
01TUJOV IMAGE PROCESSING AND COMPUTER VISION

FACE DETECTION AND TRACKING

Durata prevista dell'esercitazione: 1,5 ore (1 settimana)

Lo scopo di questa esercitazione è quello di realizzare un sistema di riconoscimento e di tracciamento di volti in un'immagine utilizzando OpenCV.

WARM UP

Un processo di riconoscimento di oggetti (volti, in questo caso) è generalmente efficiente se basato sul rilevamento di feature che includano informazioni aggiuntive riguardo la classe di oggetti da rilevare. In questa esercitazione useremo le *Haar-like features* e i *Local Binary Patterns* (LBP) per codificare i contrasti evidenziati dal volto umano e le sue relazioni spaziali con gli altri oggetti presenti in un'immagine.

Tali feature sono tipicamente estratti utilizzando un classificatore a cascata (*Cascade Classifier*) che deve essere "addestrato" in modo da riconoscere con precisione oggetti diversi: la classificazione di volti sarà, infatti, ben diversa dalla classificazione di automobili. Ai fini di questa esercitazione utilizzeremo i classificatori già addestrati messi a disposizione da OpenCV (e presenti nella cartella "data", resa anche disponibile come file zip sul portale della didattica).

La prima operazione da fare per svolgere questa esercitazione è, pertanto, quella di visualizzare il flusso video dalla webcam del vostro computer. Potete anche utilizzare un video registrato in precedenza: l'importante è che sia presente almeno un volto.

CLASSIFICATORI DI HAAR

Usate i classificatori basati su Haar per rilevare il volto all'interno dei singoli frame del video. Nel rilevare il volto, seguitene anche il movimento. Mostrate a schermo il volto/i volti riconosciuti, applicando a quanto viene riconosciuto un rettangolo colorato.

Provate i vari classificatori di Haar per i volti frontali presenti nella cartella "data" (disponibile come zip sul portale della didattica e nel package `cv2.data.harcascades`): quello che cambia, essenzialmente, è il set di immagini con cui sono stati addestrati. Scegliete, poi, quello/i che preferite.

Suggerimenti:

- i classificatori vanno caricati/utilizzati tramite `cv2.CascadeClassifier()`;
- prima di cercare eventuali volti in un frame (col metodo `detectMultiScale()`), convertite il frame in scala di grigi ed equalizzarne l'istogramma con `equalizeHist()`, in modo da effettuare una ricerca più accurata;
- nel ricercare i volti nel frame, il metodo `detectMultiScale()` potete indicare la dimensione minima possibile del volto (oggetti di dimensione minore non saranno considerati): potete fare varie prove per la dimensione minima del volto da riconoscere, ma una dimensione tipicamente soddisfacente corrisponde al 20% dell'altezza del frame;

- il metodo `detectMultiScale()` memorizza tutti volti trovati in un frame in una tupla `(x, y, w, h)`: per “incorniciarli” con un rettangolo, occorre, per ognuno di essi, disegnare il rettangolo a schermo con `rectangle()`.

LOCAL BINARY PATTERN

Ripete quanto fatto con i classificatori di Haar per riconoscere il volto ma usando i classificatori basati su LBP: c'è differenza rispetto ai primi? Qual è più accurato? Qual è più veloce?

SCENDERE IN DETTAGLIO...

Utilizzando nuovamente i classificatori basati su Haar rilevate altre caratteristiche del volto umano: occhi, bocca, naso e orecchie.

Potete migliorare il rilevamento del volto (e delle altre caratteristiche) agendo sulla qualità del video, ad esempio riducendo il rumore?

LINK UTILI

- Tutorial su Haar Cascade in OpenCV
https://docs.opencv.org/4.5.5/db/d28/tutorial_cascade_classifier.html