

Operazioni morfologiche

Operazioni morfologiche

Premessa

Sono una raccolta di operazioni definite principalmente dalle seguenti caratteristiche:

- Permettono di analizzare ed elaborare immagini con uno specifico focus sulle forme e strutture geometriche contenute.
- Si applicano principalmente ad immagini binarie o in scala di grigi.
- Possono essere applicate singolarmente.
- Possono essere combinate fra loro.

Gli ambiti di applicazione sono molteplici.

Operazioni morfologiche

Le principali

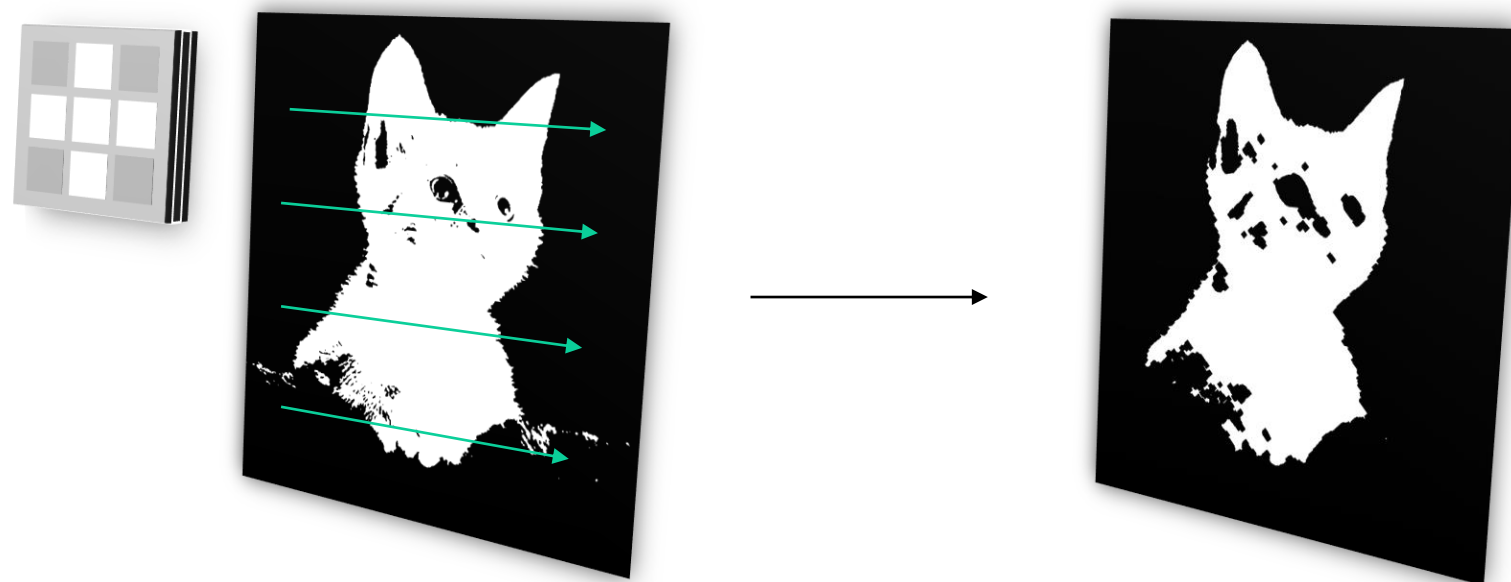
Le operazioni principali sono le seguenti:

Erosion	Rimuove pixel dai contorni delle forme degli oggetti riducendone le dimensioni ed eventualmente eliminando gli oggetti più piccoli.
Dilation	Accresce le dimensioni degli oggetti aumentando il numero di pixel nei contorni delle loro forme. Potenzialmente connette oggetti separati fra loro o chiude buchi contenuti nelle forme stesse.
Opening	Applica in successione erosione e dilatazione al fine principale di eliminare piccoli oggetti contenuti nell'immagine e ridurre il rumore.
Closing	Applica in sequenza una dilatazione ed una erosione al fine principale di riempire buchi nelle forme e ridurre gli spazi che separano gli oggetti fra loro.
Gradient	Rappresenta la differenza fra dilatazione ed erosione al fine di mettere in evidenza i contorni degli oggetti.
Top-hat	Rappresenta la differenza fra l'opening di un'immagine e l'immagine stessa.
Black-hat	Rappresenta la differenza fra l'immagine e il closing della stessa.

Operazioni morfologiche

Funzionamento

Sono versioni specializzate della convoluzione. Applicano un kernel, chiamato *elemento strutturale*, all'immagine in input ottenendo un'immagine in output delle stesse dimensioni.



Il valore associato ad ogni singolo pixel dipenderà dall'operazione scelta e dal comportamento del 'vicinato' di pixel specificati dal kernel.

Operazioni morfologiche

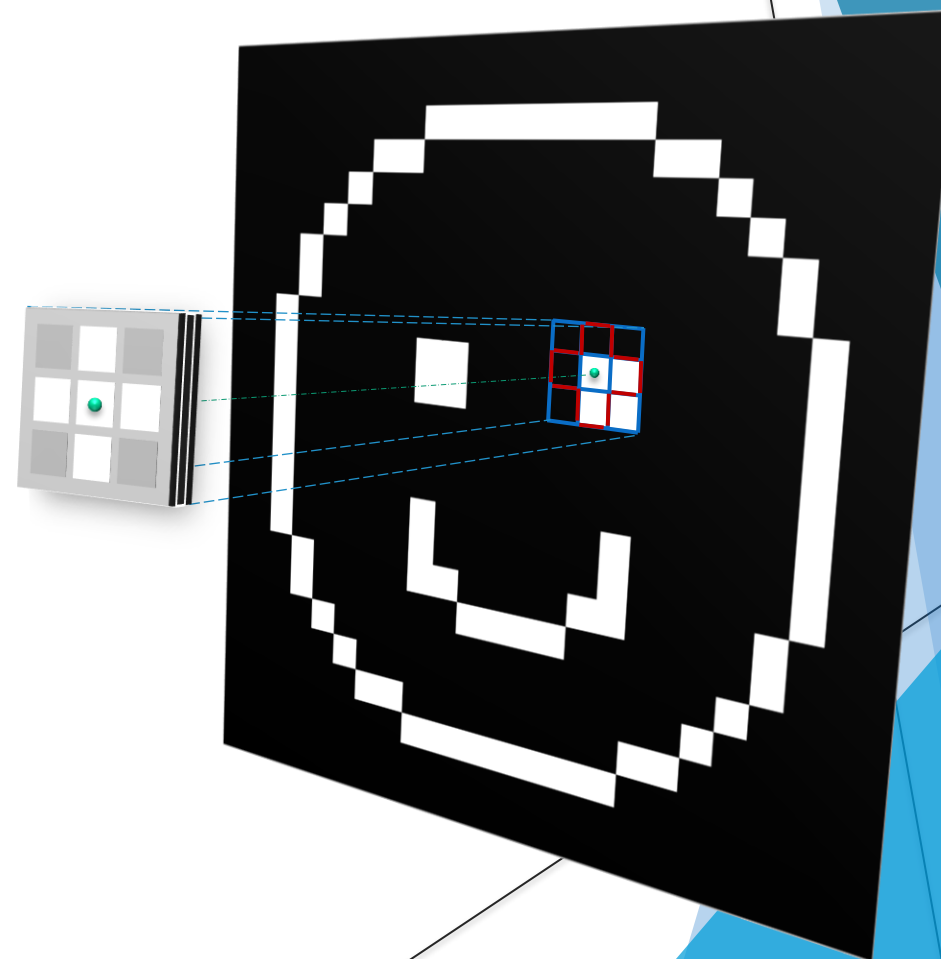
Funzionamento

Per ogni passo di convoluzione, l'elemento strutturale ancora il suo centro con il pixel designato.

La forma rappresentata nell'elemento strutturale viene proiettata sull'immagine sorgente e va a definire i pixel del 'vicinato' da considerare.

In genere si identificano tre situazioni fra la forma nel kernel e la proiezione nel vicinato:

- ❑ **Fit** : il vicinato contiene la forma.
- ❑ **Hit** : il vicinato contiene parte della forma.
- ❑ **Miss** : il vicinato non contiene la forma.



Operazioni morfologiche

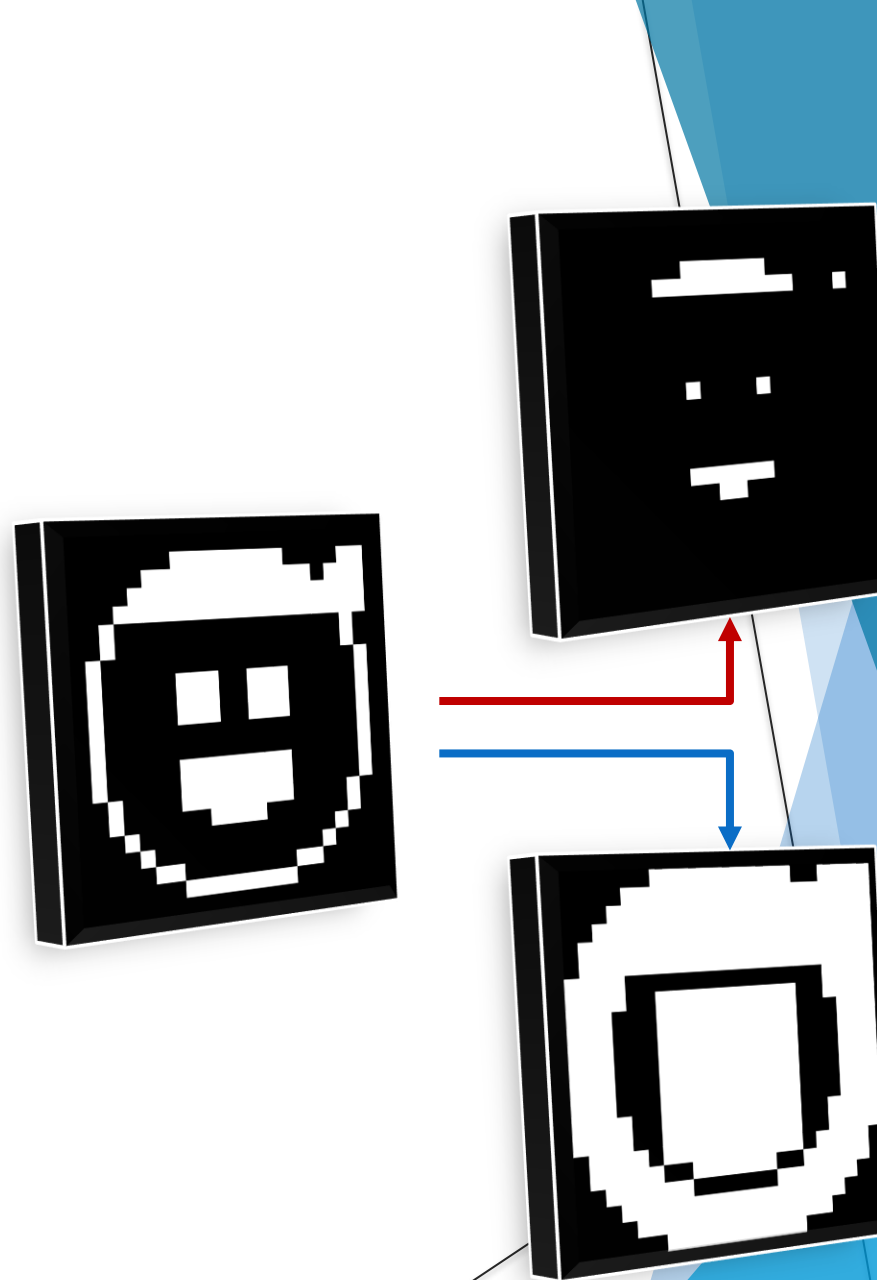
Dilate ed Erode

Le operazioni, come visto, sono diverse ma le principali, base anche di molte delle altre, sono **erosione** e **dilatazione**.

Considerando immagini binarie, 0-1:

- ❖ L'**erosione** si basa sul principio di portare ad 1 (bianco) il pixel ancorato se la forma proiettata è un **fit** nel vicinato.
- ❖ La **dilatazione** si basa sul principio di portare ad 1 il pixel ancorato se la forma proiettata è un **hit** nel vicinato.

Nota: tutto ciò che non è parte della forma viene ignorato.



Operazioni morfologiche

Dilate



Una iterazione



Cinque iterazioni



Operazioni morfologiche

Erode



Una iterazione



Cinque iterazioni



Operazioni morfologiche

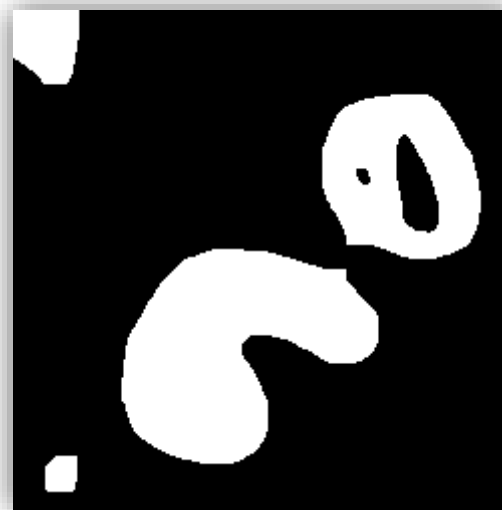
Opening



Una iterazione



Cinque iterazioni



Operazioni morfologiche

Top-Hat



Una iterazione



Cinque iterazioni



Operazioni morfologiche

Closing



Una iterazione



Cinque iterazioni



Operazioni morfologiche

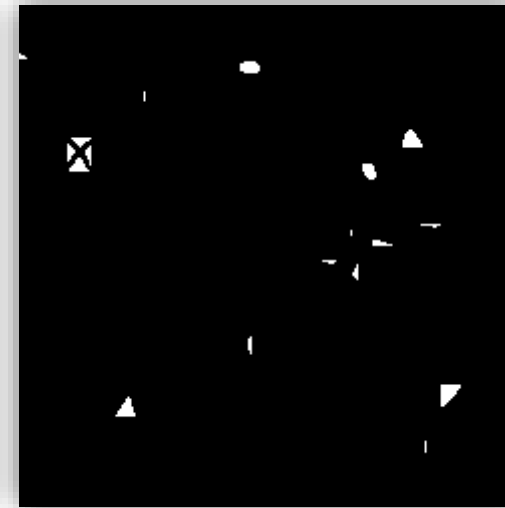
Black-Hat



Una iterazione



Cinque iterazioni



Operazioni morfologiche

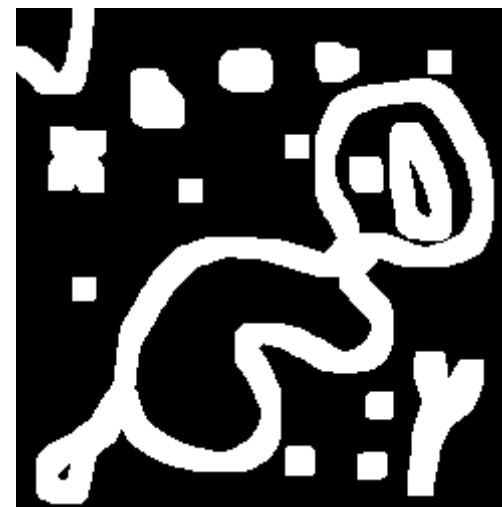
Gradient



Una iterazione



Cinque iterazioni



Operazioni morfologiche

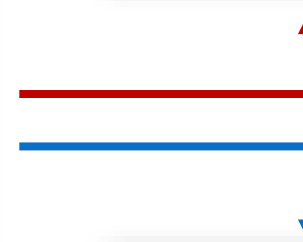
Dilate ed Erode su grayscale

Le operazioni di dilatazione ed erosione, come anche le altre operazioni, sono applicabili sia ad immagini binarie che in scala di grigi.

Per meglio comprenderlo, si possono tradurre in operazioni di ricerca di *massimo* e *minimo*.

- ❖ L'*erosione* setta al pixel ancorato il valore minimo trovato nel vicinato.
- ❖ La *dilatazione* setta al pixel ancorato il valore massimo trovato nel vicinato.

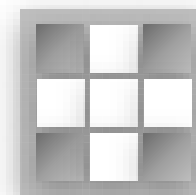
Con questa nuova definizione, non c'è differenza a cercare fra valori fra 0 e 1, 0 e 255 o altro...



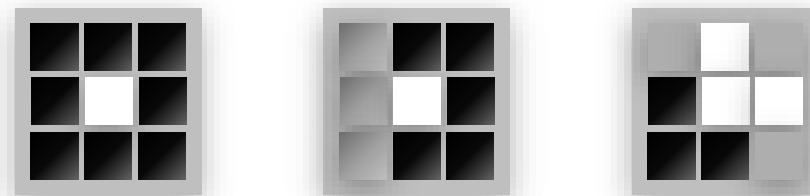
Operazioni morfologiche

Kernel binari

Con un focus sulle immagini binarie, possiamo considerare i pixel neri come ‘sfondo’, il **background**, e i bianchi, il *foreground*.



Con gli esempi precedenti, abbiamo visto un kernel 3x3 di pixel foreground a proiettare una forma a croce; i pixel grigi sono completamente ignorati. Ma è altresì possibile definire un vicinato anche costituito da background.



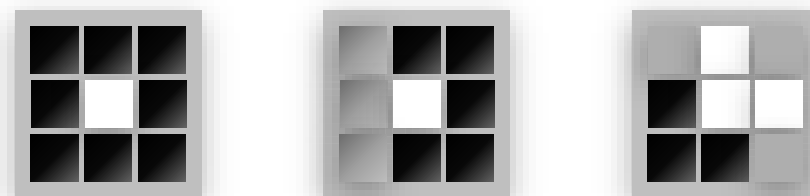
Con dei kernel così definiti è possibile specificare la ricerca di pattern ‘esatti’ di background / foreground e individuare quindi tutti i pixel della sorgente in cui avvengono **hit** o **miss**.

Operazioni morfologiche

Hit-and-Miss

Hit-and-Miss è infatti un'altra delle operazioni fondamentali che fungono da base per altre operazioni morfologiche.

In questo caso, se la forma descritta dal kernel (fatta di foreground e background) ha un match esatto dopo essersi ancorata alla sorgente, allora il pixel ancorato diventa di *foreground*, altrimenti **background**.



Con gli esempi qui indicati possiamo infatti trovare pixel di foreground isolati, pixel terminali che hanno un background sul lato destro, pixel angolari direzionati in basso a sinistra...

Proviamo?