

## Progetto 13

### Riconoscimento degli attributi facciali

P. Catapano, M. Donisi, S. Giugliano, L. Discolo

Obiettivo di questo progetto è l'analisi di un volto per determinare attributi come: colore dei capelli, presenza di occhiali, presenza della barba. Per questo progetto userete il dataset CelebA [1] costituito da circa 200k immagini di volti. Per ogni immagine è fornita l'informazione sulla presenza o meno di 40 attributi binari. Adotterete una strategia di fine-tuning utilizzando l'architettura EfficientNetB3 [2] pre-addestrata su ImageNet.

In questo progetto i passi da seguire sono:

1. **Download dei dati.** Scaricare dal sito ufficiale [1] del dataset CelebA le annotazioni degli attributi e le immagini in cui il volto già è stato centrato ritagliandolo opportunamente.
2. **Preparazione dei dati.** Dividete le immagini nei tre set di training (70%), validation (10%) e test (20%). Per ogni set create un file CSV contenente la lista dei file e per ogni file l'informazione binaria per i 40 attributi. Il file CSV deve essere utilizzato per il caricamento dai dati tramite il metodo `flow_from_dataframe` del `ImageDataGenerator` di Keras. Per tutti i set riportate le immagini nel range  $[0,1]$  e ridimensionate le immagini a  $320 \times 320$  pixel. Solo per il set di training prevedete le seguenti operazioni di *data-augmentation*: rotazione random da -10 gradi a 10 gradi, e ridimensionamento con fattore random nel range  $[0.9, 1.1]$ , traslazione verticale e orizzontale di massimo 5% dell'immagine.
3. **Architettura.** Per definire l'architettura utilizzate la funzione di Keras `EfficientNetB3` usando una dimensione di ingresso di  $320 \times 320$  pixel. Dopo lo strato di `GlobalAveragePooling2D` dovete utilizzare uno strato di `FullyConnected (Dense)` con 40 uscite con la Sigmoidale come funzione attivazione.
4. **Addestramento.** Per l'addestramento utilizzate l'ottimizzatore Adam tramite la funzione di Keras `keras.optimizers.Adam`. Mentre per la loss function adottate la Binary Cross-Entropy. Utilizzate le prestazioni sul set di validazione per selezionare i migliori valori per il learning-rate, il batch-size, il numero di epoche e il numero di strati da bloccare della rete.
5. **Valutazione delle prestazioni.** Utilizzate il test-set per valutare le prestazioni in termini di AUC (Area Under roc Curve) per ogni attributo tramite la funzione `sklearn.metrics.roc_auc_score`.



Figure 1: Esempio di volti per 4 attributi.

## References

- [1] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep Learning Face Attributes in the Wild” International Conference on Computer Vision (ICCV), 2015. <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>
- [2] M. Tan, and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” International Conference on Machine Learning (ICLR), 2019.