

Elaborazione delle Immagini

Laboratorio 8

Obiettivi:

- Classificazione di immagini a colori

Ricordate: per processare le immagini è sempre conveniente trasformare in valori double tra 0 e 1 con **im2double**.

Ricordate: imshow visualizza le immagini in modo corretto se hanno valori tra 0 e 255 (uchar8), se hanno valori tra 0 e 1 (double) o sono valori logici.

Ricordate: se volete saperne di più sulle funzioni Matlab usate, consultate l'help o la documentazione con i seguenti comandi da console:
help <funzione>
doc <funzione>

Scrivete il codice di ogni esercizio in uno script separato (labX_1.m, labX_2.m, ...)

(1)

L'obiettivo è quello di classificare le 1000 immagini (**simplicity.zip**), che compongono il dataset, nelle 10 classi: tribal, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, food.



Le immagini sono a colori. Sono dati due file di testo: **images.list** e **labels.list**. Questi file contengono la lista delle immagini del dataset e, per ognuna, la rispettiva etichetta della classe.

Per poter classificare queste immagini è necessario un pre-processing per creare i set di immagini da usare per il training e il set di immagini da usare per testare il classificatore creato.

- A1. Completate la scrittura della funzione **create_descriptor_files**, aggiungendo il codice necessario per calcolare i descrittori su ciascuna immagine. I descrittori devono essere memorizzati in array separati: un array per tipo di descrittore con un descrittore per riga. La funzione deve anche salvare i singoli array in file separati.

Potete implementare i vostri descrittori e/o usare le funzioni disponibili nell'archivio. Le tre funzioni sono:

compute_lbp: Calcola l'istogramma dei Local Binary Patterns (59 valori)

compute_CEDD: Calcola il Color and Edge Directivity Descriptor (144 valori)

compute_qhist: Calcola l'istogramma dei colori RGB quantizzati a 16 livelli ciascuno per un totale di 4096 possibili combinazioni.

- B1. Completate la scrittura della funzione **test_classifier** scegliendo un classificatore con i relativi parametri.
- C1. In uno script caricate in memoria tutti i descrittori salvati in precedenza. Caricate anche le etichette in **labels** con la funzione **redlists**.
- D1. Usando la funzione `cvpartition` create il partizionamento dei dati nei due insiemi: uno di training e di test. Memorizzate il partizionamento nella variabile **cv**.

- E1. Usate la funzione **test_classifier** passando diversi descrittori, le etichette **labels** e la struttura **cv**. Analizzate i risultati dei diversi test.
- F1. Testate anche diverse combinazioni (concatenazioni) dei descrittori. Analizzate i risultati anche visualizzando le accuraci dei vari descrittori su dei grafici.

Per saperne di più sul descrittore CEDD:

<http://orpheus.ee.duth.gr/anaktisi/pdfs/CEDD.pdf>