

Elaborazione delle Immagini

Laboratorio 9

Obiettivi:

- Applicazione degli operatori morfologici

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, ...)

Consideriamo per il momento solo immagini binarie.

Le operazioni di morfologia matematica in Matlab, si applicano usando le seguenti funzioni:

imerode : erosione delle regioni bianche

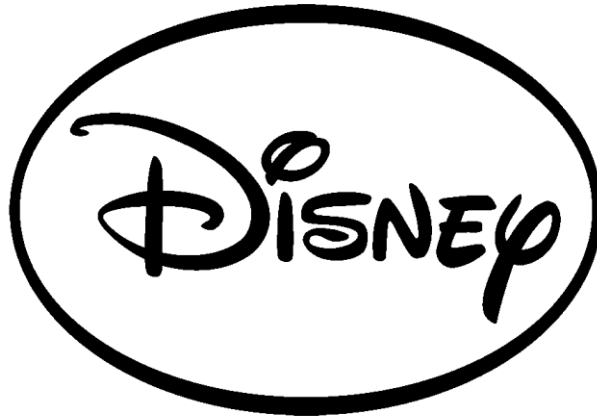
imdilate: dilatazione delle regioni bianche

imopen: operazione di opening

imclose: operazione di closing

Queste funzioni vogliono in input l'immagine binaria e l'elemento strutturante da usare per l'applicazione delle operazioni morfologiche. L'elemento strutturante è un oggetto che va creato a parte usando la funzione **strel**. Questa funzione vuole in input la forma dell'elemento strutturante ed eventualmente dei parametri aggiuntivi. [Guardate la documentazione su queste funzioni.](#)

(1)

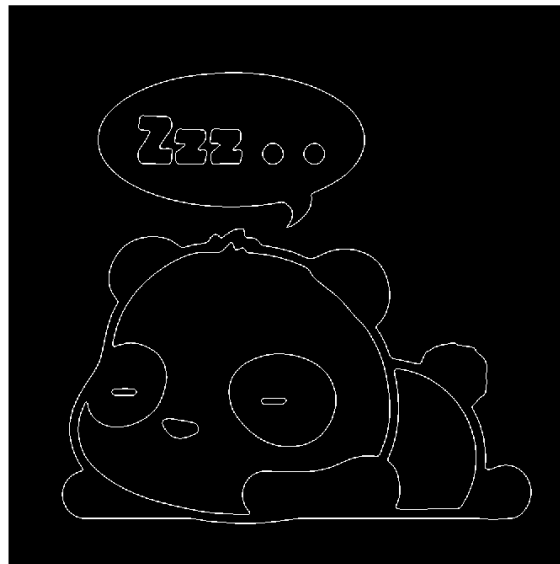


- A1. Caricate l'immagine '**disney.png**' in una variabile **im**.
- B1. Usando la morfologia matematica, vogliamo ridurre lo spessore del tratto di disegno del logo. Usiamo la funzione **imdilate** con un elemento strutturante quadrato di dimensione 3. Create l'immagine risultato **out_dilate** e visualizzatela. Perché usiamo **imdilate**?
- C1. Applicate diverse volte la funzione **imdilate** su **out_dilate** mettendo il risultato sempre in **out_dilate** fino ad ottenere una immagine con l'ovale esterno del logo molto sottile e senza avere buchi. Quante volte avete usato **imdilate**?
- D1. Applicate all'immagine **im**, l'operatore di dilatazione con un elemento strutturante quadrato e dimensione pari a $3 + 2 \cdot (\text{numero di volte che avete applicato la } \mathbf{imdilate})$. Mettete il risultato in una variabile **out**. Confrontate le immagini **out** e **out_dilate**. Che differenze notate?

(2)



- A2. Caricate l'immagine '**panda.png**' in una variabile **im**. Trasformatela in valori logici binarizzandola con la soglia $im > 0$.
- B2. Create l'immagine risultato **dil** usando la funzione **imdilate** con un elemento strutturante a disco di raggio 2. Visualizzatela.
- C2. Combinare le immagini **im** e **dil** applicando degli opportuni operatori logici. Dovete ottenere la seguente immagine di edge (**out**). [In che modo avete combinato le due immagini?](#)



- D2. Se avessimo usato **imerode** invece che **imdilate**, che operazioni sarebbero necessarie per ottenere una immagine di edge?

(3)



- A3. Caricate l'immagine '**coins.png**' in una variabile **im** e trasformatela a valori tra 0 e 1.
- B3. Binarizzate l'immagine **im** in modo tale da ottenere una maschera binaria **bw** che contiene le sole monete. Usate una soglia di 0.8 sul canale della luminanza. Visualizzate l'immagine ottenuta.
- C3. Usate la morfologia matematica per chiudere le imperfezioni che ci sono sulle monete. Non dovete cambiare la dimensione delle monete! Mette il risultato in una variabile **out**.
- D3. Confrontate le immagini **bw** e **out** fino a che ottenete un risultato soddisfacente.

Un utilizzo classico degli operatori morfologici è la chiusura di “buchi” che si possono formare a seguito di una binarizzazione (o classificazione pixel-based).

(4)

L'obiettivo è contare il numero di bottoni che ci sono nell'immagine.



- A4. Caricate l'immagine '**buttons2.png**' in una variabile **im**
- B4. Binarizzate l'immagine **im** in modo tale da ottenere una maschera binaria **bw** che contiene i soli bottoni.
- C4. Usate la morfologia matematica per chiudere i buchi dei bottoni e mettete il risultato in un'immagine **noholes**.
- D4. Dopo il punto precedente, è possibile che alcune regioni dei bottoni molto vicine tra loro si siano fuse in un'unica regione. Usate la morfologia matematica per separare le regioni dei bottoni. Mettete il risultato in una variabile **buttons**. Non è importante la forma precisa delle regioni, ma avere una regione bianca in corrispondenza di ciascun bottone.
- E4. Contare il numero di regioni in **buttons**. Questo numero è il numero di bottoni nell'immagine di partenza.

(5)



- A5. Caricate l'immagine "**portrait.jpg**" e trasformatela a livelli di grigio in una variabile **im**.
- B5. Testate i diversi operatori morfologici sull'immagine **im**. Utilizzate un elemento strutturante a disco. Analizzate e confrontate i vari risultati.
- C5. Caricate l'immagine "**leaf.jpg**" e trasformatela a livelli di grigio in una variabile **leaf**.
- D5. Testate i diversi operatori morfologici sull'immagine **leaf**. Utilizzate un elemento strutturante a disco. [Quale operatore serve per rimuovere il gambo alla foglia? Perché?](#)

(6)

Immaginate e scrivete una procedura automatica di elaborazione delle immagini con l'obiettivo di contare il numero di bottoni che ci sono nell'immagine. Non potete usare la morfologia matematica. Dovete usare un pò di fantasia!

