

UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica



Corso di Laurea in Informatica

Fondamenti di Visione Artificiale e Biometria

Progetto: Age Regression

Relatori

Prof. Michele Nappi

Dott.ssa Paola Barra

Studenti

Domenico Trotta 0522500810

Michele Castellaneta 0522500824

Anno Accademico 2019/2020

Sommario

1. Introduzione	3
2. Dataset	4
2.1 Analisi dei dati.....	5
3. Lavori precedenti.....	8
3.1 UTKFace - age and sex prediction	9
3.2 Age and Gender Estimation using CNN	10
4. Lavoro svolto	11
4.1 Caricamento dataset e Pre-processing.....	11
4.2 Caricamento modello	13
4.3 Training modello	16
4.4 Testing	17
5. Valutazioni	19
6. Altri modelli.....	26
7. Descrizione demo	27
7.1 Age prediction con una sola persona	27
7.2 Age prediction con due persone	28
7.3 Schermata impostazioni	29
8. Conclusioni	31
9. Riconoscimenti	32
10. Riferimenti	32

1. Introduzione

L'elaborato discusso in questo documento tratta di un progetto svolto durante il corso di *Fondamenti di Visione Artificiale e Biometria*, del corso di *Laurea Magistrale in Informatica dell'Università degli Studi di Salerno*.

Lo scopo di tale progetto prevedeva la costruzione di un modello che fosse in grado di effettuare un'operazione di Age Regression a partire da un dataset disponibile liberamente in rete.

Una volta ottenuto tale modello questo è stato utilizzato per realizzare una demo che effettuasse la prediction dell'età del soggetto inquadrato.

Prima di discutere del lavoro svolto e del funzionamento della demo verrà presentata una breve descrizione e analisi del dataset utilizzato e di alcuni dei lavori già svolti in letteratura in merito all'operazione di Age Regression.

Inoltre, sono stati prodotti anche altri file a cui si farà riferimento nei paragrafi successivi. I file in questione sono elencati di seguito:

- AgeReg.ipynb
- demoGUI.py
- Sommario modelli creati.xlsx

2. Dataset

Il dataset scelto per lo sviluppo del progetto è stato UTKFace [1].

UTKFace è un dataset di volti di persone con età che variano da 1 a 116 anni.

Il set di dati comprende oltre 20.000 immagini di volti (precisamente 23708) con annotazioni di età, sesso ed etnia. Le immagini presentano una grande varietà di pose, espressioni facciali, illuminazione, occlusioni e risoluzione. Oltre ad essere usato per progetti di age regression, UTKFace può essere utilizzato per progetti di face detection, age estimation, age classification o landmark localization.

Di seguito sono mostrate alcune immagini presenti nel dataset.



Le etichette di ogni volto sono incorporate nel nome del file con il seguente formato:

[age] — [gender] — [race] — [date&time].jpg

- **[age]** è un numero intero compreso tra 0 e 116, che indica l'età;
- **[gender]** indica il sesso: 0 (maschio) o 1 (femmina);
- **[race]** indica l'etnia, ed è un numero intero compreso tra 0 e 4. In ordine indica l'etnia occidentale, africana, asiatica, indiana e altri (come ispanico, latino, mediorientale);
- **[date&time]** è nel formato yyyyymmddHHMMSSFFF, che mostra la data e l'ora in cui un'immagine è stata raccolta su UTKFace.

2.1 Analisi dei dati

La fase successiva alla scelta del dataset è stata quella di uno studio approfondito dello stesso. Analizzando le immagini presenti nel dataset, si sono riscontrate delle irregolarità in alcune di esse, non rappresentanti volti umani. Tali immagini sono state rimosse dal dataset, in quanto non rappresentavano figure di interesse per lo sviluppo del progetto in questione. Probabilmente esse derivano da irregolari operazioni di “*crop & align*” effettuate sulle immagini originali.

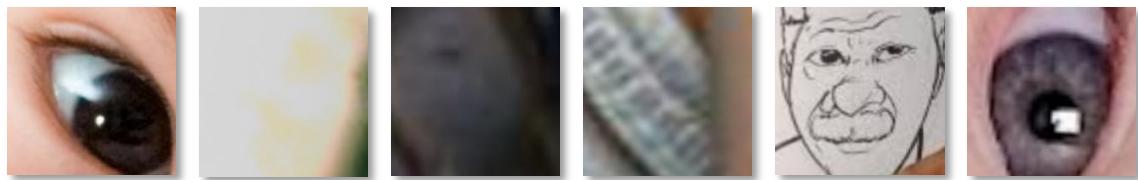


Figura 1 Esempi di immagini eliminate.

Per lo scopo del progetto, l'attributo target è stato rappresentato dall'età. Per la creazione dei modelli sono state invece considerate le caratteristiche di sesso ed etnia, oltre all'immagine stessa. Pertanto, tali attributi verranno analizzati singolarmente di seguito.

2.1.1 Distribuzione età

La prima caratteristica analizzata è stata quella dell'età. In particolare, è stata analizzata la distribuzione dell'età dei volti presenti nel dataset, il cui grafico è rappresentato nella Figura 2.

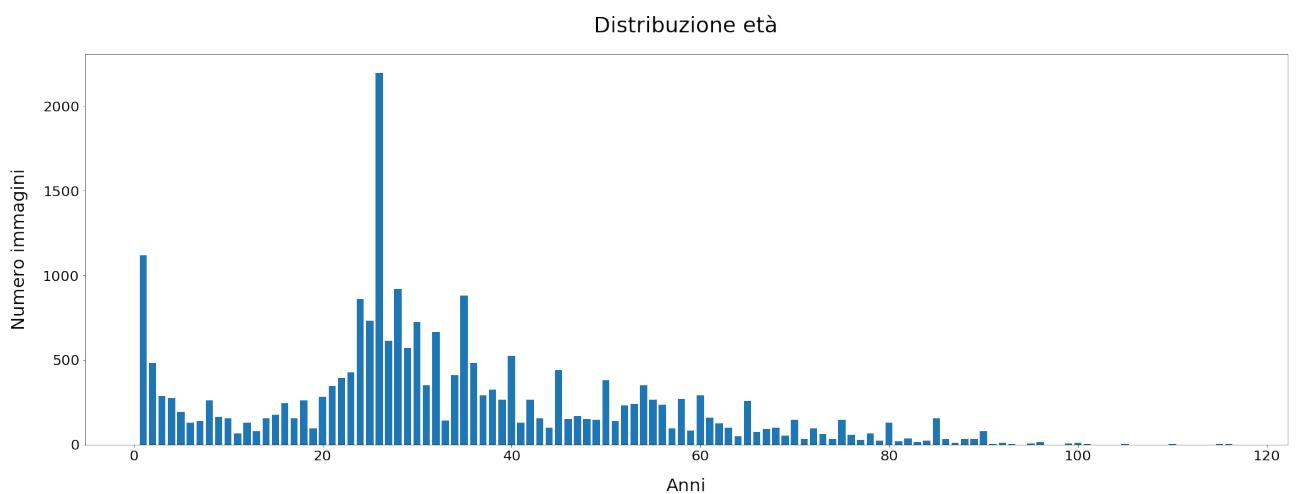


Figura 2 Grafico Distribuzione età.

Analizzando il grafico mostrato in *Figura 2* si può notare che:

- Nella fascia di età compresa tra i 25 e i 27 anni si ha un numero di volti molto maggiore rispetto alla media;
- Nella fascia d'età da 81 a 116 anni il numero di volti è molto basso rispetto alla media dei volti per ogni età.

2.1.2 Distribuzione sesso

La seconda caratteristica studiata è stata quella del sesso. In *Figura 3* viene mostrato il grafico rappresentante il numero di volti per ciascun sesso.

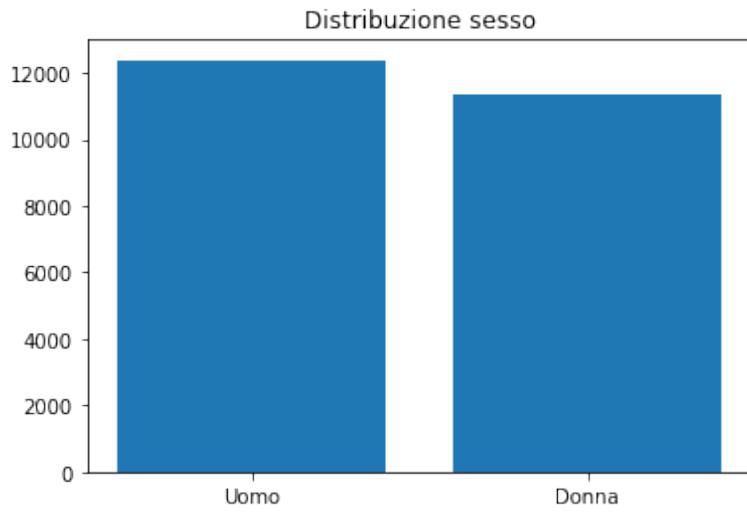


Figura 3 Grafico Distribuzione sesso.

Come si può notare osservando il grafico mostrato in *Figura 3* la distribuzione di immagini per i due sessi risulta essere piuttosto equilibrata. Nel dataset sono presenti infatti **12379** volti di uomini e **11316** volti di donne.

2.1.3 Distribuzione etnia

Infine, l'ultima caratteristica analizzata è stata l'etnia. Il dataset presentava 5 differenti tipologie di etnia:

- Occidentale;
- Africana;
- Asiatica orientale;
- Asiatica centro – meridionale;
- Altro.

Di seguito è riportato il grafico che mostra il numero di volti per ciascuna etnia.

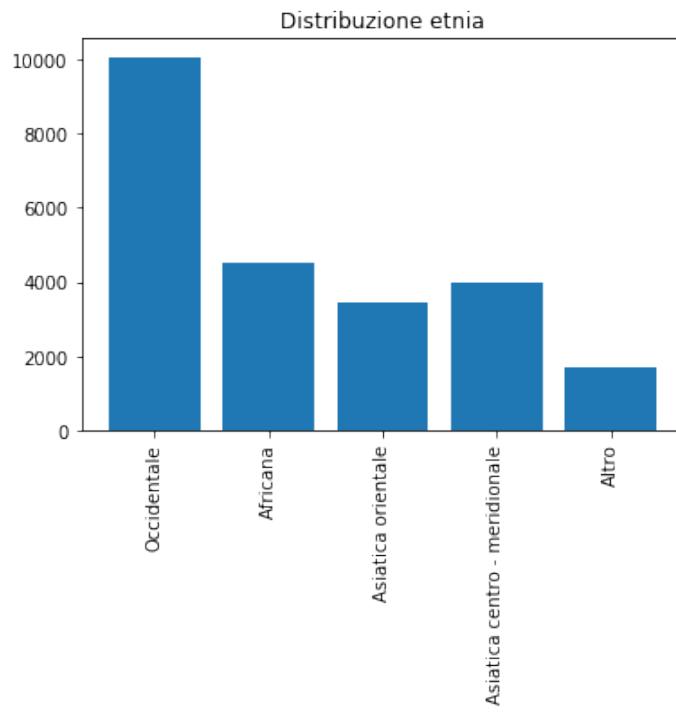


Figura 4 Distribuzione etnia.

Osservando il grafico mostrato in *Figura 4* si nota che la maggior parte dei volti ricade nell'etnia “Occidentale”, mentre il numero di volti presenti per le restanti etnie è molto inferiore. In particolare, si hanno a disposizione **10070** volti per l'etnia Occidentale, **4526** per l'etnia Africana e, rispettivamente, **3431** e **3976** per l'etnia Asiatica orientale e Asiatica centro-meridionale. Inoltre, sono presenti **1692** volti che non fanno parte delle etnie precedenti, ma di altre etnie come quella ispanica, latina o mediorientale.

3. Lavori precedenti

In questa sezione si discuterà di alcuni lavori¹ già svolti nell'ambito dell'age regression, basati sul dataset UTKFace per l'addestramento dei loro modelli. Di essi, si analizzeranno i modelli realizzati e i risultati ottenuti.

Gli obiettivi di tali lavori erano dupli, in quanto oltre alla stima dell'età di un soggetto, essi si ponevano anche l'obiettivo di stimare il suo sesso. I risultati ottenuti a quest'ultimo fine sono stati tralasciati nella successiva analisi, non essendo di interesse ai fini del progetto sviluppato.

I modelli sono stati costruiti con Keras utilizzando le Convolutional Neural Networks (CNN). Una rete neurale convoluzionale è uno speciale tipo di Deep Neural Network (DNN) che presenta ottimi risultati allo scopo di classificare immagini. Una CNN è essenzialmente costituita da un blocco di **input**, uno o più blocchi nascosti (**hidden layer**), che effettuano calcoli tramite funzioni di attivazione (e.g. **ReLU**) e un blocco di **output** che effettua la classificazione/regressione.

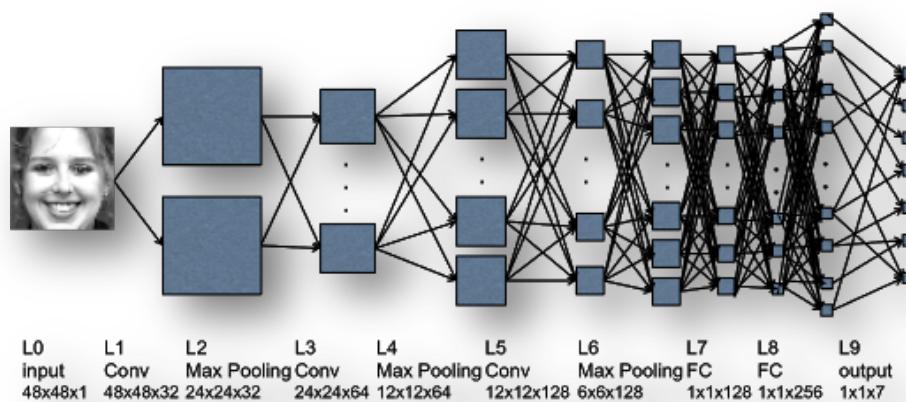


Figura 5 Struttura di una CNN.

Nella immagine in *Figura 5* è mostrata la struttura tipica di una CNN che risulta essere costituita da:

- Livelli di convoluzione;
- Livello ReLU (Rectified Linear Units);
- Livello di Pooling;
- Livello FC (o Fully connected, completamente connesso).

¹ I due lavori esaminati sono entrambi pubblicati sulla piattaforma Kaggle nella sezione kernel [1].

3.1 UTKFace - age and sex prediction [2]

Nel primo dei due lavori esaminati si è osservato che:

- Le immagini sono state lette a colori;
- Nell'estimation del sesso e dell'età la feature etnia non è stata utilizzata, pertanto si sono utilizzate soltanto le immagini;
- La CNN costruita è strutturata nel seguente modo:
 - Convolutional layer; 32 nodi, kernel size 3
 - Convolutional layer; 32 nodi, kernel size 3
 - Pooling layer; Pool size 2
 - Convolutional layer; 64 nodi, kernel size 3
 - Convolutional layer; 64 nodi, kernel size 3
 - Pooling layer; Pool size 2
 - Convolutional layer; 84 nodi, kernel size 3
 - Fully connected layer; 64 nodi
 - Fully connected layer; 1 nodi
- È stato effettuato un training con batch size uguale a 240 e un numero di epochhe pari a 2000.
- I risultati che si sono ottenuti effettuando l'addestramento su un training set composto da 15884 samples e un validation set di 7824 samples sono elencati di seguito²:
 - Loss: 4.8312
 - Val_loss: 6.7540;
- L'evaluate del modello sul test set ha fornito una loss pari a 6.0371. Quindi, l'errore medio commesso dal modello è di circa 6 anni.



Etichette:

[‘0’]: sesso.
[‘26’]: età.

```
[‘0’] [‘26’]
[array([[0.00277981]]], dtype=float32), array([[24.852373]], dtype=float32)]
```

² Sono riportati soltanto i risultati per l'attività di age prediction.

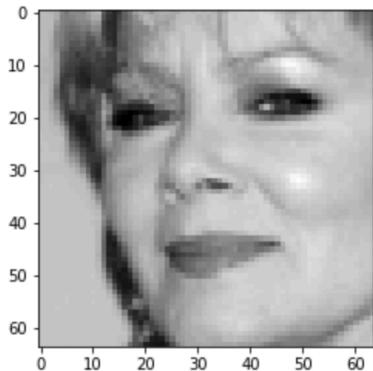
Come si può osservare, in questo esempio l'errore commesso dalla rete nello stimare l'età del soggetto risulta essere molto basso.

3.2 Age and Gender Estimation using CNN [3]

Nel secondo dei due lavori esaminati abbiamo osservato che:

- Le immagini sono state lette in scala di grigi;
- Nell'estimation del sesso e dell'età la feature etnia non è stata utilizzata, pertanto, si sono utilizzate soltanto le immagini;
- La CNN costruita è strutturata nel seguente modo:
 - Convolutional layer; 32 nodi, kernel size 3
 - Convolutional layer; 64 nodi, kernel size 3
 - Pooling layer; Pool size 2
 - Convolutional layer; 128 nodi, kernel size 3
 - Pooling layer; Pool size 2
 - Fully connected layer; 128 nodi
 - Fully connected layer; 64 nodi
 - Fully connected layer; 32 nodi
 - Fully connected layer; 1 nodi
- È stato effettuato un training con batch size uguale a 128 e un numero di epoch pari a 25.

Tramite alcuni esempi si può notare che, sebbene il modello si comporti bene per il gender prediction, non faccia lo stesso con l'age prediction.



```
Actual Gender: Female Age: 70
Values: [array([[0.6347294]], dtype=float32), array([[0.9714462]], dtype=float32)]
Predicted Gender: Female Predicted Age: Unknown
```

Figura 6 Prediction su un'immagine di esempio.

Ad esempio, osservando l'immagine in *Figura 6* si può osservare come la rete non sia riuscita a stimare l'età della persona mostrata.

Poiché per tale modello non è stata effettuata la valutazione sul test set non si dispone dell'errore medio commesso dal modello. Per cui non è possibile confrontarlo con il modello proposto discusso nel paragrafo 4.

4. Lavoro svolto

Nel seguente paragrafo verrà descritto il lavoro svolto, a partire dal caricamento del dataset fino al testing dei modelli creati.

Le valutazioni finali sui modelli creati verranno trattate nel paragrafo 5. Il codice utilizzato per effettuare le operazioni che verranno descritte di seguito è riportato nel file Jupyter Notebook *AgeReg.ipynb*.

4.1 Caricamento dataset e Pre-processing

Il primo task effettuato riguarda il caricamento delle immagini e il loro pre-processing.

La lettura delle immagini è stata effettuata tramite il metodo **imread** della libreria OpenCV. Successivamente, si è effettuato il resize (**cv2.resize**) e il reshape delle immagini in modo tale che quest'ultime avessero tutte la stessa dimensione. In particolare, la dimensione scelta è stata $(80, 80, 1)$ dove i primi due numeri indicano l'altezza e la larghezza dell'immagine, mentre il terzo indica il numero di canali della stessa. Poiché le immagini sono state lette in scala di grigi, il terzo numero risulta essere uno. Si è reso necessario optare per il caricamento delle immagini in scala di grigi per uniformare la lettura del dataset, che presentava sia immagini in scala di grigi che a colori.

Si è poi provveduto all'estrazione delle feature "sesso" ed "etnia" oltre che delle etichette "età" dal nome dell'immagini, creando un vettore che contenesse le singole feature per ogni immagine.

Di seguito è riportato il confronto tra un'immagine originale del dataset e la relativa immagine ottenuta dopo averne effettuato il caricamento e il pre-processing.

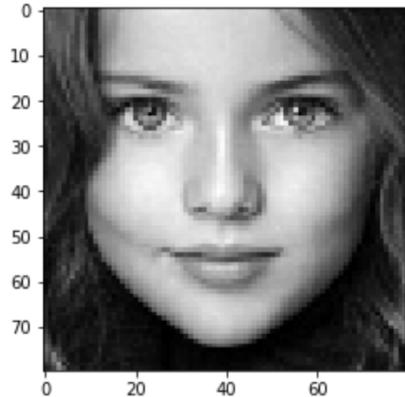


Figura 7 Sulla sinistra l'immagine originale e sulla destra l'immagine (80, 80, 1) in scala di grigi caricata.

Infine, come ultima operazione è stata effettuata una normalizzazione delle immagini dividendo i pixel di ogni immagine per 255, in modo tale che questi risultassero avere valori tra 0 e 1. Con tale operazione i valori dei pixel risultano essere più piccoli (si noti che questi valori piccoli rappresentano ancora l'immagine originale), con il beneficio che il tempo richiesto dal modello per convergere si riduce significativamente.

Inoltre, come sarà mostrato nei successivi paragrafi, il lavoro non si è limitato alla creazione di modelli che avessero come input immagini in scala di grigi. Infatti, ne sono stati creati altri che considerassero immagini a colori, nel qual caso la dimensione delle immagini è risultata essere (80, 80, 3), ad indicare che le immagini avessero tre diversi canali (RGB).

Precedente alla creazione del modello, è stata l'operazione di splitting dei dati in training set e test set, ottenuto tramite la funzione **train_test_split** di Scikit Learn. Si è deciso di utilizzare l'80% (18956 immagini) dei dati come training set e il restante 20% (4739 immagini) come test set.

Inoltre, dopo aver effettuato lo splitting, le immagini dei due set sono state salvate nelle rispettive cartelle “Training” e “Test”, in modo tale da prevederne un rapido caricamento nelle successive esecuzioni.

4.2 Caricamento modello

Il modello di training utilizzato per il progetto è rappresentato da una CNN, basata sulla libreria Keras. La scelta è ricaduta sulla tipologia di rete CNN in quanto essa viene spesso applicata alla analisi di immagini, su cui il progetto di age regression si basa. La struttura finale della rete è il risultato di numerose prove, volte ad aumentare le performance della rete. In particolare, oltre a variare le features utilizzate, si sono considerati anche diversi numeri di epoche, batch size, livelli convoluzionali e percentuali di dropout tra le principali. Si rimanda al paragrafo 6 per i dettagli sui modelli testati.

Il modello finale è costituito da:

Input Layer	Input: immagini 80 x 80 in scala di grigi (1 canale)
Convolutional layer (CL 1)	Output filters: 32 Kernel size: 3 x 3 Activation: ReLU
Convolutional layer (CL 2)	Output filters: 64 Kernel size: 3 x 3 Activation: ReLU
Pooling layer (PL 1)	Pool size: 2 x 2
Convolutional layer (CL 3)	Output filters: 128 Kernel size: 3 x 3 Activation: ReLU
Pooling layer (PL 2)	Pool size: 2 x 2
Dropout Layer	Percentage: 15 %
Flatten Layer	
Fully connected layer (FCL 1)	Output: 128 nodi Activation: ReLU
Fully connected layer (FCL 2)	Output: 64 nodi Activation: ReLU

Fully connected layer (FCL 3)	Output: 32 nodi Activation: ReLU
Fully connected layer (FCL 4)	Output: 1 nodo Activation: ReLU

Tabella 1 Struttura del modello con le sole immagini

Layer (type)	Output Shape
input_1 (InputLayer)	(None, 80, 80, 1)
conv2d_1 (Conv2D)	(None, 78, 78, 32)
conv2d_2 (Conv2D)	(None, 76, 76, 64)
max_pooling2d_1 (MaxPooling2D)	(None, 38, 38, 64)
conv2d_3 (Conv2D)	(None, 36, 36, 128)
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)
dropout_1 (Dropout)	(None, 18, 18, 128)
flatten_1 (Flatten)	(None, 41472)
dense_1 (Dense)	(None, 128)
dense_2 (Dense)	(None, 64)
dense_3 (Dense)	(None, 32)
dense_4 (Dense)	(None, 1)

Figura 8 Struttura del modello creato

Tale modello considera come input le sole immagini. Volendo considerare anche altre features, come il sesso o l'etnia, si rende necessaria una modifica alla rete. Dopo il livello di Flatten, infatti, viene aggiunto un nuovo livello, che permette di concatenare le immagini con le features, appositamente inserite in un nuovo livello di Input separato.

Flatten Layer	
Concatenate Layer	
Fully connected layer (FCL 1)	Output: 128 nodi Activation: ReLU
Fully connected layer (FCL 2)	Output: 64 nodi Activation: ReLU
Fully connected layer (FCL 3)	Output: 32 nodi Activation: ReLU
Fully connected layer (FCL 4)	Output: 1 nodo Activation: ReLU

Tabelle 2 Struttura del modello con il Concatenate Layer.

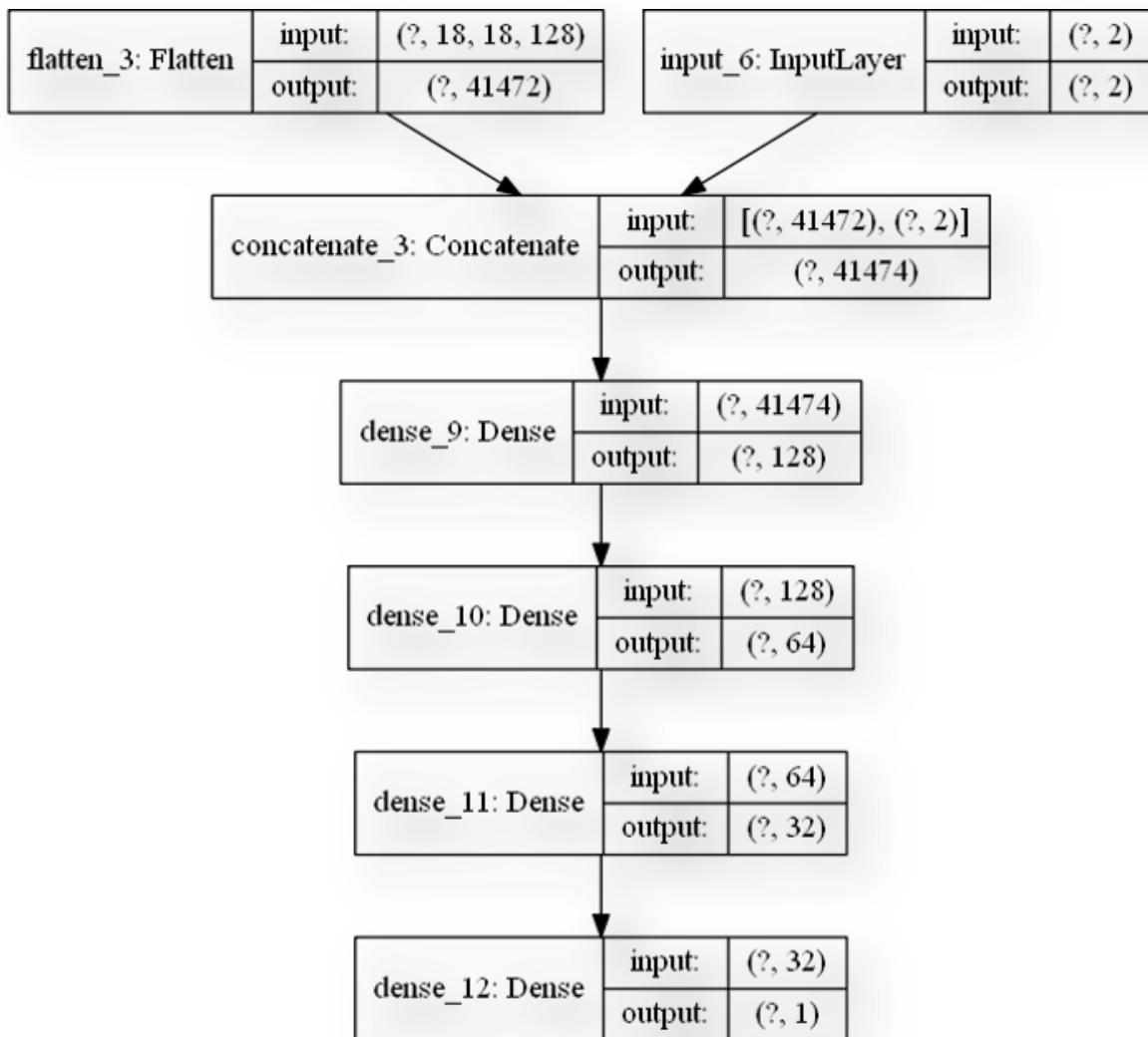


Figura 9 Esempio di concatenazione delle features

Una volta definita la struttura del modello, questo si può creare tramite la funzione **Model** di TensorFlow, che permette successivamente di compilarlo. Per i parametri della compilazione, si è deciso di optare per l'ottimizzatore Adam (computazionalmente molto efficiente), mentre si è scelta la Mean Absolute Error (MAE) come loss function. La MAE pesa tutti gli errori commessi sulla stessa scala lineare, per cui i valori outlier non avranno un peso maggiore rispetto ad altri.

4.3 Training modello

Per ottenere la migliore configurazione della rete, il training del modello è stato effettuato combinando simulazioni delle operazioni di Grid Search e Cross Validation. In primo luogo, sono state definite due diverse liste indicanti rispettivamente diversi valori di batch size e numero di epoche. Per ogni combinazione di valori, poi, è stato effettuato il training del modello tramite la funzione `fit` di Keras, utilizzando una percentuale del training set come validation set. Quest'ultimo cambia in modo randomico ad ogni epoca, permettendo così di considerare diversi dati su cui valutare il modello ed ottenere quindi una misura più precisa della bontà dello stesso. All'esaurimento delle configurazioni, è stato salvato il modello migliore (quello con la loss sul training minore). I risultati da questo ottenuti, come le accuracy e le loss sul training e validation set sono stati salvati in un file testuale specifico per il modello. Inoltre, sono stati creati dei grafici raffiguranti l'andamento della loss e dell'accuracy all'aumentare del numero delle epoche.

Il miglior modello è risultato quello avente 500 epoche e 128 elementi come batch size. Nonostante ciò, si è deciso di utilizzare come modello finale un modello da 200 epoche, che ha ottenuto risultati migliori sul test set. Si rimanda al paragrafo 6 per il confronto dettagliato tra i modelli.

La *Tabella 3* rappresenta il confronto tra i due modelli discussi, mostrando i risultati registrati da entrambi al termine della fase di training.

Nome	Epoche	Train loss	Train accuracy	Val loss
img_500_0.15_3C L_0.2	500	1,212	0.425	5,606
img_200_0.15_3C L_0.2	200	1,807	0.331	5,613

Tabella 3 Training modelli in scala di grigi con 200 e 500 epoche.

I risultati nella *Tabella 3* indicano che entrambi i modelli hanno commesso un errore medio di circa 5 anni sulle immagini di validation.

4.4 Testing

Una volta effettuato il training del modello, questo è stato testato sul test set tramite la funzione evaluate di Keras. I risultati ottenuti dal modello sul test set (test loss e test accuracy) sono stati aggiunti al file testuale descritto precedentemente.

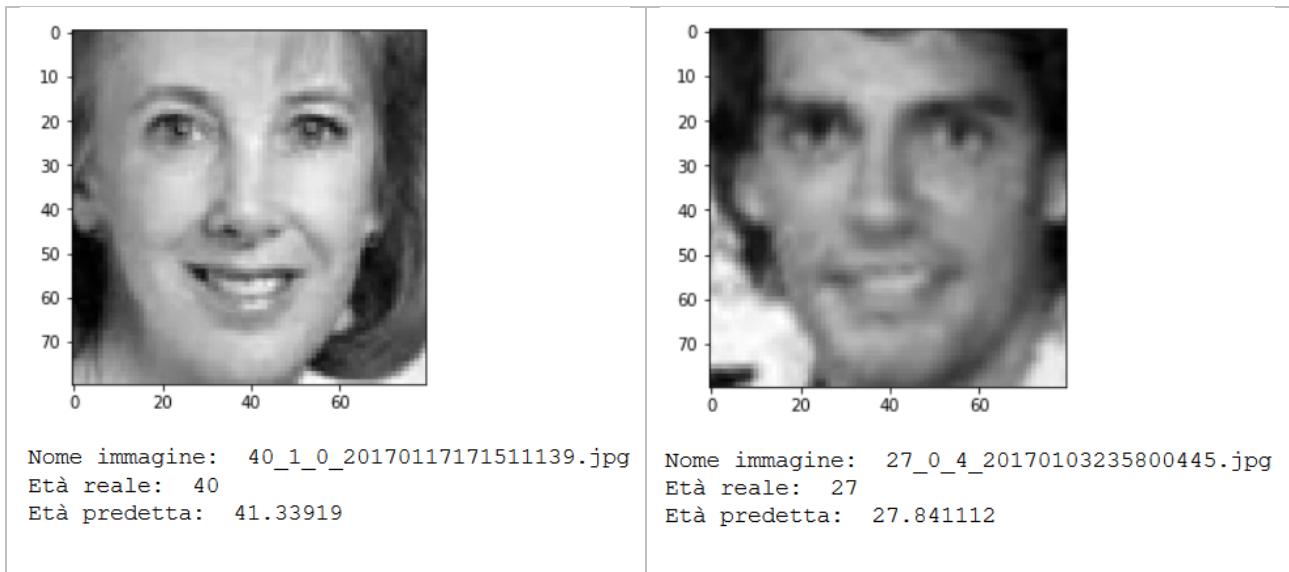
Nella *Tabella 4* sono riportati i risultati ottenuti dal modello dopo aver effettuato la fase di testing.

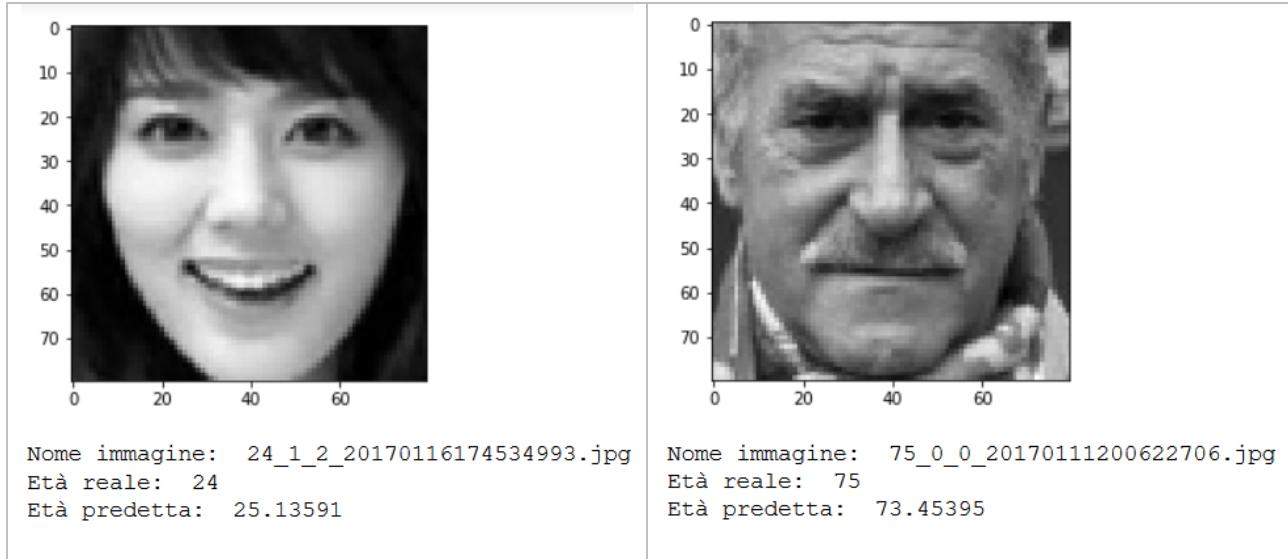
Nome	Epoche	Test loss	Test accuracy
img_200_0.15_3CL_0.2	200	5,518	0.113

Tabella 4 Risultati ottenuti sul test set per il modello da 200 epoche.

Come si può notare dai dati in *Tabella 4*, l'errore commesso dal modello sul test set è simile a quello commesso sul validation set, quindi in linea con i risultati ipotizzati.

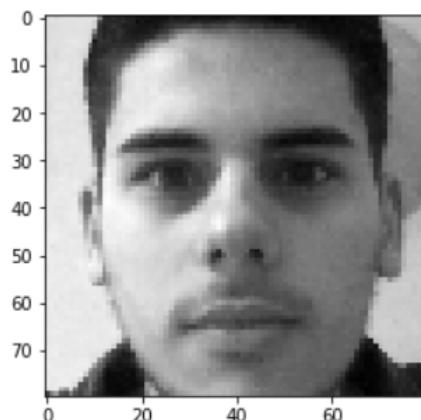
Vengono ora considerate le previsioni effettuate dal modello su alcune immagini del test set.



*Figura 10 Esempi di predictions.*

Come si può notare dai risultati, il massimo errore ottenuto dal modello è stato di due anni, nel caso dell'immagine in basso a destra.

Infine, viene mostrato il risultato ottenuto con una immagine non facente parte del test set. Tale immagine è stata acquisita tramite la webcam del laptop.



Età reale: 22
Età predetta: 26.53496

Figura 11 Prediction su esempio reale

In questo caso il modello ha avuto un errore pari a circa quattro anni, che è comunque inferiore all'errore medio ottenuto sul validation e sul test set (cinque anni)

5. Valutazioni

In tale paragrafo si discuterà delle valutazioni fatte sul modello illustrato nel paragrafo precedente, quello di 200 epoche che considera solo le immagini (in scala di grigi) e nessun'altra feature. Le valutazioni fatte per esso sono state effettuate anche per tutti gli altri modelli creati. I risultati sono riportati nel paragrafo 6 (Altri modelli).

La prima operazione effettuata è stata il caricamento del modello tramite la funzione `load_model` che restituisce il modello compilato. Tale operazione è possibile soltanto se si è provveduto al salvataggio del modello dopo averne effettuato il training. Pertanto, per ogni modello si è provveduto al suo salvataggio per poterne effettuare le successive valutazioni. Il formato utilizzato per il salvataggio del modello è il formato HDF5³.

In *Figura 12* è riportata la struttura del modello che stiamo considerando. Come si può notare, il modello accetta in input immagini con dimensione (80, 80, 1) e restituisce in output un solo valore, ovvero l'età predetta. Inoltre, una rappresentazione come quella in figura permette una comprensione più rapida dei vari passi (convoluzione, max pooling, ...) che la rete effettua e come la dimensione degli input viene modificata passo dopo passo.

³ Hierarchical Data Format versione 5 (HDF5), è un formato di file open source che supporta dati di grandi dimensioni, complessi ed eterogenei.

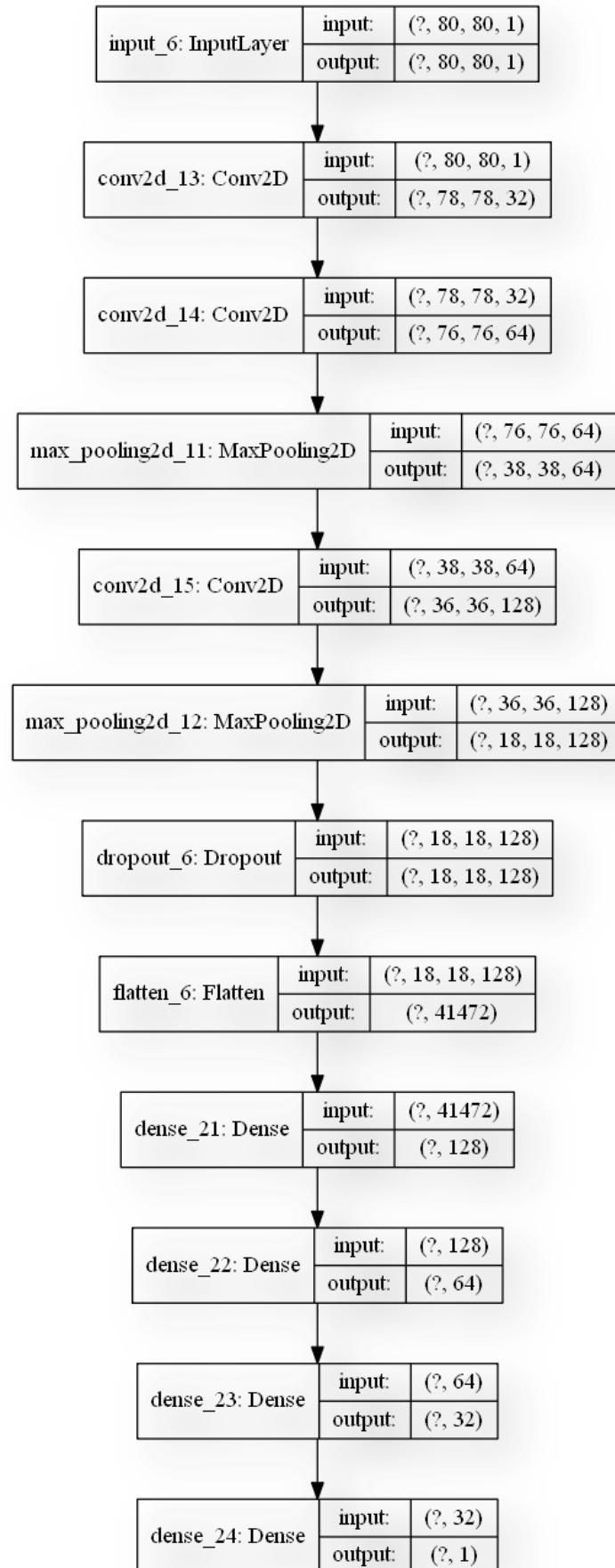


Figura 12 Struttura del modello caricato.

A questo punto si può procedere con il caricamento dell'history di training di tale modello. Questa permette di recuperare alcune informazioni riguardanti il modello come: 'accuracy', 'loss', 'val_accuracy' e 'val_loss'.

Anche in questo caso è fondamentale provvedere al salvataggio dell'history di un modello dopo averne effettuato il training se si desidera farne uso in un secondo momento: tale operazione è stata effettuata per tutti i modelli creati.

Di seguito sono mostrati due grafici che raffigurano l'andamento della loss e dell'accuracy durante le varie epoche.

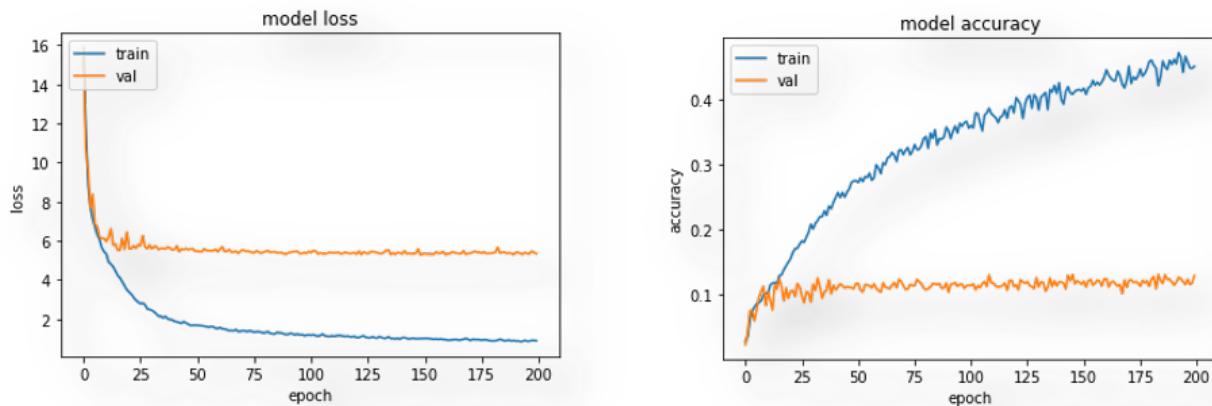


Figura 13 Grafici che descrivono l'andamento della loss (a sinistra) e dell'accuracy (a destra).

Osservando i due grafici mostrati in *Figura 13* si può notare come il modello vada in overfitting, in quanto al crescere del numero di epoche la *loss*, ma anche l'*accuracy* tendano rispettivamente a diminuire ed aumentare, mentre la *val_loss* e *val_accuracy*, ovvero la loss e l'*accuracy* registrata sul validation set tendano a stabilizzarsi. Pertanto, osservando l'andamento delle due curve mostrate nei grafici si può concludere che con un numero molto più basso di epoche avremmo ottenuto lo stesso risultato ottenuto con 200 epoche. Infatti, analizzando un modello addestrato con 80 epoche il risultato non è peggiorato.

Per cercare di eliminare o quantomeno ridurre l'overfitting si sono valutati diversi approcci come:

- cercare di livellare le istanze, cioè le immagini per ogni fascia d'età;
- trasferire circa 4000 immagini dal test set al training set;
- creare modelli con diverse configurazioni di dropout e di layer convoluzionali;
- ridurre e aumentare la dimensione delle immagini in input;
- creare modelli che considerassero anche altre feature (sesso ed etnia) oltre alle sole immagini.

Per ognuno degli approcci sopra elencati si sono addestrati dei modelli ad hoc, ma nessuno di essi ha portato ad una riduzione dell'overfitting e ad un miglioramento delle performance generali sulle prediction.

Dopo aver osservato come il modello si comporta sul training set all'aumentare del numero di epoche, viene analizzato come quest'ultimo si comporta con le immagini del test set.

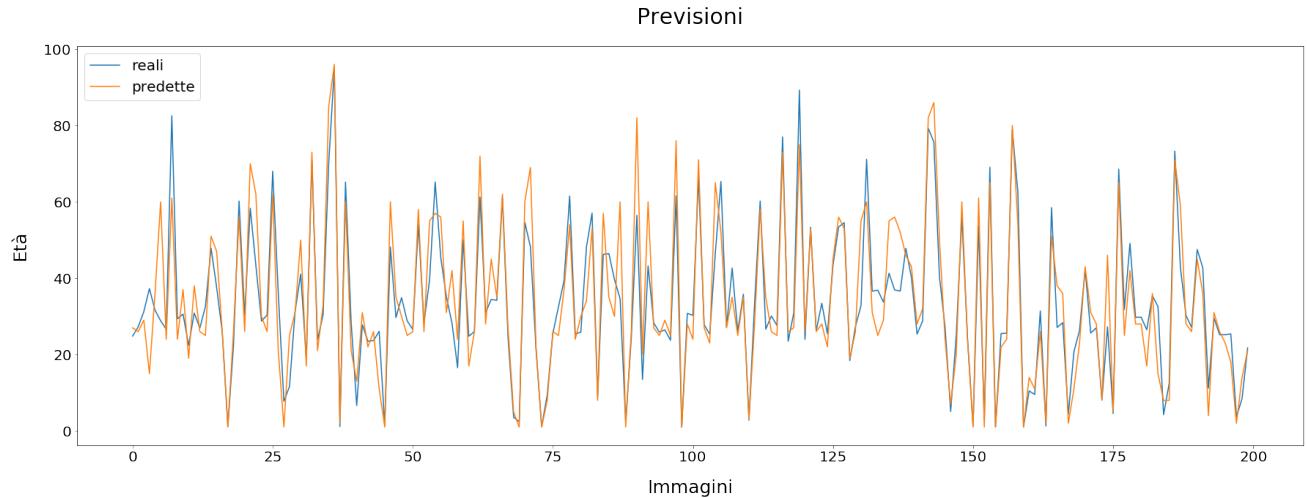


Figura 14 Età predette vs Età reali su 200 immagini di test.

Il grafico mostrato di seguito mette in relazione le prediction fatte dal modello con le età reali. Per rendere più agevole la lettura del grafico sono state considerate le prediction su soltanto 200 immagini delle oltre 4000 del test set.

Nel grafico in *Figura 14* sull'asse delle y sono riportate l'età, mentre sull'asse delle x sono riportate le immagini analizzate. Attraverso tale grafico, si può notare come in molti casi l'età predetta vada a sovrapporsi perfettamente con l'età reale, mentre in molti altri risulta non coincidere affatto.

Inoltre, per avere una visione degli errori commessi su tutte le immagini è stato realizzato il grafico in *Figura 15*. Per errore si intende la differenza in valore assoluto tra l'età reale e l'età predetta.

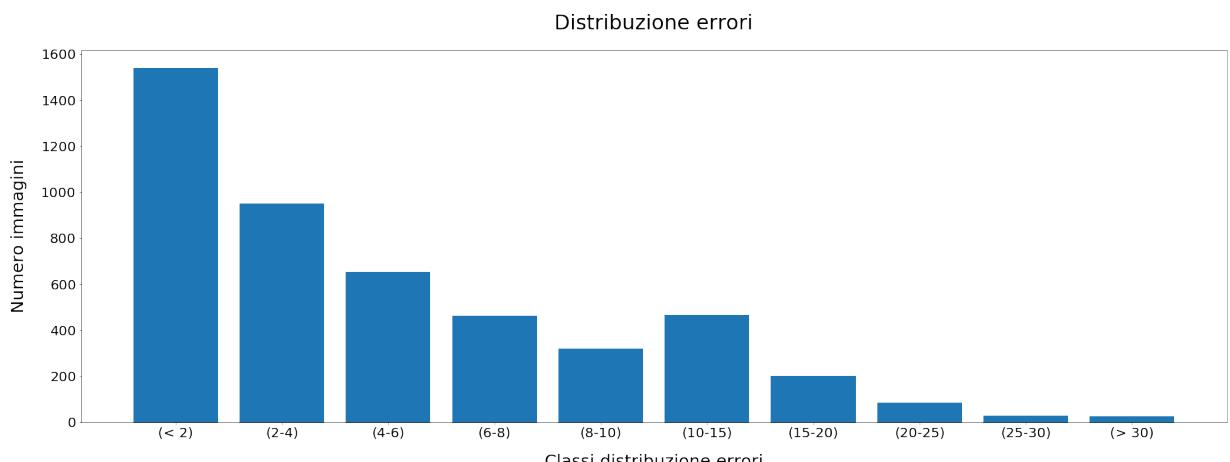


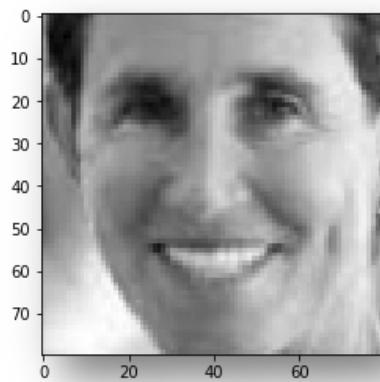
Figura 15 Errore (in anni) sulle previsioni fatte su tutte le immagini di test.

Le classi di distribuzione degli errori sono state strutturate per indicare la quantità di errore commessa dal modello (espressa in anni). Ad esempio, per circa 1500 immagini di test si è registrato un errore inferiore ai 2 anni.

Sull'asse delle x della *Figura 15* sono mostrate le classi di distribuzione degli errori, mentre sull'asse delle y sono riportate le relative occorrenze.

Si può notare come in alcuni casi (25 immagini) l'errore risulta essere molto alto, addirittura superiore ai 30 anni. Per comprendere il motivo per cui su tali immagini la rete avesse commesso un errore così alto, queste sono state analizzate singolarmente, per verificare che non presentassero anomalie rispetto alle altre immagini del test set.

Di seguito è riportata l'immagine sulla quale la rete ha commesso l'errore più alto.



Nome immagine: 5_1_0_20170104202352685.jpg.chip.jpg

Età reale: 5

Età predetta: 54.228626

Errore (in anni): 49.2286262512207

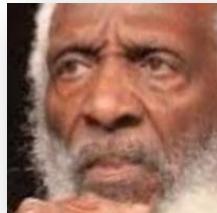
In questo caso, il modello ha associato a tale immagine un'età di circa 54 anni, mentre l'etichetta dell'immagine è pari a 5 anni. Chiaramente, immagini di questo tipo (immagini etichettate in modo errato dai creatori del dataset) se utilizzate nel training set possono influenzare le prestazioni del modello.

Purtroppo, analizzando manualmente le immagini con etichetta da 1 a 5 anni si è riscontrato che l'immagine riportata non è un caso isolato.

Di seguito sono riportate alcune delle immagini trovate per la quale l'etichetta non coincide con l'età reale del volto. Sotto ad ogni immagine è riportata la rispettiva etichetta.



5 anni



5 anni



1 anno



1 anno

L'errore medio commesso dal modello su 200 immagini è di **5.66** anni, mentre calcolando l'errore medio sull'intero test set (4739 immagini) si ha un errore pari a **5.51**.

Chiaramente, la media risulta essere influenzata da outlier come quelli analizzati sopra e pertanto, si è deciso di calcolare l'errore mediano, poiché la mediana non risulta essere influenzata da valori estremamente alti o bassi.

L'errore mediano su tutte le immagini di test è pari a **3.64** anni. La mediana risulta essere più bassa della media, che risente degli outlier.

Infine, di seguito è riportata una tabella che riassume le valutazioni fatte per i seguenti modelli:

- **img_200_0.15_3CL_0.2**: modello analizzato precedentemente, ovvero un modello che considera soltanto le immagini (in scala di grigi).
- **RGBimg_200_0.15_3CL_0.2**: a differenza del precedente, questo modello prende in input immagini a colori.
- **RGBsex_200_0.15_3CL_0.2**: modello che considera non solo le immagini (a colori), ma anche la feature sesso.
- **RGBall_200_0.15_3CL_0.2**: modello che oltre alle immagini (a colori) considera anche tutte le altre feature (sesso ed etnia).

La tabella inoltre, contiene informazioni su:

- **Train loss**: loss media del training set;
- **Train accuracy**: accuracy media del training set;
- **Val loss**: loss media del validation set;
- **Test accuracy**: accuracy media del test set;
- **Test loss**: loss media del test set;
- **Median error**: errore mediano del test set.

Nome modello	Train loss	Train accuracy	Val loss	Test accuracy	Test loss	Median error
img_200_0.15_3CL_0.2	1.807	0.331	5.613	0.113	5.518	3.640
RGBimg_200_0.15_3CL_0.2	1.954	0.295	5.685	0.106	5.567	3.700
RGBall_200_0.15_3CL_0.2	1.901	0.293	5.683	0.101	5.646	3.779
RGBsex_200_0.15_3CL_0.2	1.896	0.303	5.675	0.107	5.520	3.728

Tabella 5 Riepilogo dei risultati.

Da notare che numero il numero di epoche, la percentuale di dropout e il numero di layer con voluzionali sono identici per tutti e quattro i modelli considerati.

Inoltre, tutti i modelli sopra elencati sono stati utilizzati nella realizzazione della demo: per ulteriori approfondimenti si rimanda al paragrafo 7 (Descrizione demo).

6. Altri modelli

In questo paragrafo è riportata una tabella riepilogativa delle diverse configurazioni di modelli creati e dei risultati ottenuti su ciascuno di essi. La seguente tabella è stata estratta dal file *Sommario modelli creati.xlsx*.

Sul file Excel, per agevolare la lettura dei risultati è stata fornita la possibilità di ordinare i modelli in base ad uno dei valori di interesse, come Test loss, Test accuracy,

Nome file	Features utilizzate			Immagini			Struttura rete				Risultati				
	Immagini	Sesso	Etnia	Scala di grigi	Colori	Numero epoche	% Dropout	Layer convoluzionali	% Validation	% Test set	Train loss	Train accuracy	Val loss	Test loss	Test accuracy
img_80_0.15_3ClInv_0.2	✓			✓		80	15	3	10	20	4.202	0.153	6.534	6.052	0.090
img_80_0.15_2Cl_0.2	✓			✓		80	15	2	10	20	3.900	0.160	6.440	5.986	0.093
sex_50_0.15_3Cl_0.2	✓		✓	✓		50	15	3	10	20	4.204	0.148	6.395	5.870	0.092
img_50_0.15_3Cl_0.2	✓			✓		50	15	3	10	20	4.198	0.134	6.374	5.910	0.080
all_50_0.15_3Cl_0.2	✓	✓	✓	✓		50	15	3	10	20	3.975	0.155	6.262	5.749	0.099
img_80_0.15_3Cl_0.2_0.025	✓			✓		80	15	3	2.5	20	3.164	0.197	6.255	5.759	0.098
img_80_0.15_3Cl_0.2	✓			✓		80	0	3	10	20	2.852	0.247	6.220	6.165	0.091
img_80_0.25_3Cl_0.2	✓			✓		80	25	3	10	20	3.731	0.155	6.183	5.970	0.102
RGBimg_80_0.15_3Cl_0.2_reduced	✓				✓	80	15	3	10	20	3.119	0.194	6.175	5.719	0.100
img_80_0.15_3Cl_0.2_validation_data	✓			✓		80	15	3	10	20	2.939	0.211	6.155	5.681	0.104
sex_80_0.15_3Cl_0.2	✓	✓	✓	✓		80	15	3	10	20	3.263	0.199	6.103	5.814	0.106
img_80_0.10_3Cl_0.2	✓			✓		80	10	3	10	20	3.005	0.201	6.073	5.892	0.094
all_80_0.15_3Cl_0.2	✓	✓	✓	✓		80	15	3	10	20	3.124	0.203	6.072	5.770	0.112
img_80_0.15_4Cl_0.2	✓			✓		80	15	4	10	20	3.107	0.198	6.047	5.964	0.101
img_80_0.15_3Cl_0.2	✓			✓		80	15	3	10	20	2.988	0.214	5.913	5.648	0.105
img_80_0.15_3Cl_0.01	✓			✓		80	15	3	10	1	2.916	0.216	5.820	5.350	0.135
RGBimg_200_0.15_3Cl_0.2	✓				✓	200	15	3	10	20	1.954	0.295	5.685	5.567	0.106
RGBall_200_0.15_3Cl_0.2	✓	✓	✓	✓		200	15	3	10	20	1.901	0.293	5.683	5.646	0.101
RGBSex_200_0.15_3Cl_0.2	✓	✓	✓		✓	200	15	3	10	20	1.896	0.303	5.675	5.520	0.107
img_200_0.15_3Cl_0.2	✓			✓		200	15	3	10	20	1.807	0.331	5.613	5.518	0.113
img_500_0.15_3Cl_0.2	✓			✓		500	15	3	10	20	1.212	0.425	5.606	5.764	0.092

Figura 16 Tabella riassuntiva dei modelli creati.

Osservando la tabella, si può notare che il miglior risultato sul test size è dato dal modello **img_200_0.15_3CL_0.2**.

7. Descrizione demo

In questa sezione verrà descritta la demo realizzata per testare il modello in casi reali. L'analisi verterà principalmente sulle diverse impostazioni di test che vengono offerte nella demo. Quindi, tramite delle immagini verrà mostrata l'interfaccia della demo, mentre per la visione del codice si rimanda al file demoGUI.py.

7.1 Age prediction con una sola persona

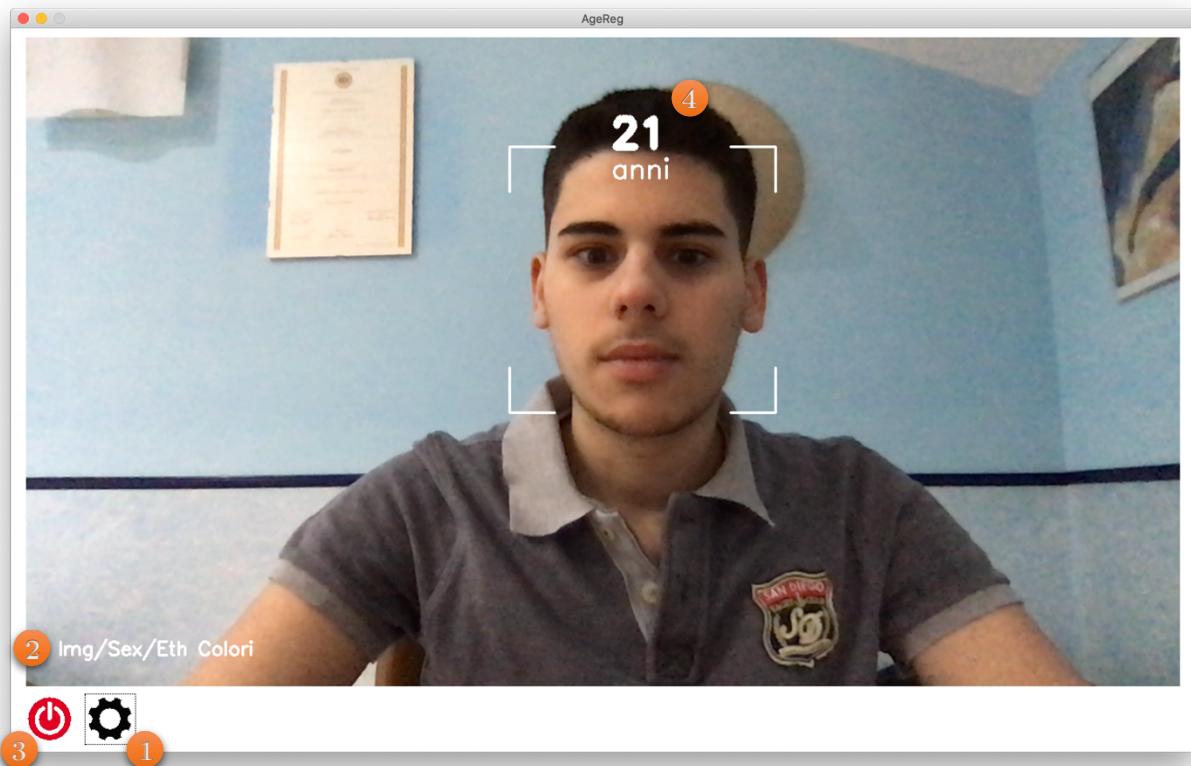


Figura 17 Prediction con un singolo soggetto rilevato.

- 1 Tasto per accedere alle impostazioni della demo descritte nel paragrafo 7.3.
- 2 Viene mostrato il nome del modello che si sta utilizzando, in questo caso il modello **Img/Sex/Eth Colori**, ovvero un modello che fa uso non solo delle immagini, ma anche del sesso e dell'etnia per effettuare la prediction dell'età.
- 3 Tasto per chiudere la demo.

- 4 Per ogni frame catturato viene effettuato la prediction dell'età, ma prima di stampare il risultato della prediction viene effettuata una media con le acquisizioni precedenti. Si è scelto di optare per una media di tutte le acquisizioni poiché il numero predetto risultava essere piuttosto variabile a seconda dell'illuminazione e dell'orientamento del volto. In questo modo, dopo qualche secondo il valore mostrato tende a stabilizzarsi rendendone più agevole la lettura.

Per quanto concerne il rilevamento dei volti si sono utilizzati classificatori a cascata basati su feature di Haar che è un metodo di rilevamento di oggetti molto efficace proposto da Paul Viola e Michael Jones nel 2001. Per ulteriori informazioni si rimanda alla documentazione di Cascade Classifier [4].

7.2 Age prediction con due persone

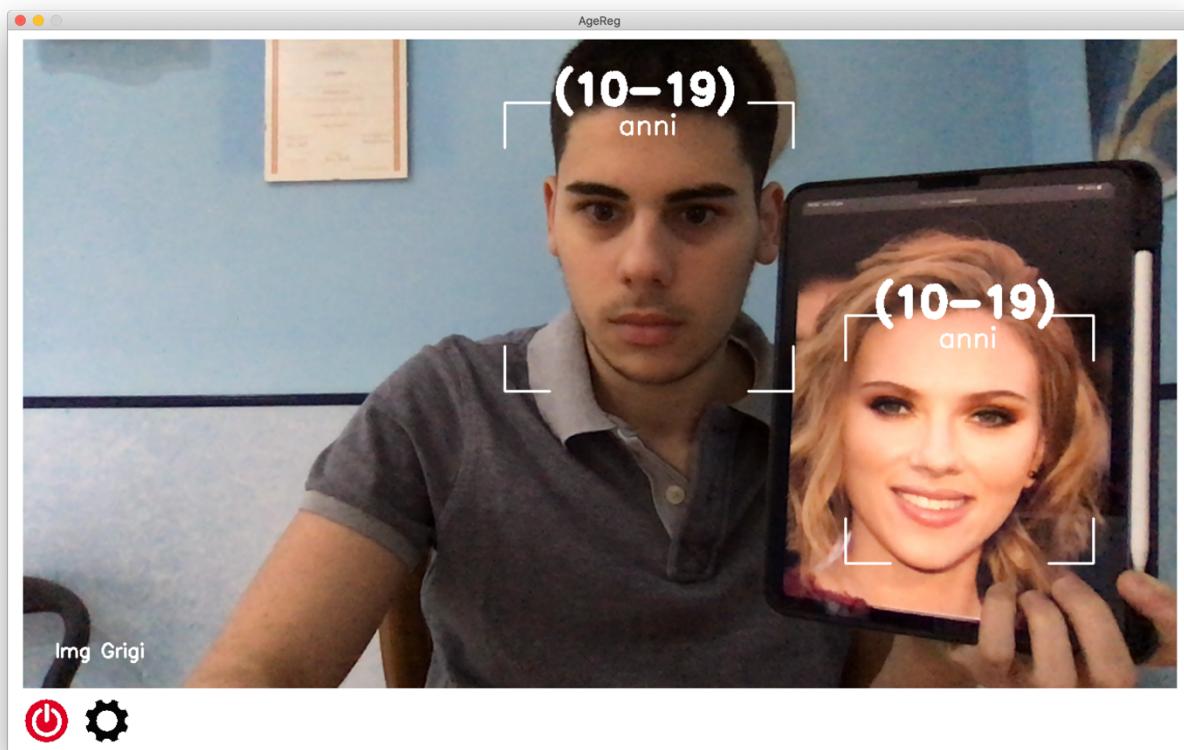


Figura 18 Prediction con due o più soggetti rilevati.

L'unica differenza rispetto al modello precedente è che in questo caso viene visualizzata una fascia d'età piuttosto che un numero.

Si è optato per la scelta di una fascia d'età poiché risulta essere più difficile associare la media ad ogni volto rilevato, in quanto il primo volto identificato in un'acquisizione potrebbe essere rilevato come secondo volto in un'acquisizione successiva e viceversa.

Un'alternativa considerata è quella di avvalersi di un ulteriore modello, che effettui un'operazione di face recognition [5], in modo tale che ogni età calcolata (calcolando la media delle varie acquisizioni che vengono fatte in real-time) sia associata univocamente ad uno degli n volti individuati. Pertanto, se venissero rilevati n volti, per ognuno di essi vi sarebbe un media rappresentante l'età di quel determinato volto.

Con un approccio di questo tipo, l'ordine con cui i volti vengono rilevati risulterebbe essere indifferente, poiché ogni soggetto (riconosciuto per mezzo di un'operazione di face recognition) avrà associata una propria media, aggiornata all'acquisizione e alla prediction dei frame.

7.3 Schermata impostazioni

Il menu impostazioni della demo realizzata è mostrato nell'immagine seguente.

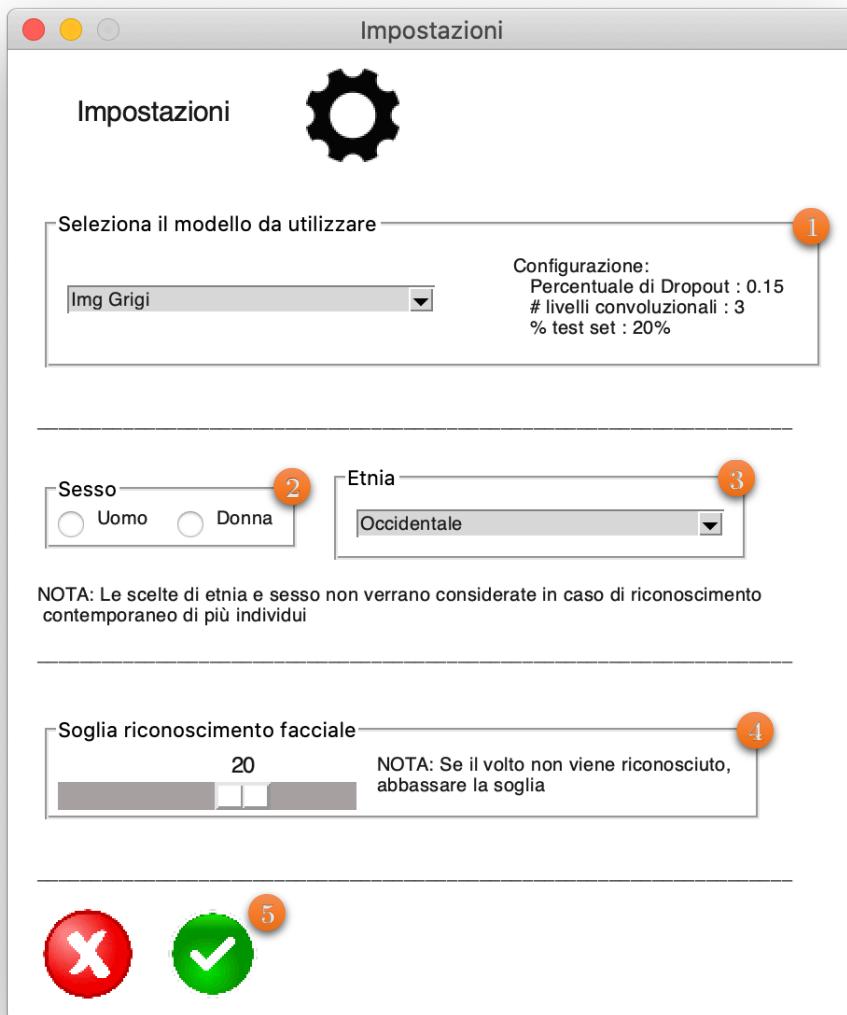


Figura 19 Impostazioni.

Con tale menu mostrato in *Figura 19* si vuole dare la possibilità di poter effettuare uno switch tra i diversi modelli addestrati e valutarne i risultati in real-time. La schermata offre le seguenti opzioni:

- 1 Possibilità di selezionare il modello da utilizzare tra una serie di modelli proposti. È possibile selezionare modelli che effettuano la prediction basandosi solo ed esclusivamente sull'immagine acquisita oppure selezionare modelli che considerano anche altre caratteristiche come il sesso e l'etnia. Inoltre, è possibile anche scegliere se si vuole testare un modello che faccia uso di immagini a colori o in scala di grigi.
- 2 Possibilità di selezionare il sesso del soggetto nel caso in cui si fosse selezionato un modello che faccia uso di tale caratteristica.
- 3 Possibilità di selezionare l'etnia del soggetto nel caso in cui si fosse selezionato un modello che faccia uso di tale caratteristica.
- 4 Possibilità di modificare la soglia del riconoscimento facciale. Se il volto non viene riconosciuto vi è la possibilità di abbassare la soglia, mentre nel caso in cui si voglia essere più selettivi ed evitare il più possibile false alarms è possibile alzare tale soglia.
- 5 Per confermare le modifiche effettuate è necessario cliccare la spunta verde.

8. Conclusioni

In questo documento è stato presentato l'approccio seguito per la realizzazione di un modello che fosse in grado di effettuare il prediction dell'età di un soggetto rilevato.

Partendo dai lavori già svolti, si è provveduto alla realizzazione di diversi modelli con l'obiettivo di trovare quello che fornisse le prestazioni migliori. Per i risultati ottenuti, questi sono stati, seppur non di molto, migliori di quelli dei lavori già svolti su tale dataset. Pertanto, considerando che il risultato è fortemente dipendente dal dataset e che per la creazione di quest'ultimo non è stata posta una particolare attenzione da parte dei creatori, il risultato registrato viene considerato piuttosto soddisfacente.

9. Riconoscimenti

Si ringraziano il professore Michele Nappi e la dottoressa Paola Barra per le risorse e il supporto fornитoci per lo svolgimento di questo progetto.

10. Riferimenti

- [1] <https://susanqq.github.io/UTKFace/>
- [2] <https://www.kaggle.com/werty12121/utkface-age-and-sex-prediction>
- [3] <https://www.kaggle.com/rahuldshetty/age-and-gender-estimation-using-cnn>
- [4] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- [5] <https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>