Elaborazione delle Immagini

Laboratorio 1

Obiettivi:

- Imparare a manipolare le immagini
- Applicare operazioni puntuali tra immagini
- Ricavare e comprendere l'istogramma di una immagine
- Equalizzare una immagine
- Applicare la gamma correction

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 (lab1_1.m, lab1_2.m, ...)

---

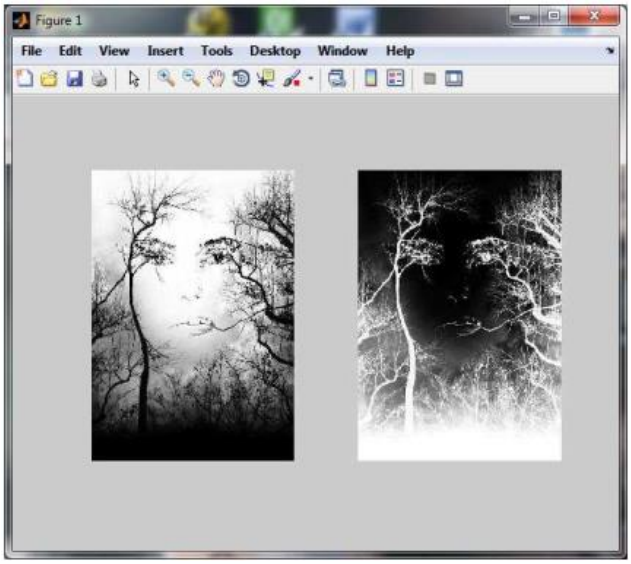
(1)

a1. Caricate l'immagine '**optical.jpg**' in una variabile **im1**.

b1. Verificate che l'immagine caricata è monocromatica e ha valori compresi tra 0 e 255 guardando le variabili nel workspace.

c1. Create una immagine **im2**, che rappresenta l'immagine negativo di **im1**.

d1. Visualizzate con **subplot** e **imshow** le due immagini in una stessa finestra.



e1. Provate a creare una immagine **im3** sommando im1 e im2. Cosa ottenete? Perchè?

---

(2)

a2. Caricate l'immagine '**moon.jpg**' in una variabile **moon** e l'immagine '**clouds.jpg**' in una variabile **clouds**.

b2. Vogliamo combinare le due immagini per ottenere una sola immagine. Per fare questo, è necessario che le due immagini abbiano la stessa dimensione. Ridimensionate una delle due immagini usando **imresize.**

c2. In una variabile **ims** mettete la somma puntuale di **moon** e **clouds**.

d2. In una variabile **imd** mettete la differenza puntuale di **moon** e **clouds**.

e2. Visualizzate i risultati con **imshow** in due finestre diverse. Cosa si nota?

La somma ha causato la saturazione sul bianco di alcuni pixel. Questo è dovuto al fatto che i pixel sono usciti dal range corretto.

La differenza ha causato la "scomparsa" di una delle due immagini. Questo è dovuto al fatto che i valori di questi pixel sono diventati negativi e imshow non visualizza valori negativi.

Infine, se le due immagini sono state tenute in uchar8, la somma e differenza di valori senza segno con range limitato dà problemi nei calcoli.

Importante: quando si manipolano le immagini, è sempre bene trasformarle in valori double tra 0 e 1 usando **im2double**

Per combinare le immagini senza uscire dal range di valori si può usare una tecnica di **blend** dei valori. Praticamente dobbiamo combinare linearmente due pixel pesandoli in modo tale da non farli uscire dal range ammesso. Una funzione di blend ha questa forma:

$$out = \alpha \times input1 + (1-\alpha) \times input2 \qquad 0 \le \alpha \le 1$$

a3. Provate ad applicare la funzione di blend alle immagini **moon** e **clouds** dell'esercizio precedente. Sperimentate con il valore di alpha. Il problema di questa tecnica è che le immagini tendono a "scurirsi". Perchè?

Un altro metodo per combinare le due immagini è quello di usare delle **maschere**. Se sommiamo le due immagini, abbiamo un problema nella regione in cui si hanno sia pixel di luna che di nuvole. L'idea è quella di fare in modo che in questa regione si abbiamo o solo pixel di nuvole o solo pixel di luna spegnendo gli altri pixel.

Ipotizziamo di spegnere i pixel di luna che si sovrappongono alle nuvole. Il seguente codice realizza la combinazione con l'uso di maschere (clouds e moon sono tra 0 e 1):

```
mask_clouds = (clouds>T); % T da scegliere in modo opportuno!!
moon_not_cloud = moon .* (1-mask_clouds);
out = moon_not_cloud + clouds;
figure,imshow(out);
```

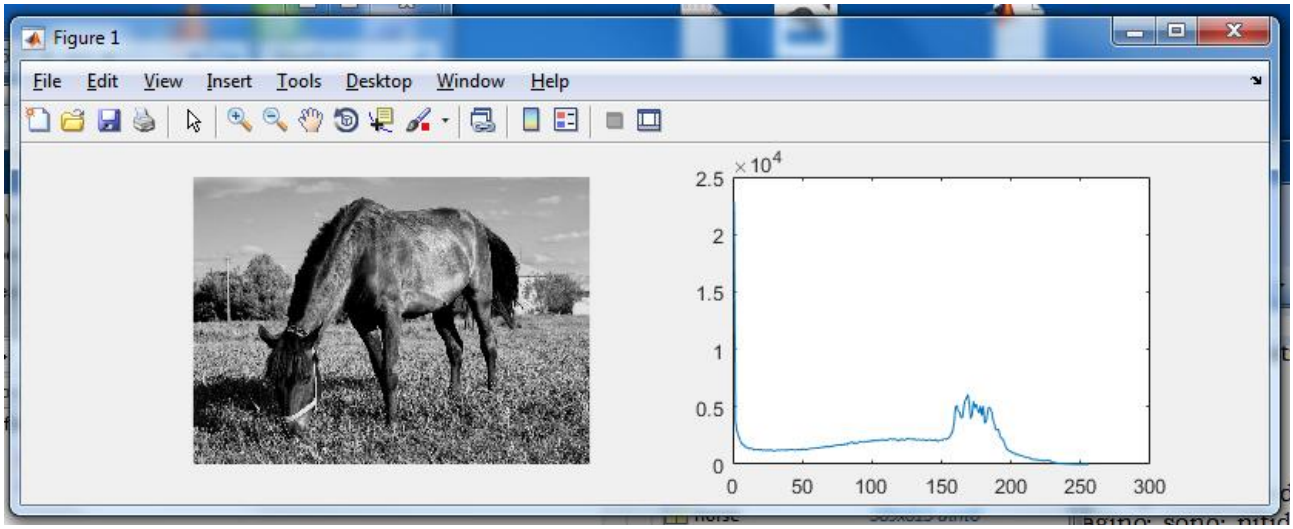
b3. Testate il codice mettendo il valore di soglia T.

c3. Analizzate il codice e cercate di capire perchè dovrebbe funzionare.

a4. Caricate l'immagine '**horse.jpg**' in una variabile **horse,** l'immagine '**nrg.jpg**' in una variabile **nrg,** e l'immagine '**family.jpg**' in una variabile **family**.

b4. usando il comando **imhist**, calcolate gli istogrammi delle tre immagini e metteteli nelle variabili **horse_hist**, **nrg_hist**, **family_hist**.

c4. Usando **subplot, imshow** per le immagini e **plot** per l'istogramma, visualizzate le immagini con a fianco il loro istogramma in tre finestre diverse. Cosa si nota?



Ricordate cosa rappresenta l'istogramma. La forma dell'istogramma ci dà l'idea delle caratteristiche delle immagini. Se le immagino sono nitide, poco contrastate oppure hanno problemi con le alte e/o basse luci.

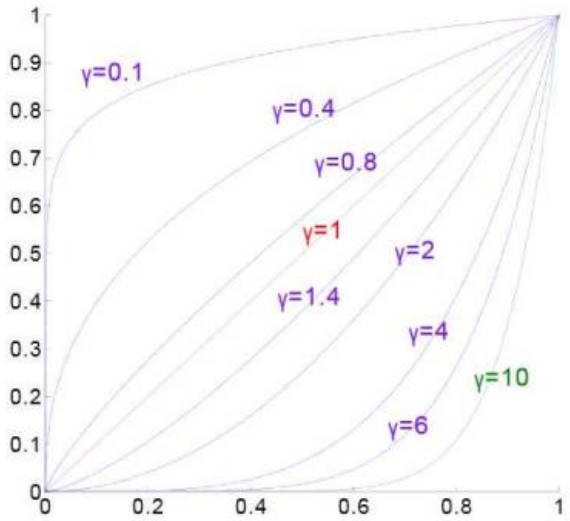
a5. Caricate le immagini '**nrg.jpg**' e '**unequalized.jpg**' e mettetele in due variabili **nrg** e **unequalized.**

Trasformate le immagini usando la funzione **histeq** e memorizzate i risultati nelle variabili **nrg_eq**, e **unequalized_eq**. histeq equalizza l'istogramma dell'immagine andando a distribuire uniformemente i valori dell'istogramma.

b5. Visualizzate gli istogrammi delle immagini equalizzate e confrontateli con gli istogrammi non equalizzati. Cosa si nota?

c5. Provate ad equalizzare l'immagine '**horse.jpg**'.

La tecnica di gamma correction permette di migliorare la leggibilità del contenuto di una immagine. L'immagine di input deve essere a valori tra 0 e 1.



a6. Caricate le immagini '**nrg.jpg**', '**contrast.jpg**' e '**unequalized.jpg**' in tre variabili **nrg**, **contrast** e **unequalized**.

b6.- Applicare la funzione gamma sulle tre immagini come segue: out_image = in_image.^gamma. Provate con una gamma 0.5. Mettete i risultati in tre variabili **nrg_out**, **contrast_out** e **unequalized_out**.

c6. Visualizzate i risultati e confrontate gli istogrammi prima e dopo la gamma. Cosa deducete?

Gli istogrammi dopo la gamma 0.5 sono "spostati" verso la parte alta dei valori. Perchè? Provate a modificare la gamma e vedere il risultato.

Compiti a casa - Lab1

Scrivete gli script o le funzioni Matlab richieste e consegnatele in un file zip tramite l'apposito link sul sito del corso.

Scrivete voi una funzione **myhistogram** che calcola l'istogramma di una immagine a livelli di grigio:

function out_hist = myhistogram(image)

…

end

Scrivete voi una funzione **mycumulativehist** che calcola l'istogramma cumulativo:

function out_cumhist = mycumulativehist(image)

…

end

Scrivete voi una funzione **equalize_image** che calcola l'immagine equalizzata di quella di input. Ricordate che per equalizzare una immagine è necessario il suo istogramma cumulativo. Per mappare un valore di input in quello di ouput equalizzato trovate la formula sulle slide delle lezioni:

$$T(in_k) = out_k = \sum_{j=1}^{k} \frac{n_k}{n} (\times 255)$$

function out_image = **equalize_image**(image)

…

end