
01TUJXX IMAGE PROCESSING AND COMPUTER VISION

INTRODUZIONE A OPENCV

Durata prevista dell'esercitazione: 3 ore (2 settimane)

Lo scopo di questa esercitazione è quello di familiarizzare con OpenCV in Python, apprendendo l'utilizzo di alcune funzioni fondamentali delle librerie stesse. Come ulteriore aiuto, potete utilizzare i link utili riportati al termine di questo documento.

WARM UP

Si scarichino (dalla cartella *slide/OpenCV/esempi-codice* disponibile sul portale della didattica) i file contenenti i tre progetti di esempio sull'utilizzo di OpenCV in Python, con e senza GUI.

Aiutandovi con le slide dedicate a OpenCV (sempre presenti sul portale della didattica), provate a eseguire i progetti di esempio forniti, verificando il corretto funzionamento delle applicazioni.

ESERCIZIO 1: OPERAZIONI DI BASE

Partendo dai progetti già condivisi o creandone uno nuovo, realizzate un programma che permetta di eseguire alcune *operazioni basilari sulle immagini*. In particolare:

1. Caricare un'immagine a colori (propria o con licenza Creative Commons).
2. Stampare a schermo alcuni pixel dell'immagine (a scelta).
3. Cambiare la scala di colori dell'immagine a RGB, visualizzarla e ristampare gli stessi pixel di prima. Confrontando i valori stampati in precedenza con quelli attuali, cosa si può notare?
4. Ripetere la stessa operazione del punto precedente, ma usando HSV come scala di colori.
5. Rimpiazzare ora un canale dell'immagine originale con una matrice di zeri. Qual è l'effetto macroscopico?
6. Rimpiazzare, infine, lo stesso canale di prima con una matrice i cui valori sono a 255. Qual è l'effetto macroscopico?

ESERCIZIO 2: OPERAZIONI ARITMETICHE

Partendo dai progetti già condivisi o creandone uno nuovo, realizzate un programma che permetta di eseguire alcune *operazioni aritmetiche tra immagini*.

In particolare, mostrare a schermo il flusso video ottenuto dalla webcam o da un video scaricato sul computer (possibilmente con licenza Creative Commons). Se non fosse possibile, utilizzare un'immagine statica.

Scelta poi un'immagine ("logo") le cui dimensioni siano inferiori al frame video, applicatela nell'angolo in basso a destra di ogni frame del video. Per far ciò, impostate un ROI (Region Of Interest) sui frame del video e poi "sommate" il logo all'area così delineata.

Utilizzate tre meccanismi di "somma": l'*addition*, il *blending* e l'intera sostituzione dei pixel dell'area con il logo. Ci sono differenze nel risultato?

ESERCIZIO 3: ISTOGRAMMA E SUA EQUALIZZAZIONE

Partendo dai progetti già condivisi o creandone uno nuovo, realizzate un programma che permetta di *calcolare, visualizzare e poi equalizzare l'istogramma* di un'immagine. Anche per questo esercizio potete utilizzare un'immagine statica o agire direttamente su un flusso video.

Iniziando con l'immagine/frame in scala di grigi, calcolate e visualizzate (in una finestra dedicata) l'istogramma dell'immagine tramite le funzioni messe a disposizione da OpenCV. Ripetete poi la stessa operazione con le funzioni messe a disposizione da Matplotlib.

Calcolate e mostrate, successivamente, l'istogramma per ogni canale della versione a colori, utilizzando uno dei due modi sperimentati in precedenza. L'istogramma visualizzato vi sembra corretto e coerente rispetto a quello che potreste ottenere in GIMP o simili?

Provvedete, infine, ad equalizzare l'istogramma dell'immagine e confrontarlo con l'istogramma originale. Vi aspettavate un risultato simile?

LINK UTILI

- Operazioni di base sulle immagini
https://docs.opencv.org/4.5.5/d3/df2/tutorial_py_basic_ops.html
- Operazioni aritmetiche sulle immagini
https://docs.opencv.org/4.5.5/d0/d86/tutorial_py_image_arithmetics.html
- Visualizzazione di istogrammi
https://docs.opencv.org/4.5.5/d1/db7/tutorial_py_histogram_begins.html
- Equalizzazione di istogrammi
https://docs.opencv.org/4.5.5/d5/daf/tutorial_py_histogram_equalization.html
- Documentazione OpenCV (3.4.2) in Python <https://madebyollin.github.io/opencv-python-docs/>