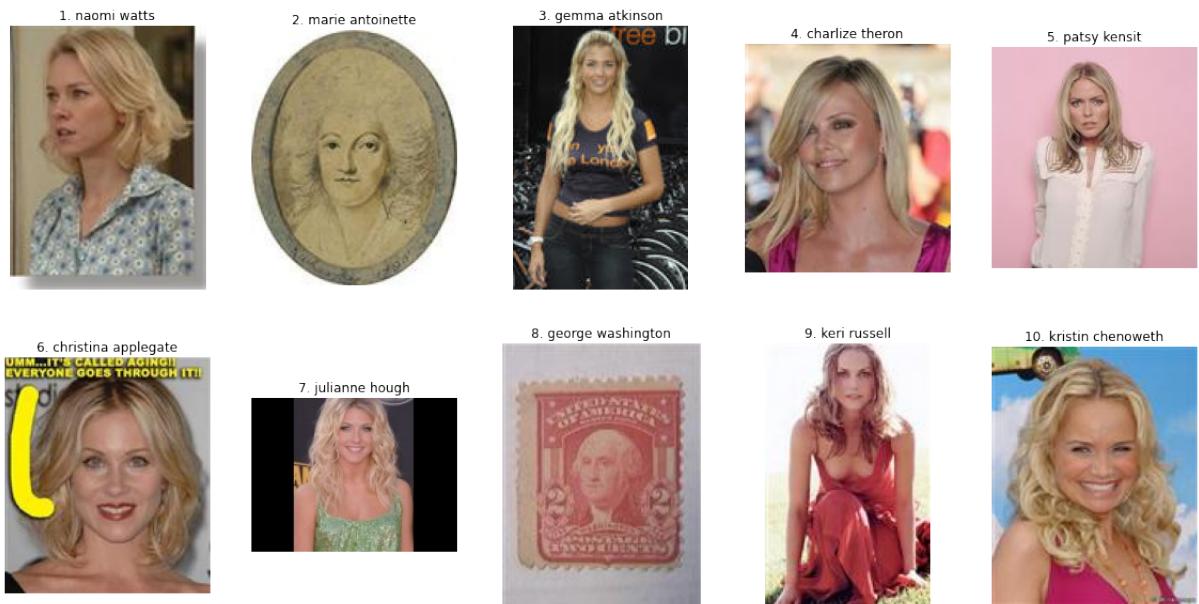
Come struttura dati per permettere il retrieval veloce dei visi più simili al viso in input, è stato utilizzato il KDTree messo a disposizione dalla libreria sklearn, struttura basata sul partizionamento dello spazio per l'organizzazione dei punti in uno spazio k-dimensionale [3]. Inoltre sono state inserite nell'albero le feature appartenenti a 99148 foto, compresse dall'autoencoder in circa 160 minuti. Le query di test sull'albero hanno dimostrato di essere molto veloci presentando tempi medi pari a 0.4 secondi per estrarre le prime 10 foto più simili.

3.4 Test

In figura 6 e in figura 7, sono presentati due esempi di estrazione dei 10 volti più simili contenuti nel dataset di partenza, rispetto a due dei membri del gruppo che ha lavorato al seguente progetto.

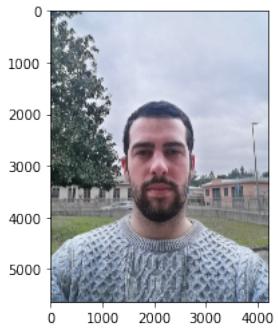


(a) Immagine di input



(b) 10 volti più simili in ordine di distanza

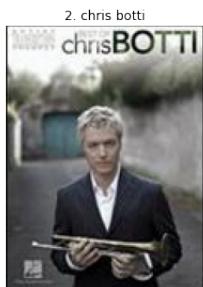
Figura 6: Esempio di CBIR



(a) Immagine di input



1. dmitry medvedev



2. chris botti



3. chris botti



4. jake gyllenhaal



5. julianne hough



6. benedicto xvi



7. julianne hough



8. richard dawkins



9. john cleese



10. doris lessing

(b) 10 volti più simili in ordine di distanza

Figura 7: Esempio di CBIR

Riferimenti bibliografici

- [1] “Mfcc technique for speech recognition,” Jun 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/>
- [2] “Cascade classifier.” [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- [3] “K-d tree,” Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/K-d_tree