

A thick vertical teal bar runs down the left side of the page. A horizontal teal arrow points to the right from this bar, containing the word 'Documentazione'. In the bottom-left corner, several thin, curved teal lines sweep upwards and to the right.

Documentazione

Age Estimation

Sistemi Multimediali

Francesco Lavecchia
GIOVANNI DIMAURO

Introduzione

Il progetto realizzato stima l'età facciale di una persona partendo da un'immagine statica o da un flusso video, che può essere un file o il flusso catturato dalla webcam. Il progetto, sviluppato interamente in Python 3.8, utilizza le seguenti librerie:

- **OpenCV:** È una libreria open source utilizzata nella Computer Vision. Offre un numero elevato di algoritmi ottimizzati che hanno un'eccellente accuratezza in termini di prestazioni e velocità. Questi algoritmi possono essere utilizzati per rilevare e conoscere volti, identificare oggetti, classificare azioni umane nei video, tracciare movimenti della telecamera, tracciare oggetti in movimento, estrarre modelli 3D di oggetti, e tanto altro.
- **TensorFlow:** È una libreria open source sviluppata da Google il cui scopo è estendere l'uso del deep learning a una vasta gamma di attività. I modelli matematici utilizzati in TensorFlow sono reti neurali, che a seconda dell'architettura del livello e dei neuroni che la compongono, possono essere modellate partendo da un semplice modello di regressione fino ad un'architettura di machine learning molto più complessa.
- **PySimpleGUI:** È una libreria open source che permette di sviluppare GUI da zero in poche righe di codice. Tale GUI può essere completamente personalizzabile e il processo di realizzazione è molto intuitivo.
- **numpy:** È una libreria open source che contiene diverse funzioni e metodi utili per il calcolo scientifico, in particolar modo per il calcolo vettoriale e matriciale.

Le librerie utilizzate sono rappresentate nel file requirements.txt. È possibile installare le librerie nel proprio ambiente virtuale utilizzando il comando `pip install -r requirements.txt`.

Il modello

Per questo progetto è stato utilizzato un modello di rete neurale convoluzionale pre-addestrato, con una percentuale di accuratezza dell'83%. Il modello è stato addestrato partendo dal dataset *UTKFace*, che contiene più di 23mila immagini RGB in formato .jpg contenenti volti umani, ognuna di dimensioni 200x200. La label di ogni immagine del dataset è una stringa del tipo [età]_[sesso]_[razza]_[data&ora].jpg. Per l'addestramento di questo modello è stata utilizzata soltanto la porzione di sottostringa contenente l'età.

Il progetto utilizza il modello per risolvere un problema di classificazione dell'età. Per bilanciare ogni classe del classificatore, sono state definite 11 classi, ognuna contenente un range d'età:

- | | | |
|---------------|---------------|-----------------|
| 1. 1-2 anni | 5. 26-27 anni | 9. 46-54 anni |
| 2. 3-9 anni | 6. 28-31 anni | 10. 55-65 anni |
| 3. 10-20 anni | 7. 32-36 anni | 11. 66-116 anni |
| 4. 21-25 anni | 8. 37-45 anni | |

Questa scelta è dettata dal fatto che per ogni range d'età, nel dataset sono più o meno contenute lo stesso numero di immagini e ciò favorisce un maggior equilibrio.

Descrizione del progetto

Per la realizzazione del progetto è stata creata un'interfaccia grafica. La prima schermata in Figura 1 contiene tre Button: "Foto" per predire il range d'età dei volti presenti in un'immagine (Formati accettati: .bmp, .dib, .jpe, .jpeg, .jpg, .jp2, .pbm, .png, tutti supportati da OpenCV); "Video" per predire il range d'età dei volti presenti in un video (Formati accettati: .avi, .mp4, tutti supportati da OpenCV); "Webcam" per predire il range d'età dei volti catturati in real-time dalla webcam.

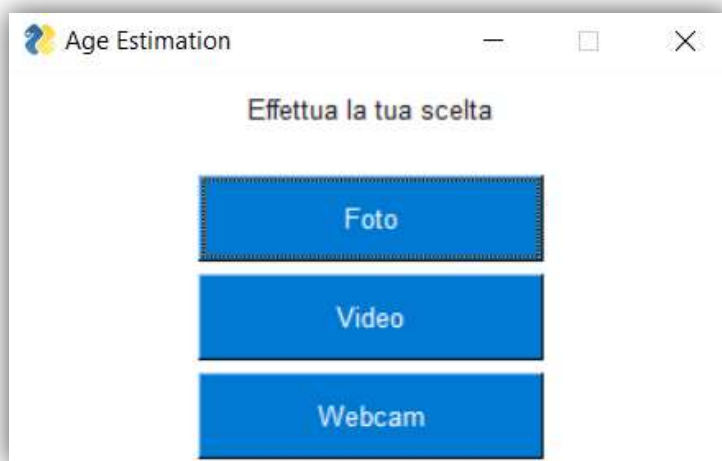


Figura 1

Cliccando su "Foto", si aprirà la schermata raffigurata in Figura 2. Il tasto "Sfoggia" è di tipo *FileBrowse*, e ha il compito di guidare l'utente nel selezionare un file immagine. I file accettati sono soltanto quelli descritti precedentemente e supportati da OpenCV. Una volta scelto il file immagine, il suo path verrà copiato all'interno della *InputBox*. Cliccando su "Seleziona" partirà il processo di predizione dell'età per l'immagine scelta. Se nessun file di tipo immagine è stato selezionato, il tasto "Seleziona" mostrerà un pop-up che avviserà l'utente di scegliere un file immagine prima di poter procedere. Il tasto "Cancella" pulirà la *InputBox* permettendo all'utente di scegliere un altro file. La stessa schermata raffigurata in Figura 2 verrà mostrata se si clicca il button "Video" raffigurato in Figura 1, con la differenza che i file selezionabili saranno file video di tipo .mp4 o .avi.

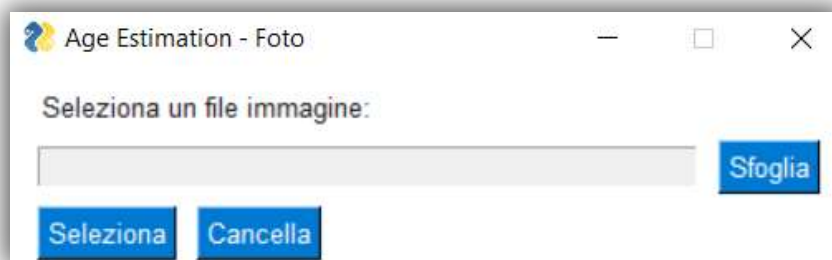


Figura 2

Cliccando sul button “Webcam” raffigurato in Figura 1, il software catturerà il flusso video della webcam, analizzando ogni frame alla ricerca di volti da cui predire l'età. Un esempio in Figura 3.

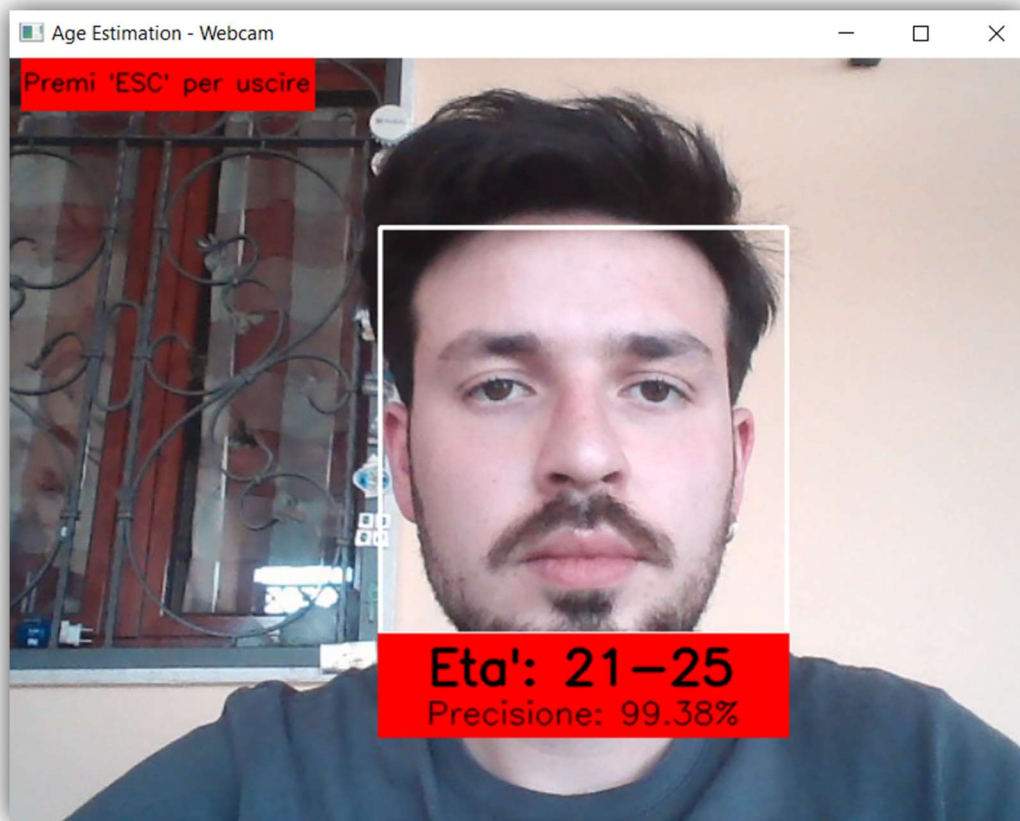


Figura 3

Per ogni funzione selezionata, dovranno essere caricate le rispettive librerie (OpenCV, TensorFlow, etc.) e questo comporta, soprattutto nei computer meno prestanti, un rallentamento delle prestazioni. Perciò è stata inserita una schermata di “Caricamento in corso...” (Figura 4) che ha come obiettivo quello di informare l'utente dell'avvio del software. Senza questa schermata, l'interfaccia verrebbe visualizzata dopo qualche secondo dall'avvio dello script.

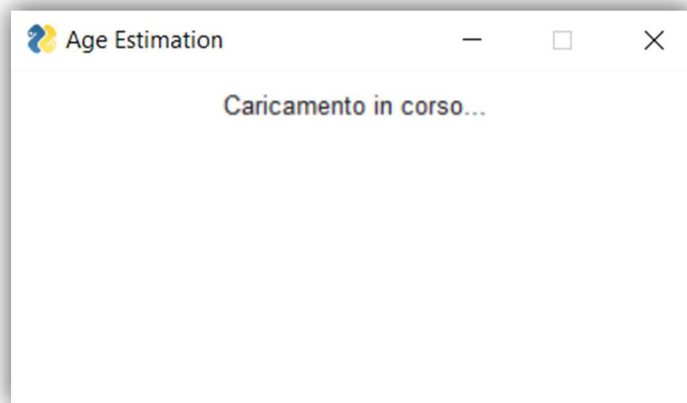


Figura 4

Descrizione dei moduli

1. setup.py

Il modulo setup.py è il motore del progetto. Definisce le undici classi del classificatore, carica il modello (*model.h5*) ed il classificatore Haar cascade di OpenCV (*haarcascade_frontalface_default.xml*), utile a rilevare i volti umani nelle immagini. Contiene inoltre la definizione di due funzioni:

1. `classify_image(image)`:
 - Effettua una copia dell'immagine ricevuta in input convertendola in Scala di Grigi;
 - Enumera i volti presenti all'interno dell'immagine fornita in input tramite l'Haar cascade;
 - Per ogni volto trovato:
 - Costruisce sull'immagine un rettangolo bianco che delinea i limiti del volto trovato;
 - Cattura la regione di interesse, ovvero il volto trovato (ROI – Region Of Interest);
 - Fornisce al modello la ROI in scala di grigi, che restituisce la classe di appartenenza e la percentuale di accuratezza;
 - Viene eseguito il metodo `create_text` fornendo in input l'immagine elaborata, la label della classe di appartenenza, la label della percentuale di accuratezza, e le coordinate del volto trovato).
 - Viene restituita in output l'immagine elaborata.
2. `create_text(image, age_text, pred_text, x, y, w, h)`:
 - Costruisce sull'immagine data in input un rettangolo di colore rosso posizionato sotto al volto trovato (ROI);
 - All'interno del rettangolo in rosso vengono inserite due label: una contenente la classe di appartenenza, che viene fornita da `age_text`, e l'altra contenente la percentuale di accuratezza, che viene fornita da `pred_text`
 - `x, y, w, h` sono rispettivamente: coordinata x della ROI, coordinata y della ROI, larghezza della ROI e altezza della ROI.

2. layouts.py

Il modulo layouts.py contiene 4 funzioni, ognuna delle quali restituisce in output il layout di una schermata.

I layout sono stati costruiti con l'utilizzo del framework PySimpleGUI, che permette una più facile implementazione delle interfacce grafiche rispetto al GUI framework standard di Python, tkinter. PySimpleGUI implementa tkinter rendendolo più semplice nell'utilizzo.

I layout costruiti in questo modulo sono:

1. `load_layout()` che restituisce il layout della schermata di caricamento.
2. `main_layout()` che restituisce il layout della schermata principale, che contiene i tre button: *Foto*, *Video* e *Webcam*.

3. `select_image_layout()` restituisce il layout della schermata di selezione di un file immagine.
4. `select_video_layout()` restituisce il layout della schermata di selezione di un file video.

Nonostante `select_image_layout` e `select_video_layout` possano sembrare uguali tra loro, in realtà non lo sono, perché ognuno dei button "Sfoglia" presente in entrambe le interfacce permette di selezionare diversi tipi di file.

3. `run.py`

Il modulo `run.py` è quello che contiene la funzione `main()` del progetto. La funzione `main()` effettua i seguenti passi:

1. Costruisce la schermata di caricamento (Figura 4) e importa il modulo `setup.py`, che a sua volta importa `numpy`, `OpenCV` e `Keras` di `TensorFlow`.
2. Una volta terminata l'importazione dei moduli, viene distrutta la schermata di caricamento e viene mostrata quella del menu principale (Figura 1). Abbiamo quindi tre button. Di seguito il comportamento di tali button alla loro pressione:
 - a. Foto
 - i. Viene mostrata la schermata di Selezione File Immagine (Figura 2).
 - ii. Una volta scelta l'immagine, alla pressione di "Seleziona", si chiuderà la schermata corrente e verrà mostrata la schermata di caricamento.
 - iii. Verrà data l'immagine scelta in pasto a `OpenCV`, la quale verrà elaborata. In particolare, verrà data in pasto alla funzione `classify_age` di `setup.py`. In seguito, all'immagine risultante verrà aggiunta una label in alto a sinistra, la quale invita l'utente a premere il tasto `ESC` per chiudere la finestra contenente l'immagine show (`OpenCV` non riconosce l'evento che viene sollevato al click della X in alto a destra della schermata). In seguito, l'immagine risultante verrà mostrata in output.
 - iv. Se l'utente preme `ESC`, la schermata si chiuderà e si tornerà al menu principale.
 - b. Video
 - i. Viene mostrata la schermata di Selezione File Video (Figura 2).
 - ii. Una volta scelto il video, alla pressione di "Seleziona" si chiuderà la schermata corrente e verrà mostrata la schermata di caricamento.
 - iii. Verrà dato il video scelto in pasto a `OpenCV`, il quale elaborerà frame per frame. In particolare, ogni frame verrà dato in input alla funzione `classify_age` di `setup.py`. In seguito, in ogni frame del video, verrà aggiunta una label in alto a sinistra, come per "Foto".
 - iv. Se l'utente preme `ESC` o se il video termina, la schermata si chiuderà e si tornerà al menu principale.
 - c. Webcam
 - i. Verrà catturato il flusso della webcam.
 - ii. Il flusso verrà elaborato, dando in pasto ogni frame a `OpenCV`. Ogni frame verrà elaborato dalla funzione `classify_age` di `setup.py`.
 - iii. Verrà aggiunta la label in alto a sinistra su ogni frame come per le due funzionalità precedenti.
 - iv. Se si preme `ESC`, la schermata si chiuderà e si tornerà al menu principale.

3. A questo punto possiamo scegliere un'altra funzione tra Foto, Video e Webcam, oppure chiudere la finestra con la X in alto a destra (Windows) o in alto a sinistra (Mac o Linux). Il processo terminerà in return 0.

4. Moduli accessori

Ci sono tre moduli accessori all'interno del progetto, che implementano le stesse funzioni dei button Foto, Video e Webcam. L'unica differenza è che questi moduli sono sviluppati con un'interfaccia a riga di comando piuttosto che grafica. Per Foto, Video e Webcam abbiamo rispettivamente: `estimate_image.py`, `estimate_video.py`, `estimate_webcam.py`.