

Documentazione codice

Oggetto detector

L'oggetto detector viene creato affinché possano essere riconosciuti i volti nelle immagini. Viene istanziato attraverso il metodo "CascadeObjectDetector" che prende in input il file "haarcascade_frontalface_alt.xml" all'interno del quale si trova il modello di classificazione.

inizializzazione(detector, percorso, random)

I parametri in **input** rappresentano:

- **detector**: oggetto utilizzato per il riconoscimento dei volti nelle immagini
- **percorso**: stringa che contiene il percorso della cartella che contiene il dataset
- **random**: booleano che, se impostato a true, esegue uno shuffle dei file contenuti nella cartella indicata dal percorso

Funzionamento:

Memorizza l'elenco delle cartelle contenute all'interno del percorso specificato in input, e se random è true viene rimescolato l'ordine. Successivamente accede ad ogni singola cartella per analizzare le immagini contenute. Effettua un pre-processing delle immagini:

1. Converte l'immagine in scala di grigi, in modo da ridurre i canali da 3 a 1
2. Rileva il volto attraverso il detector e, se presente, si ritaglia l'immagine, altrimenti si passa all'immagine successiva
3. Riduce la dimensione in modo da avere immagini con una risoluzione di 64x64
4. Vettorizza l'immagine affinché si ottenga un vettore da 4096 componenti, e si aggiunge la riga alla matrice che conterrà tutte le immagini vettorizzate

Questi passi vengono ripetuti per un massimo di 2500 immagini.

Infine ritorna in **output** la matrice trasposta affinché abbia come dimensioni 4096xN, dove N è il numero di immagini, e ogni colonna corrisponderà a ciascuna di esse.

media(M)

Il parametro in **input** rappresenta una matrice.

Funzionamento:

Calcola la media lungo le righe della matrice in input eseguendo la formula:

$$m_i = \frac{1}{N} \sum_{j=0}^{N-1} x_{i,j} \quad i = 0, \dots, M - 1$$

dove N è il numero di colonne ed M il numero di righe.

Ritorna in **output** un vettore colonna da 4096 che rappresenta la media del dataset.

covarianza(M, media)

I parametri in **input** rappresentano:

- **M**: matrice contenente le immagini nelle colonne
- **media**: media della matrice lungo le righe

Funzionamento:

Calcola la matrice di covarianza della matrice in input sfruttando la media attraverso la formula:

$$C = \frac{1}{N} \sum_{j=0}^{N-1} (x_j - m)(x_j - m)^T$$

dove N è il numero di colonne.

Ritorna in **output** la matrice di covarianza del dataset, che avrà come dimensioni 4096x4096.

svd_c(M)

Il parametro in **input** rappresenta una matrice.

Funzionamento:

Effettua la Singular Value Decomposition della matrice in input, al fine di ottenere $M = U\Sigma V^T$, dove:

- **U** è la matrice dei vettori singolari sinistri di M, disposti nelle colonne, e si ottiene calcolando gli autovettori di MM^T
- **V** è la matrice dei vettori singolari destri di M, disposti nelle colonne, e si ottiene calcolando gli autovettori di $M^T M$
- **Σ** è la matrice che contiene i valori singolari non nulli di M, disposti lungo la diagonale principale, e si ottiene calcolando le radici quadrate degli autovalori non nulli di MM^T o $M^T M$

Ritorna in **output** il vettore S che contiene gli elementi della diagonale principale di Σ in ordine decrescente, e le matrici degli autovettori U e V che, poiché la matrice di covarianza è simmetrica, in questo caso coincideranno.

proiezione(componenti, gallery, eigenfaces, media)

I parametri in **input** rappresentano:

- **componenti**: il numero di componenti, e quindi di eigenfaces, da utilizzare per calcolare i coefficienti
- **gallery**: matrice che contiene le immagini della galleria di identità disposte lungo le colonne
- **eigenfaces**: matrice che contiene le eigenfaces disposte lungo le colonne
- **media**: vettore colonna che contiene la media del dataset

Funzionamento:

Proietta le immagini della galleria nello spazio, sfruttando la media e le eigenfaces, in modo da calcolare i coefficienti utilizzando un certo numero di componenti. Utilizza la formula:

$$a_i = (x_j - m) u_i \quad i = 0, \dots, M - 1$$

dove x_j è la j-esima immagine e M è il numero di componenti.

Ritorna in **output** una matrice che contiene nelle colonne i coefficienti a M componenti per ogni immagine della galleria.

ricostruzione(a, eigenfaces, media)

I parametri in **input** rappresentano:

- **a**: il vettore contenente i coefficienti di un'immagine
- **eigenfaces**: matrice che contiene le eigenfaces disposte lungo le colonne
- **media**: vettore colonna che contiene la media del dataset

Funzionamento:

Ricostruisce la migliore approssimazione dell'immagine utilizzando i coefficienti e sfruttando la media e le eigenfaces. Si utilizza la formula:

$$\tilde{x} = m + \sum_{i=0}^{M-1} a_i u_i$$

dove M è il numero di componenti del vettore a.

Ritorna in **output** l'immagine ricostruita vettorizzata.

acquisizione(detector)

Il parametro in **input** rappresenta l'oggetto che permette di riconoscimento dei volti nelle immagini.

Funzionamento:

Memorizza l'immagine di test a cui viene applicato lo stesso pre-processing dell'inizializzazione.

Ritorna in **output** l'immagine vettorizzata.

varianza(S, n)

I parametri in **input** rappresentano:

- **S**: vettore contenente i valori singolari non nulli della matrice di covarianza disposti in ordine decrescente
- **n**: percentuale della varianza da considerare

Funzionamento:

Scala i valori di S in modo che sommino 1, e calcola l'indice del valore singolare fino al quale la somma dei precedenti non superi la percentuale n.

Ritorna in **output** l'indice, che rappresenta il numero di componenti da utilizzare, e quindi la percentuale di varianza desiderata.

distanze(a_g, a_t)

I parametri in **input** rappresentano:

- **a_g**: matrice che contiene i coefficienti di tutte le immagini della galleria disposti nelle colonne
- **a_t**: vettore colonna che contiene i coefficienti dell'immagine di test

Funzionamento:

Calcola la distanza euclidea tra il vettore a_t e ogni vettore contenuto in a_g utilizzando la formula:

$$\|x - y\| = \sqrt{\sum_{i=0}^{M-1} |a_{x_i} - a_{y_i}|^2}$$

dove M è il numero di componenti.

Ritorna in **output** il vettore delle distanze, una per ogni immagine della galleria.

argmin(d)

Il parametro in **input** rappresenta un vettore.

Funzionamento:

Ricava l'indice del minimo elemento nel vettore.

Ritorna in **output** l'indice della distanza più piccola.

mapping(index)

Il parametro in **input** rappresenta un indice.

Funzionamento:

Mappa l'indice in 4 range, associando un numero da 1 a 4 a seconda del range di appartenenza.

Ritorna in **output** l'identità associata all'indice.