

Introduzione al dato immagine e cenni di computer vision



Prof.ssa Lucia Migliorelli

lmigliorelli@unite.it

Corso di Sistemi multimediali e web per il turismo

Dipartimento di Scienze Politiche, Università di Teramo



**Reale o
generata?**













Digital image processing, digital image analysis, computer vision (1)

Vari concetti:

- Digital Image processing
- Digital Image Analysis
- Computer Vision

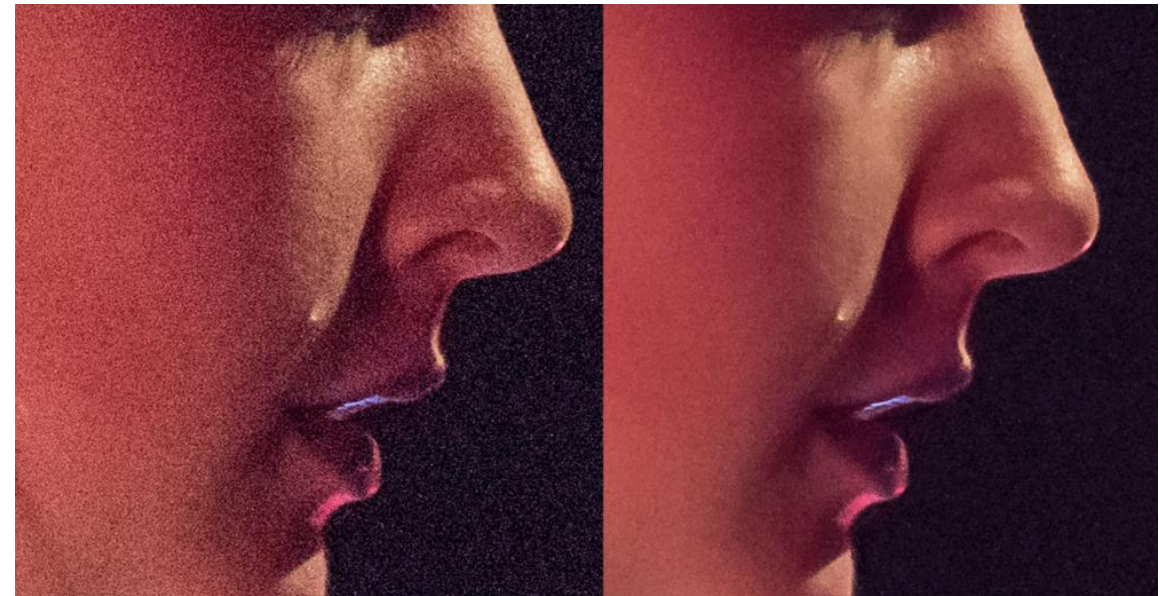
Discipline le cui definizioni e confini non sono univocamente riconosciuti:

- La distinzione tra tipo di output prodotto è esplicativa, ma in pratica le discipline si sovrappongono ampiamente, "sconfinando" l'una nell'altra

Digital image processing, digital image analysis, computer vision (2)

Digital Image Processing (DIP) = elaborazione di immagini digitali

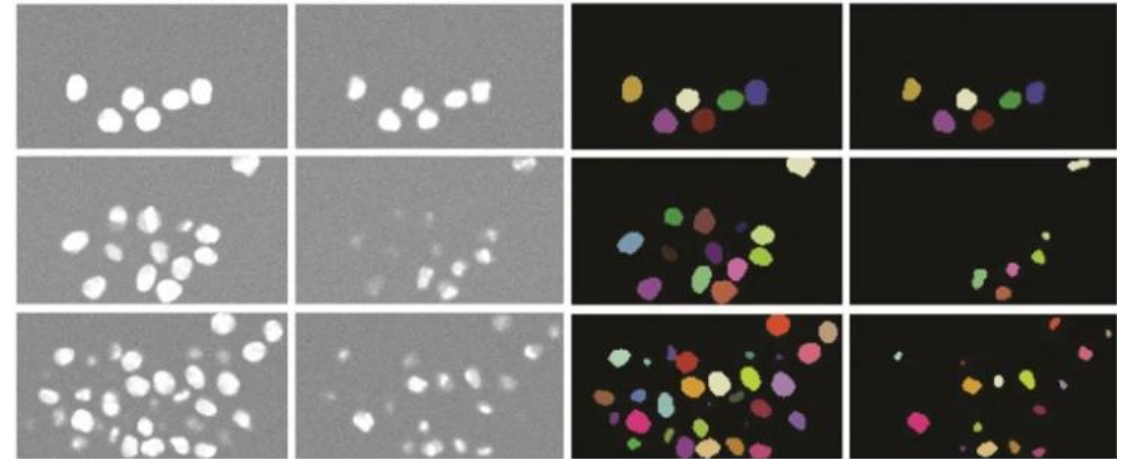
- Obiettivo: migliorare l'immagine o trasformarla (fotoritocco).
- Esempi:
 - Migliorare contrasto e luminosità.
 - Ridurre il rumore.
 - Applicare filtri .



Digital image processing, digital image analysis, computer vision (3)

Digital Image Analysis (DIA) = analisi di immagini digitali

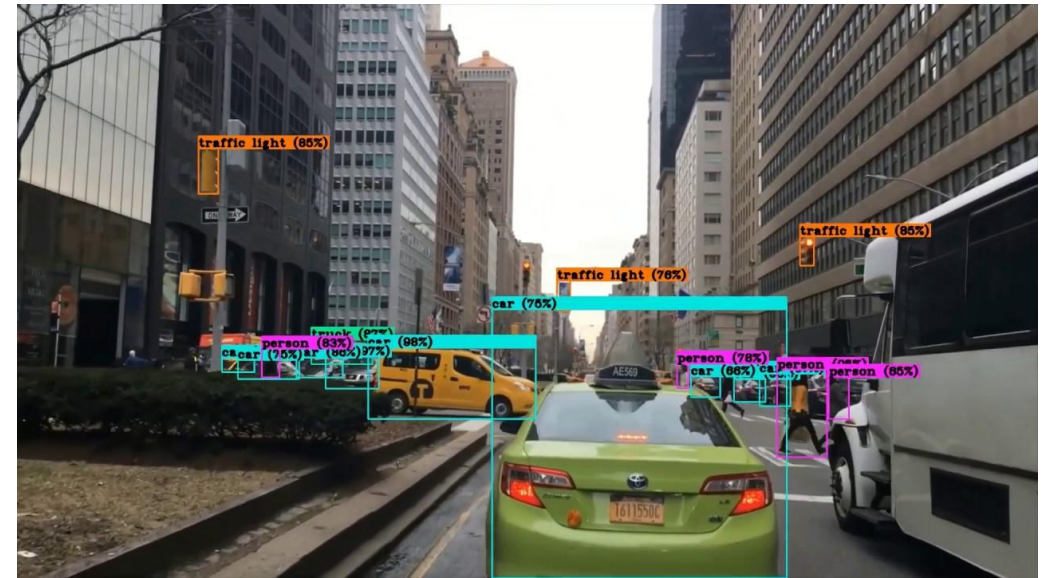
- Obiettivo: estrarre informazioni utili dall'immagine.
- Esempi:
 - Contare quanti oggetti ci sono in una foto.
 - Segmentare una regione (es. separare cielo e terra in una foto turistica).
 - Trovare forme o misurare grandezze (es. area di un monumento in una foto satellitare).
- Qui non interessa più "manipolare/abbellire" la foto, ma capire cosa contiene.



Digital image processing, digital image analysis, computer vision (4)

Computer Vision (CV) = visione artificiale

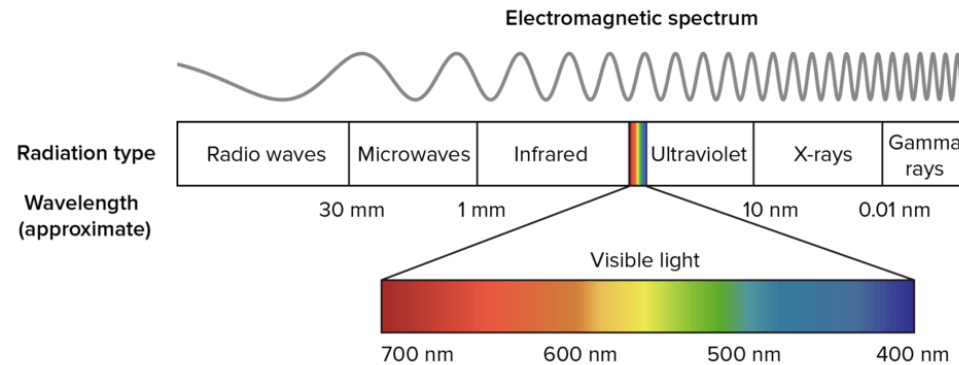
- Obiettivo: dare a una macchina la capacità di “vedere” e interpretare il mondo come noi.
 - Estensione della DIA
- Esempi:
 - Riconoscere i volti.
 - Capire i gesti di una persona.
 - Guidare un'auto senza conducente.
- Qui entriamo nel campo dell'intelligenza artificiale applicata alle immagini.



Dal fascio di luce al pixel: come nasce un'immagine digitale (1)

Cos'è la luce?

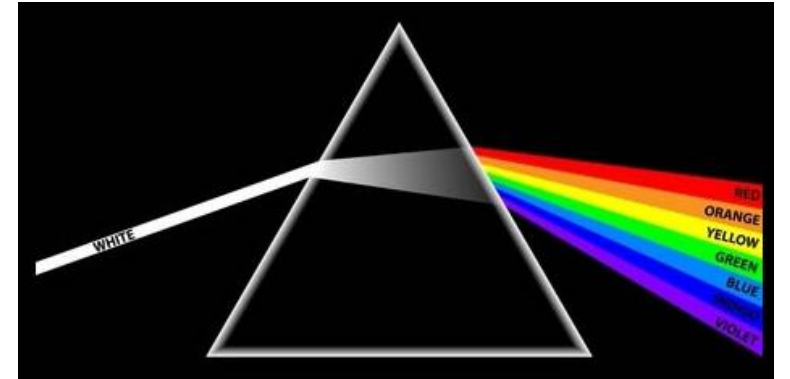
- Radiazione elettromagnetica
- Lo spettro del visibile va da 400 nm (violetto) a 700 nm (rosso)
 - Ogni lunghezza d'onda in questo intervallo corrisponde ad un colore percepito



Dal fascio di luce al pixel: come nasce un'immagine digitale (2)

Cos'è la luce?

- La luce bianca è composta da tutte le lunghezze d'onda visibili.
- Con un prisma si può scomporre nei colori fondamentali (arcobaleno).



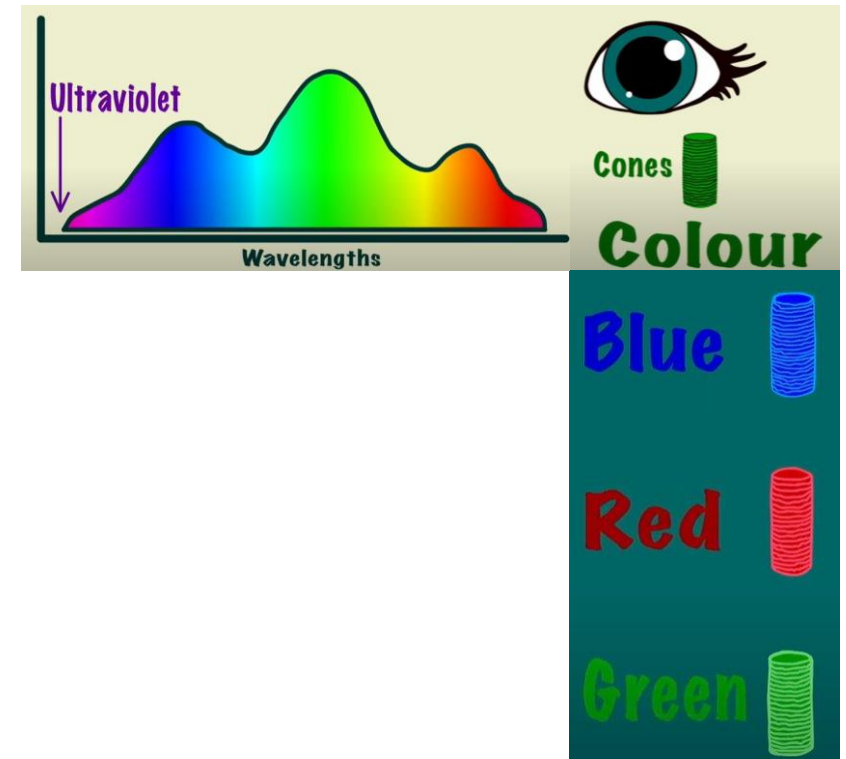
Dal fascio di luce al pixel: come nasce un'immagine digitale (3)

Percezione della luce nell'occhio umano

La retina ha due tipi di “sensori biologici”:

- **Bastoncelli** → molto sensibili alla luce, servono per la visione notturna, ma non distinguono i colori.
- **Coni** → meno sensibili alla luce, ma responsabili della visione a colori.

I bastoncelli ci permettono di vedere al buio (in bianco e nero), i coni ci permettono di distinguere i colori alla luce.



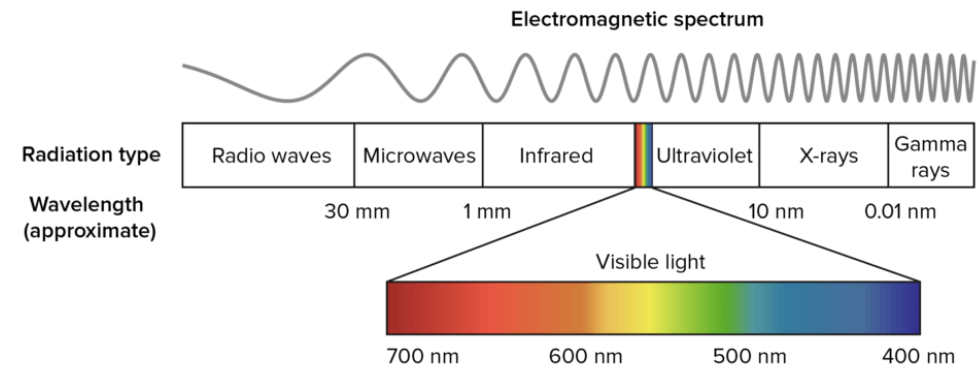
Dal fascio di luce al pixel: come nasce un'immagine digitale (4)

Percezione della luce nell'occhio umano (2)

I coni non sono sensibili a tutti i colori, ma solo a tre bande principali:

- **Coni L (long, “rossi”)** → massima sensibilità intorno a ~560–570 nm (giallo/rosso).
- **Coni M (medium, “verdi”)** → massima sensibilità intorno a ~530 nm.
- **Coni S (short, “blu”)** → massima sensibilità intorno a ~420 nm.

Nessun cono “vede” il colore puro, ognuno risponde più o meno a una gamma di lunghezze d'onda.



Dal fascio di luce al pixel: come nasce un'immagine digitale (5)

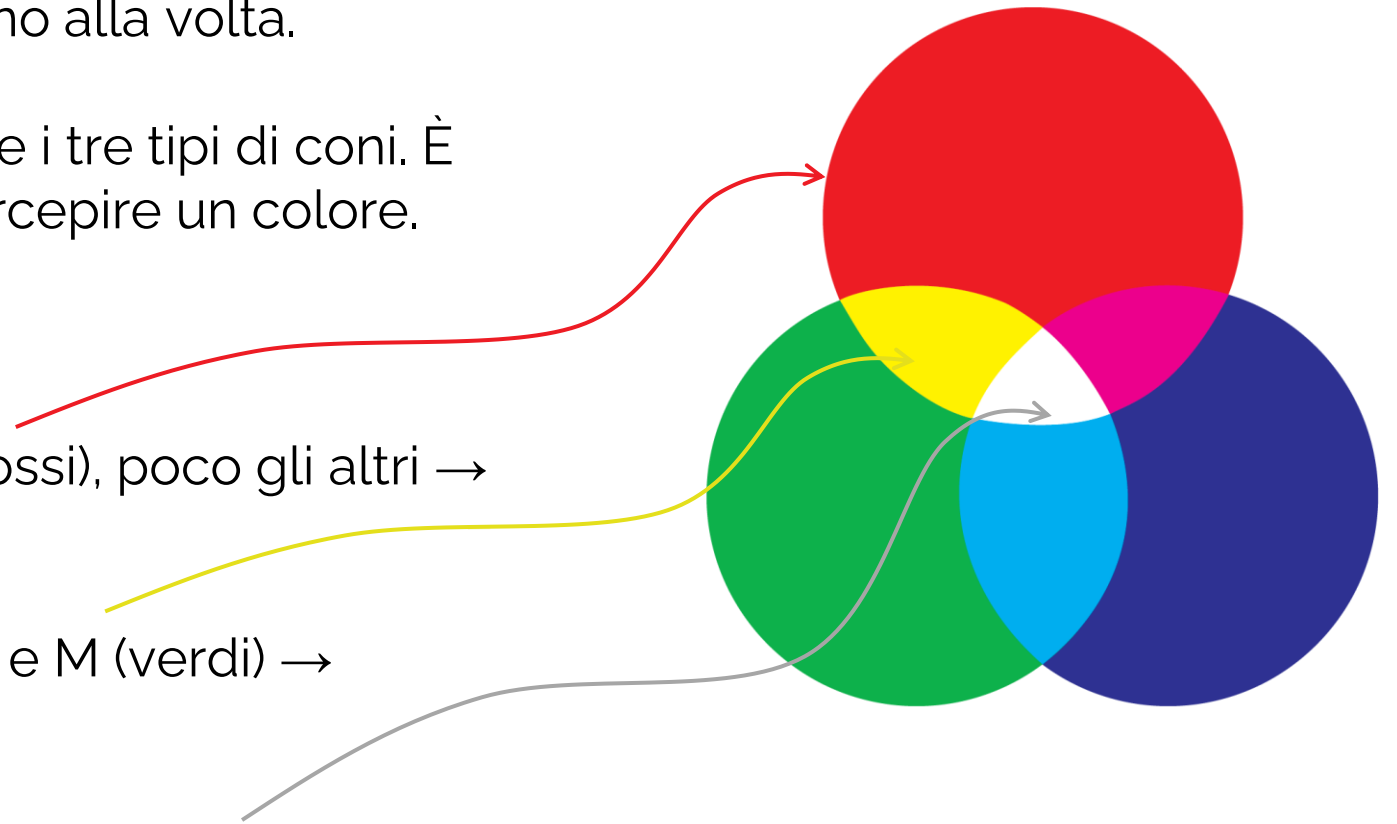
Come nasce il colore nella percezione

Il nostro cervello non legge un cono alla volta.

Guarda quanto rispondono insieme i tre tipi di coni. È questa combinazione che ci fa percepire un colore.

Esempio:

- Se rispondono molto i coni L (rossi), poco gli altri → percepisco rosso.
- Se rispondono insieme L (rossi) e M (verdi) → percepisco giallo.
- Se rispondono tutti e tre → percepisco bianco o grigio.

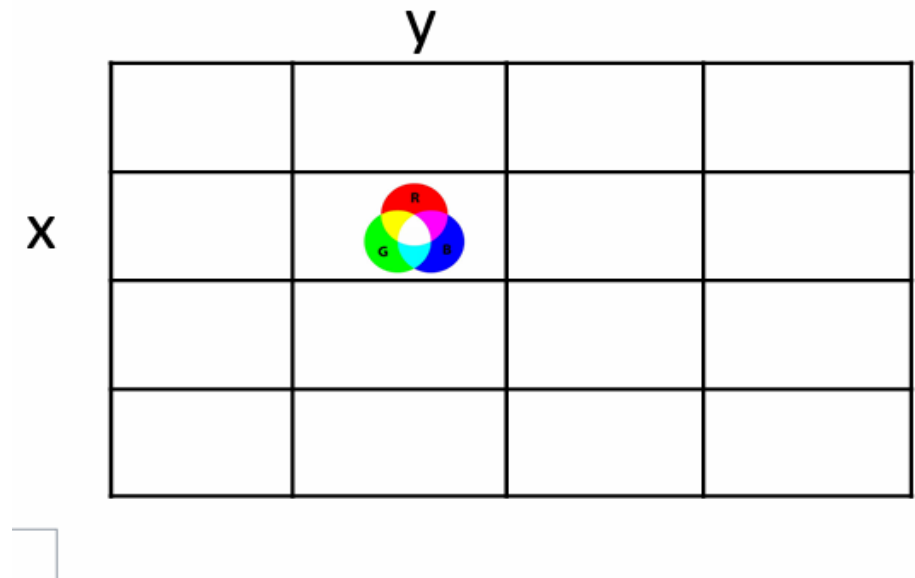


Immagini digitali (caso RGB)

L'occhio «riduce» lo spettro continuo della luce a tre valori (RGB)

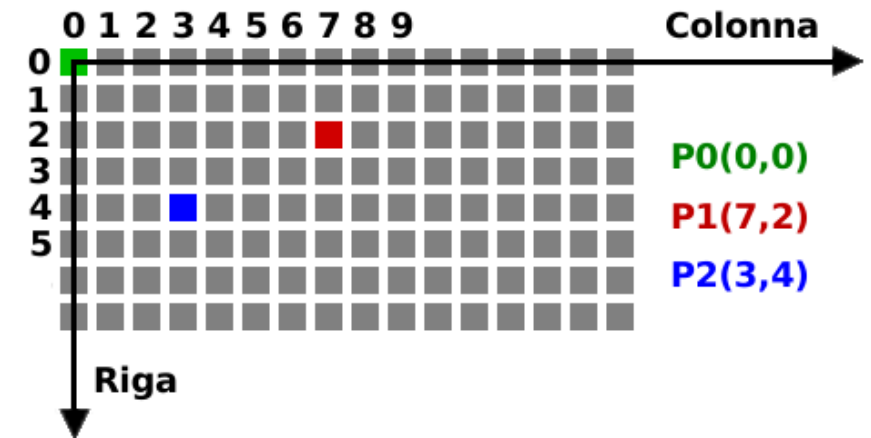
Come imita il computer l'occhio umano?

- Anche il computer riduce lo spettro a 3 canali: R, G, B.
- Ogni pixel = 3 numeri (uno per ogni canale).
- Combinando R, G, B → si ricostruiscono tutti i colori visibili.



Immagini digitali

- Nelle immagini digitali (raster) i pixel sono disposti ordinatamente all'interno di una matrice rettangolare. Per individuare ciascuno di questi puntini si utilizza un sistema di coordinate cartesiane con l'origine in alto a sinistra e l'asse positivo delle y rivolto verso il basso.
- Da un punto di vista fisico ciascun pixel, secondo le tecnologie correnti, può emettere normalmente: luce monocromatica (M), luce tricromatica (RGB=Red,Green,Blue)



Immagini digitali (raster)– pixel logica vs fisica

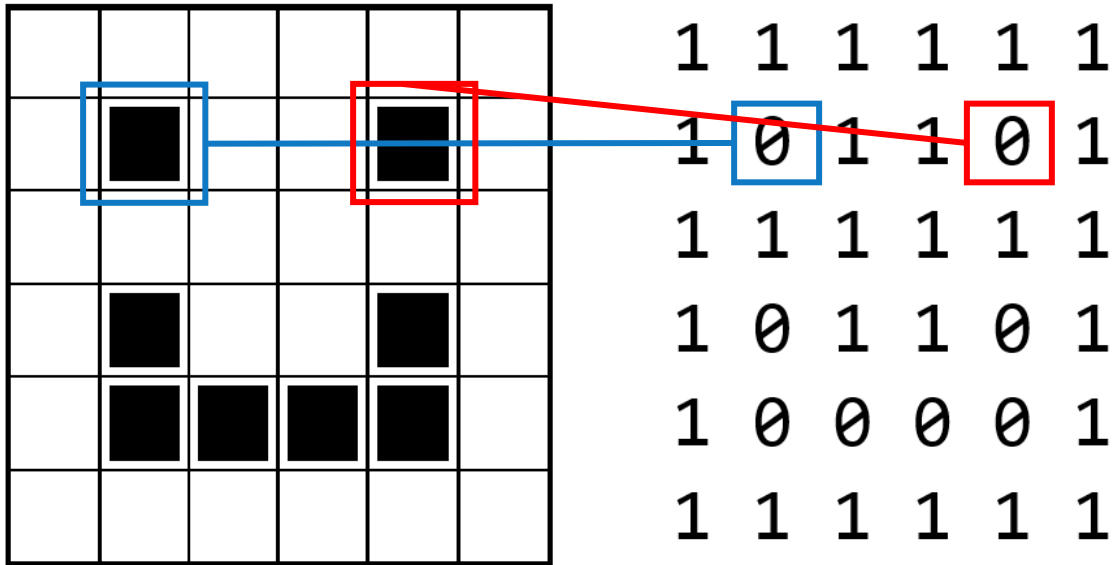
Pixel come unità logica

- Nel file digitale il pixel è solo un numero → descrive il colore in quel punto.
- È un'unità astratta che non ha dimensioni fisiche.
- Esempio: un'immagine di dimensione 1920×1080 ha sempre 2.073.600 pixel (2 MP), sia guardandola su uno smartphone, sia su una TV enorme.
 - I numeri che descrivono l'immagine sono legati al concetto di definizione dell'immagine

Unità fisica

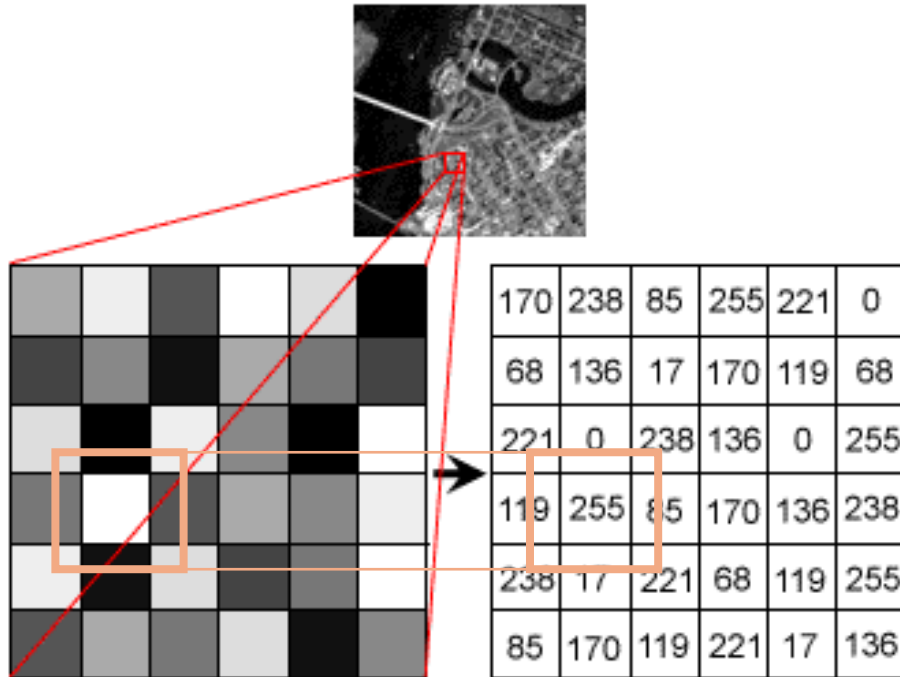
- Quando quell'immagine viene mostrata su uno schermo ogni pixel logico deve “prendere spazio”.
 - Su uno schermo piccolo → i pixel logici diventano punti fisici più piccoli.
 - Su uno schermo grande → gli stessi pixel diventano punti fisici più grandi

Immagini digitali codifica binaria



- Il pixel assume valore 1 o 0. Per rappresentarlo in memoria si usa un bit che può essere acceso (bianco=1) oppure spento (nero=0).
- Efficienti in termini di spazio di memoria richiesto
- Adatte per testi, codici a barre, impronte digitali, schemi/piante, ...
- Sebbene teoricamente ogni pixel di un'immagine binaria possa essere rappresentato da un singolo bit, in pratica, a seconda del formato del file o del sistema di elaborazione dell'immagine, potrebbero essere usati più bit.

Immagini digitali codifica in scala di grigi



- In un'immagine in scala di grigi, ogni pixel rappresenta un livello di grigio. La scala di grigio varia dal bianco al nero, passando attraverso i toni di grigio. Più alto è il valore associato a un pixel, più chiaro è il grigio; più basso il valore, più scuro il grigio.
- Se ogni pixel è rappresentato da un byte, allora ogni pixel può assumere uno tra 2^8 (= 256 sfumature) possibili valori
 - Valori da 0 (nero) a 255 (bianco): Quando si utilizza un byte per pixel, si hanno 256 possibili livelli di intensità.

Immagini digitali codifica RGB



49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

Red

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

Green

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

Blue

Nelle immagini RGB ogni pixel ha un'intensità di R (0-255) G (0-255) B (0-255). Quindi abbiamo **3 byte per pixel**.

L'immagine RGB è uno stack di 3 matrici

I Canali del colore

Pixel at (0, 0) - Red: 147, Green: 150, Blue: 146
Pixel at (50, 20) - Red: 165, Green: 166, Blue: 161
Pixel at (160, 160) - Red: 244, Green: 114, Blue: 32
Pixel at (180, 280) - Red: 0, Green: 0, Blue: 255



Pixel at (160, 160) - Red: 244, Green: 114, Blue: 32

Valori di rosso, verde e blu nelle varie posizioni della matrice immagine.

Un'immagine digitale è composta da tanti pixel, cioè piccoli quadratini colorati.

Ogni pixel ha tre valori numerici:

- R (Red) = quantità di rosso,
- G (Green) = quantità di verde,
- B (Blue) = quantità di blu.

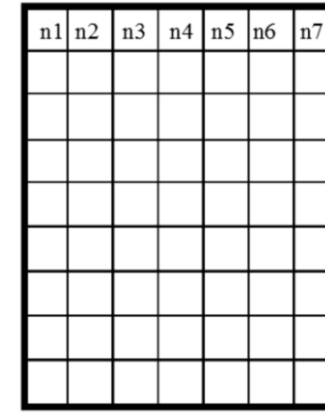
Combinando questi tre numeri otteniamo il colore del pixel.

Ad esempio:

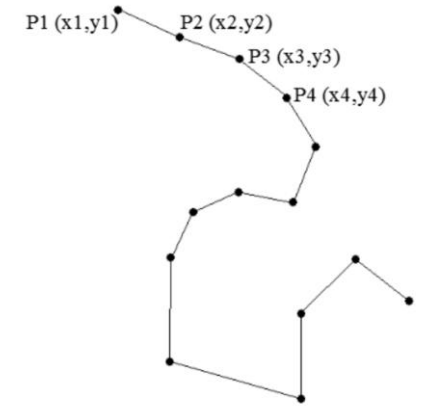
- R=244, G=114, B=32 → un arancione acceso,
- R=0, G=0, B=255 → blu puro.

Immagini digitali – metodi di rappresentazione

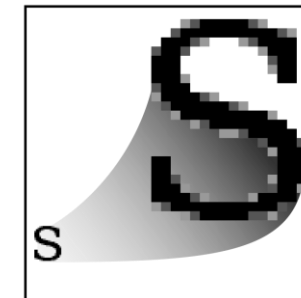
- **Raster:** Informazione grafica rappresentata come matrice (griglia) di pixel (ciò di cui abbiamo parlato fino adesso)
- **Vettoriale:** Informazione grafica rappresentata da relazioni matematiche (funzioni che descrivono luoghi di punti)
 - Immagini definite a partire da forme geometriche definite sul piano Cartesiano come linee, curve, poligoni. Ad esempio si usano per i font.



RASTER



VETTORIALE



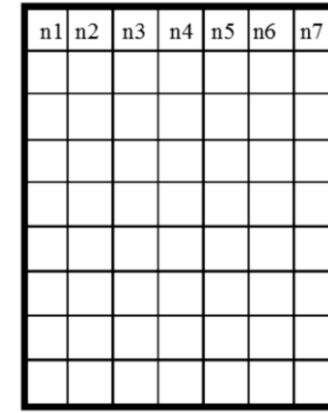
Raster
GIF, JPEG, PNG



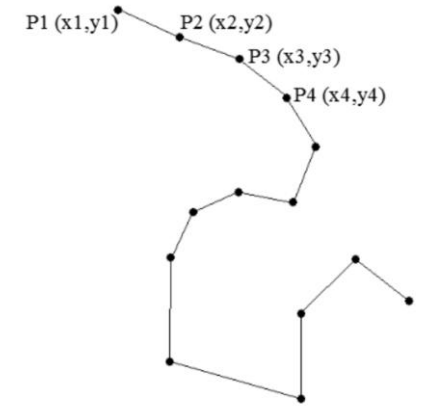
Vector
SVG

Immagini digitali – metodi di rappresentazione

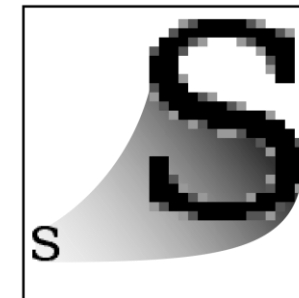
- **Un'immagine raster** è una griglia di pixel, ognuno con il suo colore.
- Come funziona? Ogni pixel è definito da valori numerici (ad esempio i tre canali R, G, B).
- Caratteristiche:
 - Utilizzate per fotografie
 - Se ingrandite troppo, diventano sgranate (si vedono i pixel).
- Esempi: foto fatte con lo smartphone, immagini JPEG, PNG, BMP.



RASTER



VETTORIALE



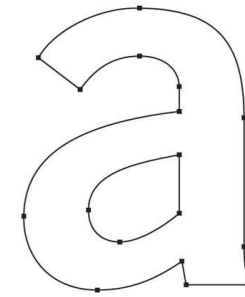
Raster
GIF, JPEG, PNG



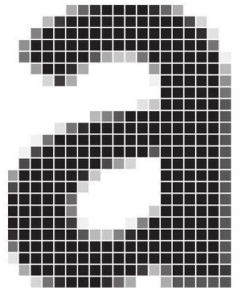
Vector
SVG

Immagini digitali – metodi di rappresentazione

- **Un'immagine vettoriale** non è fatta da pixel, ma da formule matematiche che descrivono linee, curve, cerchi e poligoni.
- Come funziona: il computer disegna l'immagine ogni volta a partire da queste istruzioni matematiche.
- Caratteristiche:
 - Non perdono qualità se ingrandite (sono scalabili all'infinito).
 - Perfette per loghi, icone, grafici, testi.
 - Meno adatte a fotografie.
- Esempi: loghi in formato SVG, file CAD, grafica in Adobe Illustrator o Inkscape



VECTOR



RASTER

Immagini digitali – metodi di rappresentazione

- **Un'immagine vettoriale** non è fatta da pixel, ma da formule matematiche che descrivono linee, curve, cerchi e poligoni.

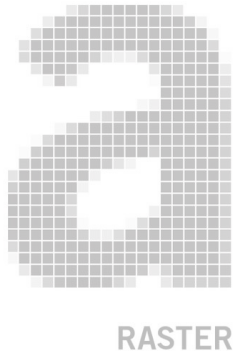
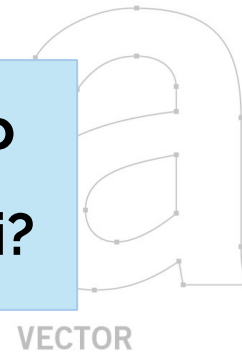
- Come funziona: il computer disegna l'immagine ogni volta a partire da queste informazioni.

- Caratteristiche:

- Non perdono qualità se ingrandite (sono scalabili all'infinito).
- Perfette per loghi, icone, grafici, testi.
- Meno adatte a fotografie.

- Esempi: loghi in formato SVG, file CAD, grafica in Adobe Illustrator o Inkscape

Perché le foto da smartphone non sono rappresentate come immagini vettoriali?



Immagini digitali – metodi di rappresentazione

- Un'immagine vettoriale non è fatta da pixel, ma da formule

matematiche che

- Come funziona:

partire da queste

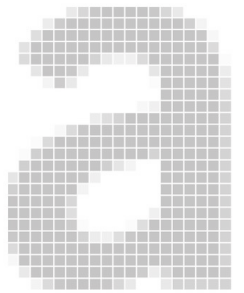
- Caratteristiche:

- Non perdono qualità
- Perfette per loghi
- Meno adatte a foto

- Esempi: loghi in formato SVG, file CAD, grafica in Adobe Illustrator o Inkscape

Perché le foto da smartphone non sono rappresentate come immagini vettoriali?

- Una fotografia contiene milioni di dettagli diversi: sfumature di luce, ombre, texture,... Per descrivere tutto questo con formule matematiche (linee, curve, aree colorate) servirebbero tantissimi elementi geometrici.
- Il formato raster è perfetto, perché «basta» registrare il colore di ogni pixel



RASTER

Profondità del colore immagini raster vs immagini vettoriali

- **Nelle immagini raster)**

- Ogni pixel memorizza direttamente il suo colore.
- La profondità (es. 24 bit = 8 bit per R, G, B) dice quanti colori diversi può avere ciascun pixel. Più bit → più sfumature → foto più realistiche.
- Un cerchio rosso viene memorizzato come tanti pixel vicini con valori (255,0,0).

- **Nelle immagini vettoriali)**

- Ci sono forme riempite con un colore.
- Quel colore è codificato in un modello numerico (ad esempio RGB a 24 bit, 8 per canale)
 - Cerchio = riempimento (255,0,0) → Rosso
 - bordo) = (0,0,0) → nero

Memorizzazione di immagini (formati raster)

JPEG (Joint Photographic Experts Group)

- 8 bit per colore (RGB), no trasparenza
- Compressione lossy (con perdita)
- Ideale per fotografie

PNG (Portable Network Graphics)

- 8 bit per colore (RGB) + 8 bit per trasparenza (canale alpha)
- Compressione lossless (senza perdita di qualità)
- Ideale per immagini con sfondo trasparente, loghi e grafica

BMP (Bitmap)

- 8 bit per colore (RGB) + 8 bit per trasparenza
- Nessuna compressione
- La sua diffusione è dovuta al suo uso in Microsoft Windows, non si usa quasi più

Memorizzazione di immagini (formati vettoriali)

SVG (Scalable Vector Graphics)

- Formato aperto basato
- E' un file XML che descrive l'immagine da visualizzare («disegna un cerchio rosso»)

AI (Adobe Illustrator artwork)

- Proprietario
- Formato vettoriale nativo per Adobe Illustrator

PDF (Portable Document Format)

- Proprietario
- Basato sul linguaggio di descrizione di pagina di Adobe, definisce anche immagini vettoriali, oltre alle caratteristiche del testo

Altri formati

- EPS (Encapsulated PostScript, usato in tipografia e grafica),
- DXF, AutoCAD DWG (formati specifici per il disegno tecnico (CAD))

Memorizzazione di immagini (formati raster - vettoriali)

SVG (Scalable Vector Graphics)

- Formato aperto basato
- E' un file XML che descrive l'immagine da visualizzare («disegna un cerchio rosso»)

AI (Adobe Illustrator a

- Proprietario
- Formato vettoriale

PDF (Portable Document Format)

- Proprietario
- Basato sul linguaggio di descrizione di pagina di Adobe, definisce anche immagini vettoriali, oltre alle caratteristiche del testo

Altri formati

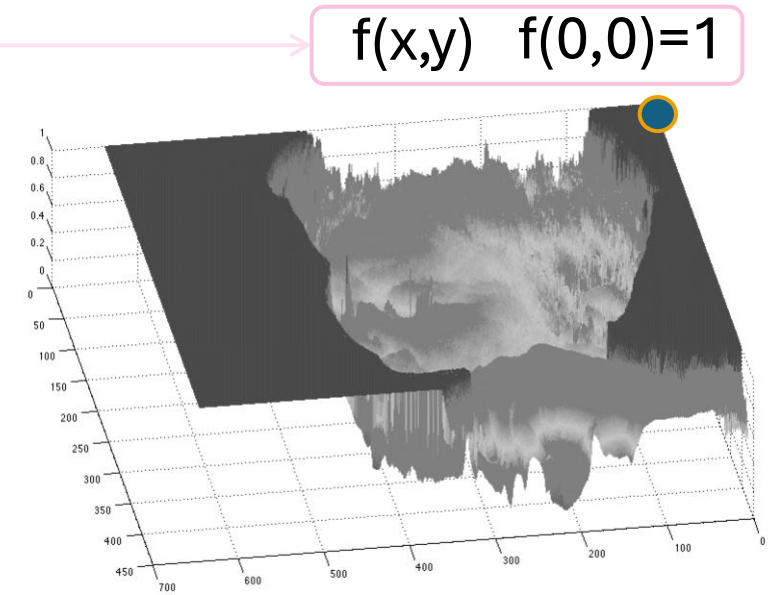
- EPS (Encapsulated PostScript, usato in tipografia e grafica),
- DXF, AutoCAD DWG (formati specifici per il disegno tecnico (CAD))

- I **formati raster** (JPEG, PNG, BMP) memorizzano pixel.
- I **formati vettoriali** (SVG, AI, PDF...) memorizzano **istruzioni matematiche** per disegnare linee, curve e riempimenti di colore.

Come interpretiamo le immagini, il concetto di funzione

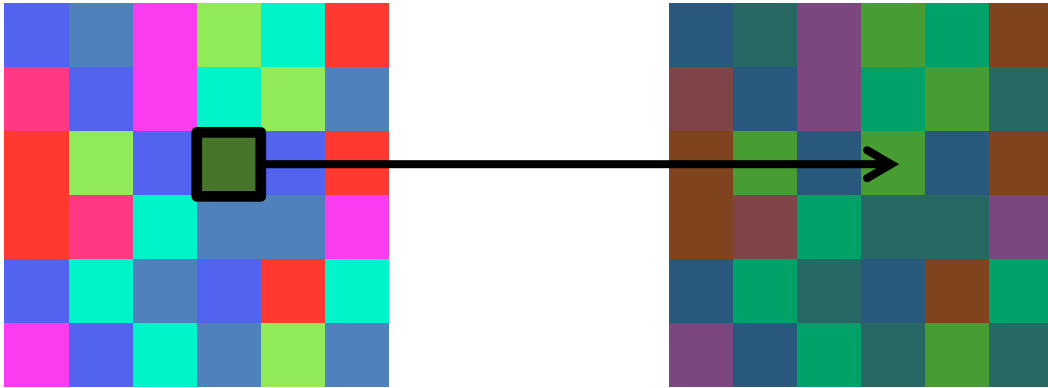
- Un'immagine digitale si può pensare come una funzione che prende in ingresso due numeri: la posizione (x,y) di un punto.
- In uscita la funzione restituisce un valore di luminosità (quanto chiaro o scuro è quel punto, nel caso di immagini in scala di grigi ad esempio).
- Le coordinate (x,y) vanno da 0 al numero massimo di pixel dell'immagine (larghezza e altezza).
- Il valore di luminosità va da 0 = nero a L_{max} = bianco (con valori intermedi = grigi).

- NOTA: Nell'immagine la scala è invertita solo per motivi di visualizzazione (ovvero l'intensità massima=1 è il nero)
- Data la rappresentazione in figura il pixel in posizione $x,y = 0,0$ ha valore di luminosità ($f(0,0)$) pari ad 1

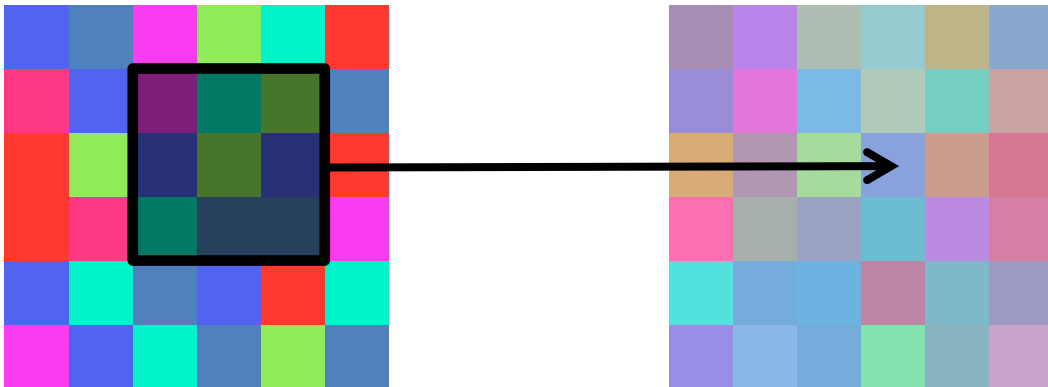


Filtraggio di immagini

Point Operation



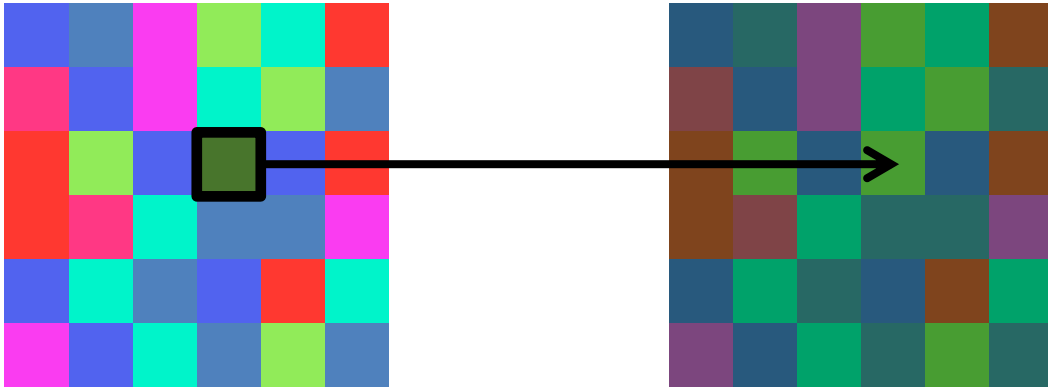
Neighborhood Operation o filtraggio



Qual è la differenza tra i due approcci?

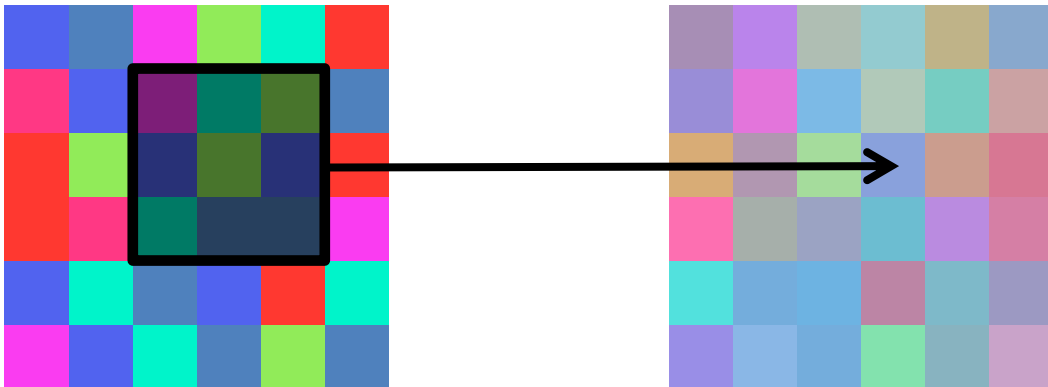
Filtraggio di immagini

Point Operation



- L'operazione è applicata individualmente a ogni pixel dell'immagine.
- Il valore del pixel dopo l'operazione è determinato solo dal valore originale di quel pixel, indipendentemente dal valore dei pixel circostanti.

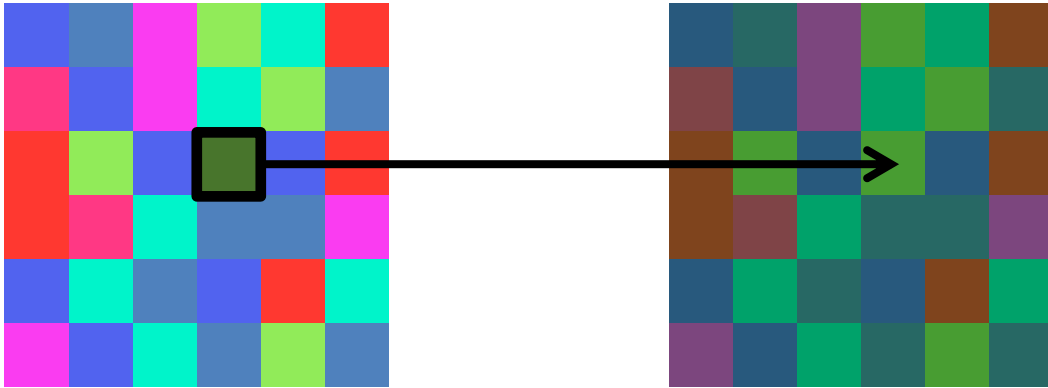
Neighborhood Operation o filtraggio



- Il nuovo valore del pixel dipende non solo dal valore del pixel originale ma anche dal valore dei pixel vicini

Filtraggio di immagini

Point Operation



- L'operazione è applicata individualmente a ogni pixel dell'immagine.
- Il valore del pixel dopo l'operazione è determinato solo dal valore originale di quel pixel, indipendentemente dal valore dei pixel circostanti.

Neighborhood Operation o filtraggio



Ai fini del corso si considerano solamente le operazioni del tipo point operation

Il valore del pixel dipende non solo dal pixel originale ma anche dal valore dei

Esempio di point processing operation (1)

original



darken



lower contrast



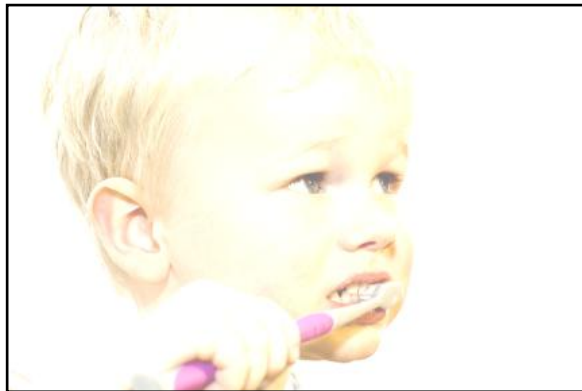
non-linear lower contrast



invert



lighten



raise contrast



non-linear raise contrast



Come le implementiamo? (2)

original



darken



lower contrast



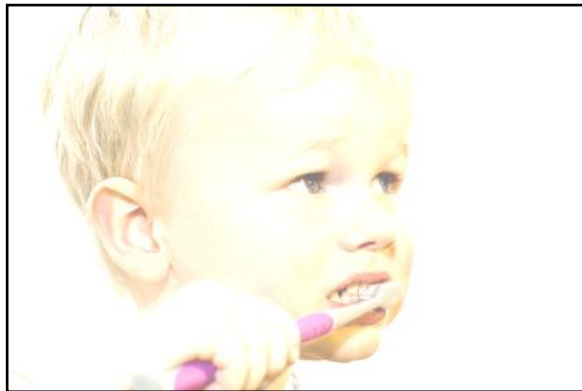
non-linear lower contrast



invert



lighten



raise contrast

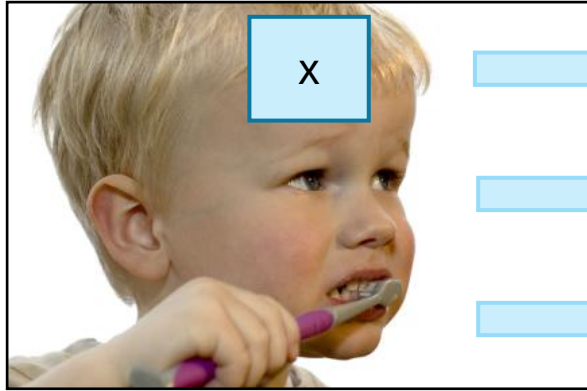


non-linear raise contrast



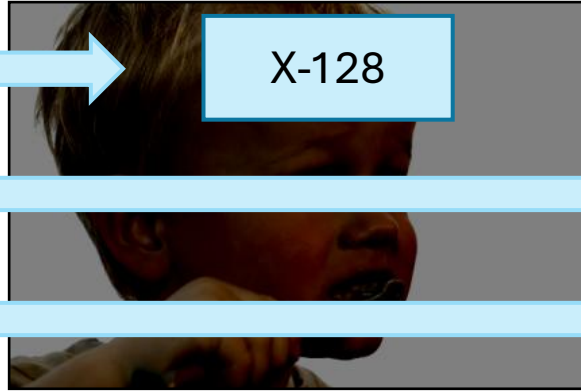
Come le implementiamo? (3)

original



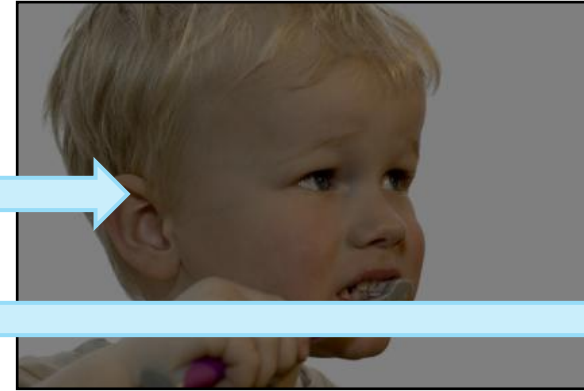
x

darken



X-128

lower contrast



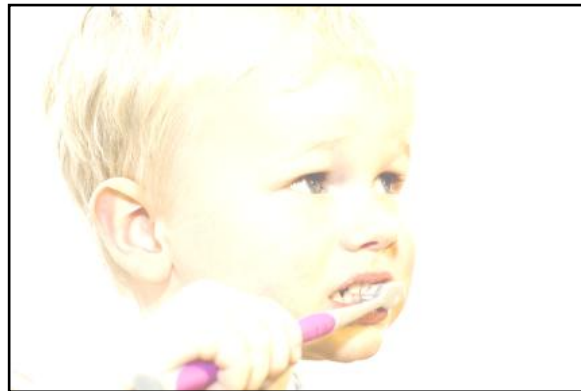
non-linear lower contrast



invert



lighten



raise contrast

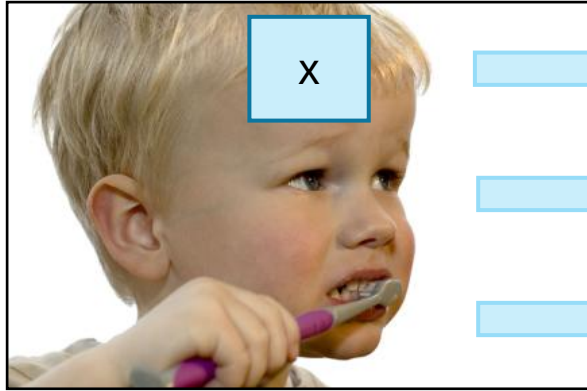


non-linear raise contrast

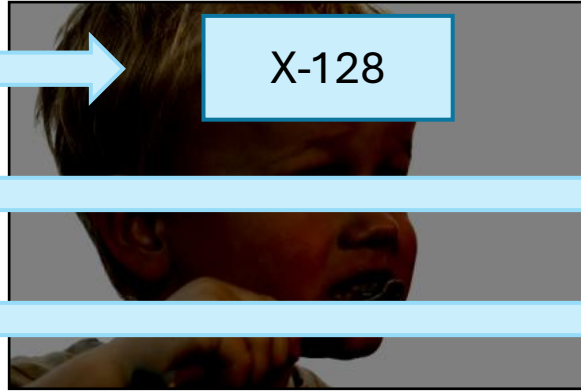


Come le implementiamo? (4)

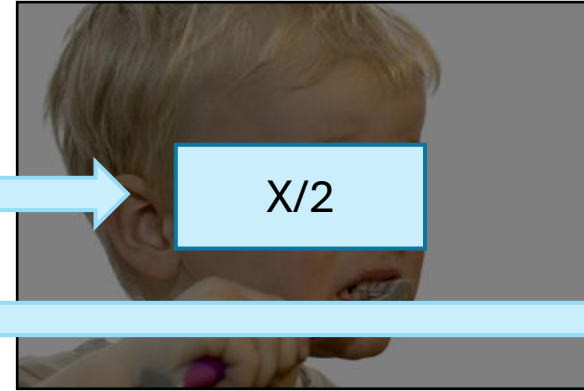
original



darken



lower contrast



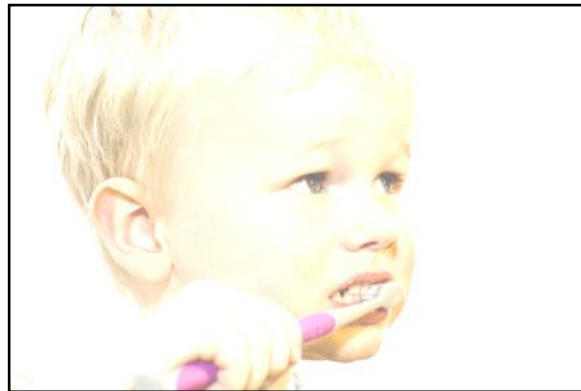
non-linear lower contrast



invert



lighten



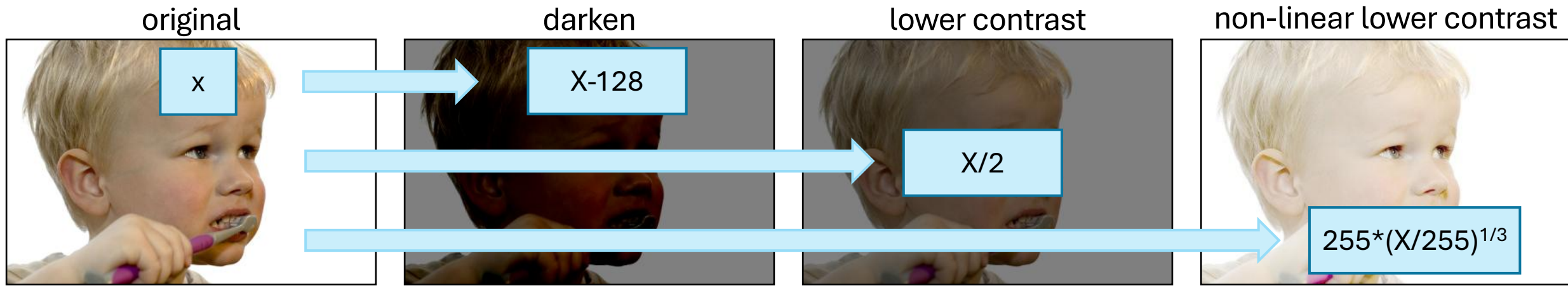
raise contrast



non-linear raise contrast

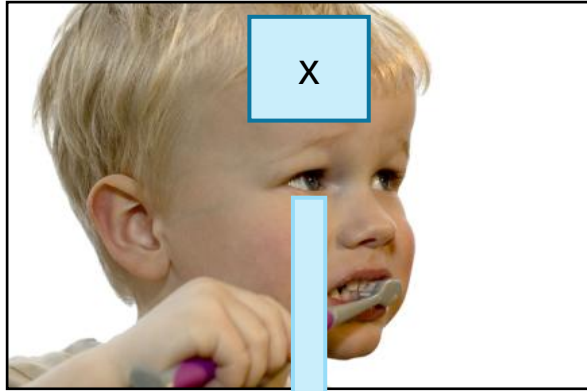


Come le implementiamo? (5)



Come le implementiamo? (6)

original



x

invert

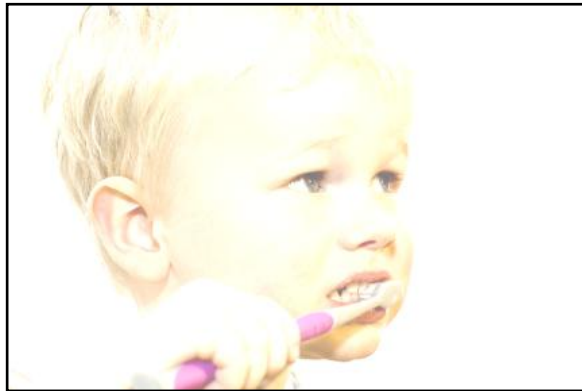


$255-x$

darken



lighten



lower contrast



raise contrast



non-linear lower contrast

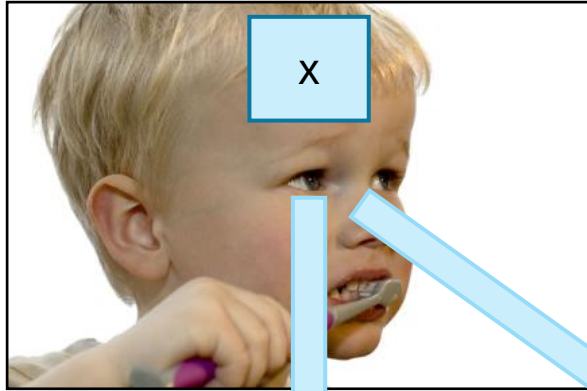


non-linear raise contrast



Come le implementiamo? (7)

original



x

darken



lower contrast



non-linear lower contrast

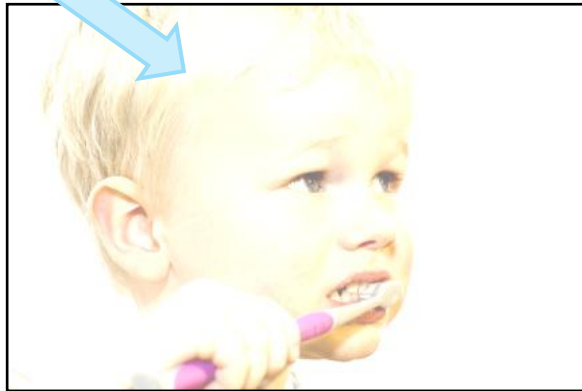


invert



$255-x$

lighten



$x+128$

raise contrast

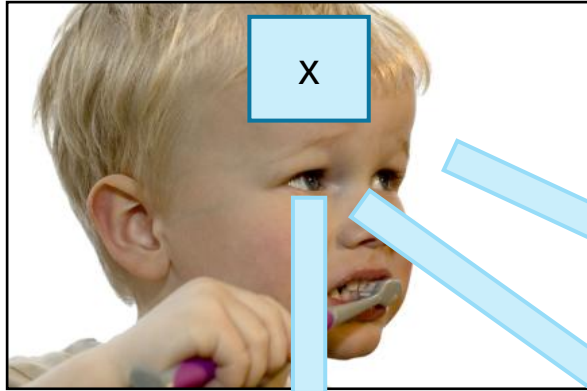


non-linear raise contrast



Come le implementiamo? (8)

original



x

darken



lower contrast



non-linear lower contrast

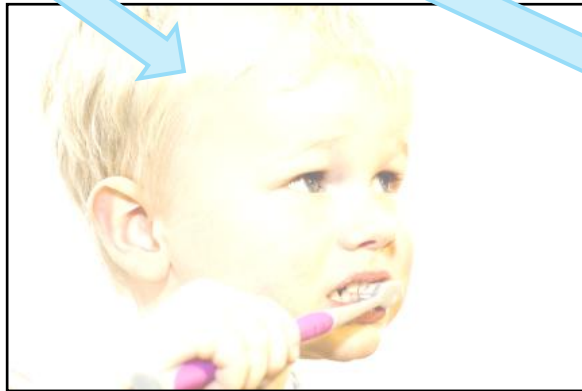


invert



$255-x$

lighten



$x+128$

raise contrast



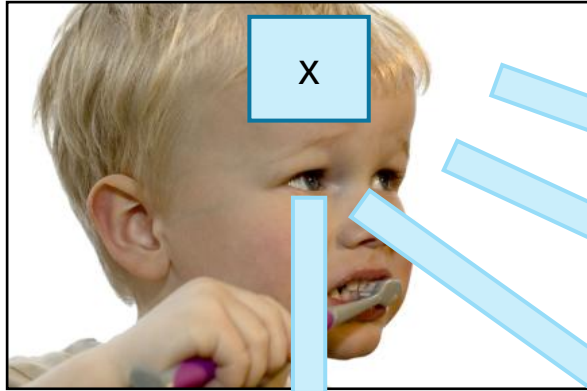
$x*2$

non-linear raise contrast



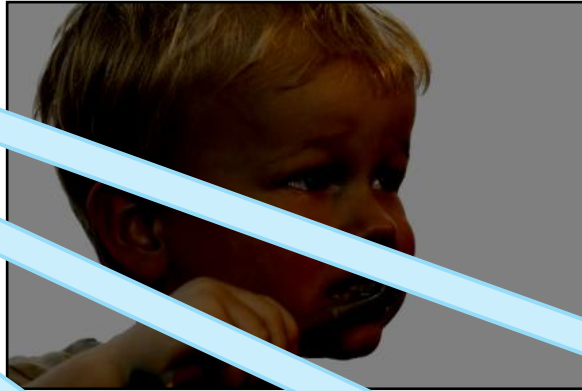
Come le implementiamo? (9)

original



x

darken



lower contrast



non-linear lower contrast

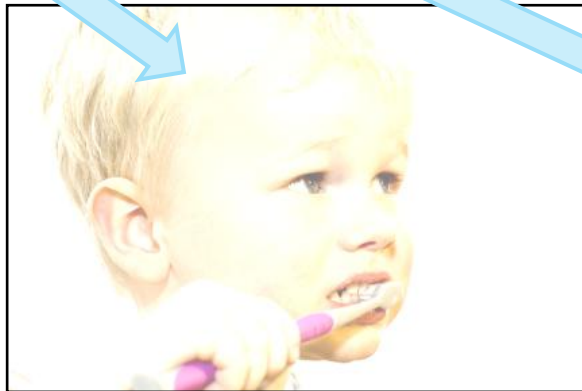


invert



$255-x$

lighten



$X+128$

raise contrast



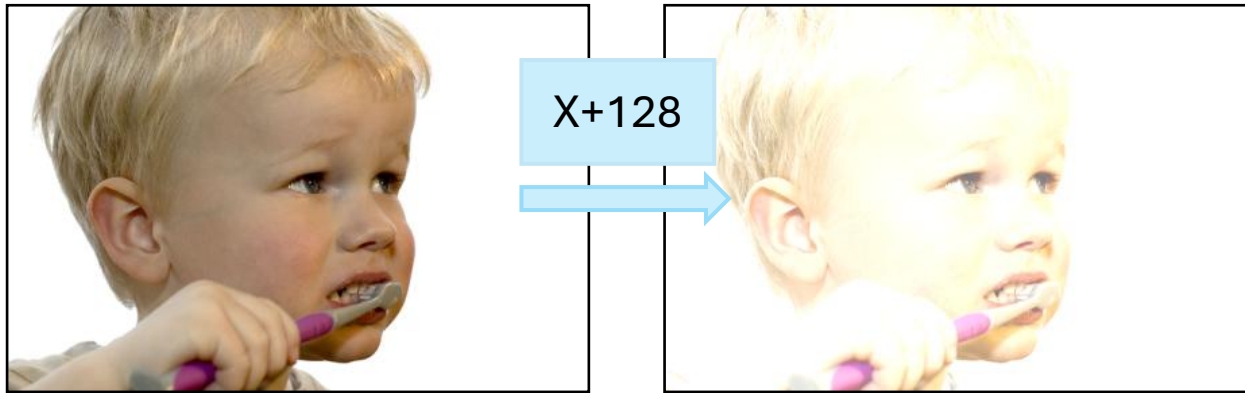
$X*2$

non-linear raise contrast



$255*(X/255)^2$

Errori di codifica: valori fuori dal range

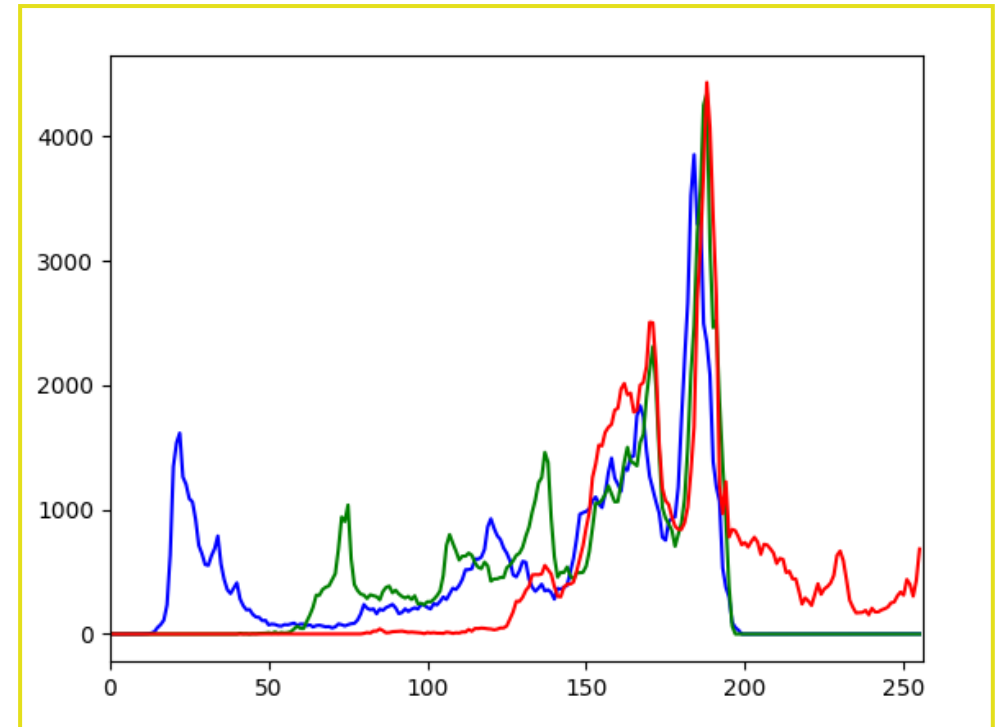


- Consideriamo il caso della trasformazione lineare che schiarisce i pixel di un'immagine.
- I pixel devono rimanere nel range $[0, K-1]$ (di solito $[0, 255]$).
- Tuttavia sommando il 128 alcuni valori possono diventare troppo alti (oltre il 255) → più chiari del bianco.
- Ma questi valori **non esistono** in un'immagine digitale, quindi vengono "tagliati".
- Sommando 128 ad un pixel di valore 130 ottengo 258 → questo valore non può essere rappresentato quindi viene approssimato a 255.

Fenomeno di saturazione

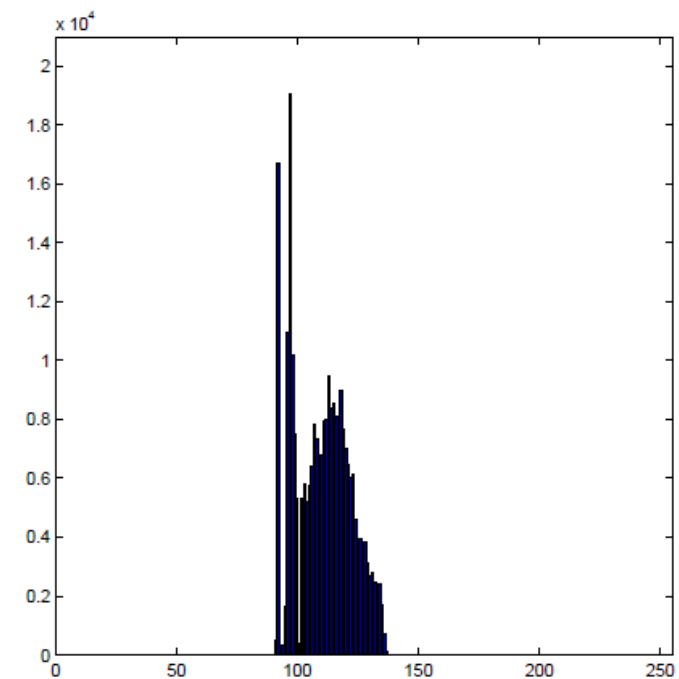
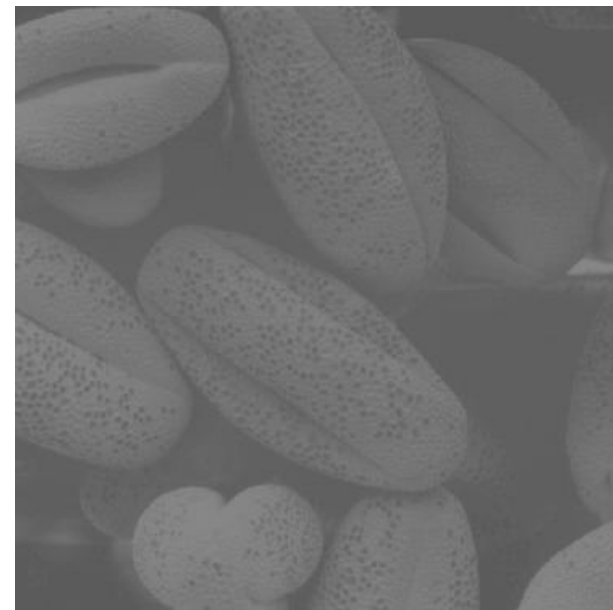
Istogramma di colore (1)

- Un **istogramma di colore** è un grafico che conta **quanti pixel** di un'immagine hanno un certo valore di colore o di luminosità.
- **In un'immagine in scala di grigi:** ogni pixel può andare da **0 = nero** a **255 = bianco**. L'istogramma mette sull'asse orizzontale tutti questi valori (0-255) e sull'asse verticale quante volte ciascun valore compare nell'immagine.
 - Se un'immagine ha tanti pixel scuri, l'istogramma avrà delle colonne alte vicino allo 0. Se invece è molto chiara, le colonne saranno alte vicino al 255.
- **In RGB** si può calcolare un istogramma per **ogni canale** (rosso, verde, blu), ottenendo tre grafici che descrivono come sono distribuiti i colori nell'immagine.



Istogramma di colore (2)

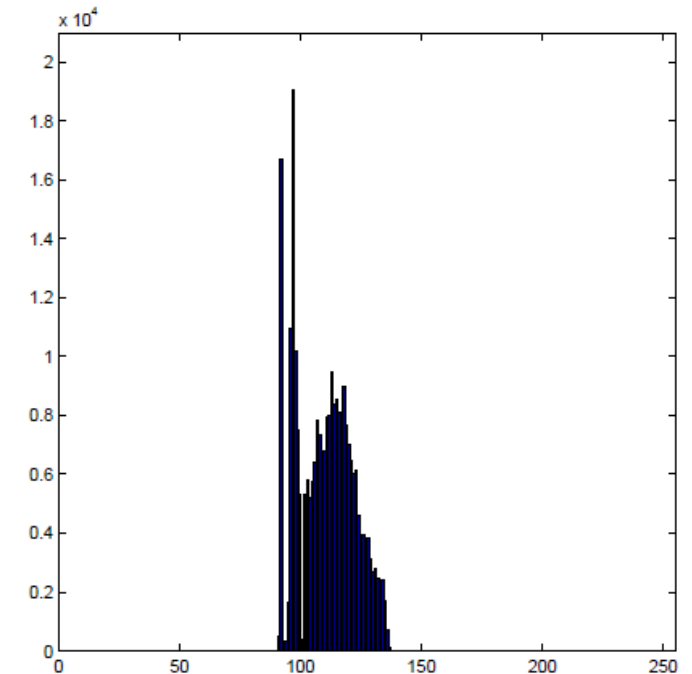
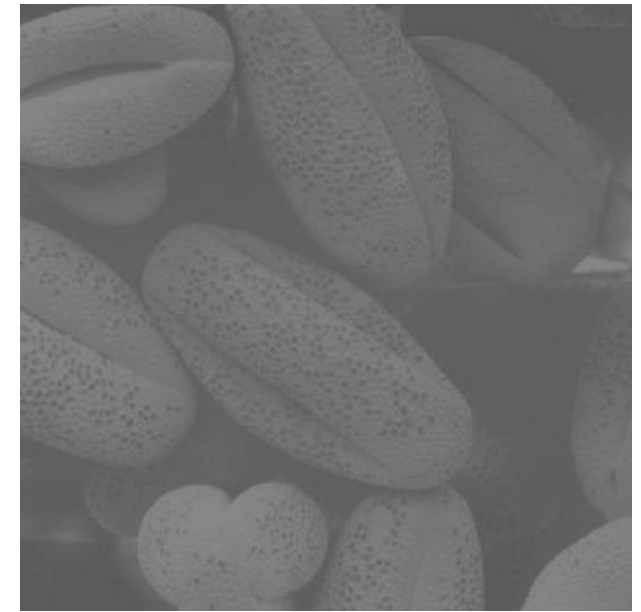
Cosa ci dice l'istogramma?



Istogramma di colore (3)

Cosa ci dice l'istogramma?

- **Distribuzione dei toni:** ci fa capire se l'immagine è scura, chiara o ben bilanciata.
- **Contrasto:** se i valori si concentrano solo in una zona ristretta (es. tra 90 e 140 su 0-255), significa che l'immagine ha poco contrasto → sembra "piatta" o "grigia".
- **Informazioni perse:** dall'istogramma non possiamo ricostruire l'immagine originale, perché immagini diverse possono avere la stessa distribuzione dei pixel.



Resizing dell'immagine (1)

2	3
4	5

Ho un'immagine di dimensioni 2x2 (2 righe e 2 colonne) la debbo portare a dimensione 4x4 (4 righe e 4 colonne) o 5X5, ... , che cosa faccio?

Resizing dell'immagine (2)

2	3
4	5

Ho un'immagine di dimensioni 2x2 (2 righe e 2 colonne) la debbo portare a dimensione 4x4 (4 righe e 4 colonne) o 5x5, ... , che cosa faccio?

5x5

2	x	x	x	3
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
4	x	x	x	5

4x4

2	x	x	3
x	x	x	x
x	x	x	x
4	x	x	5

Che valore assumono i nuovi pixel (x) dell'immagine?

Resizing dell'immagine (3)

2	3
4	5

Ho un'immagine di dimensioni 2x2 (2 righe e 2 colonne) la debbo portare a dimensione 4x4 (4 righe e 4 colonne) o 5x5, ... , che cosa faccio?

5x5				
2	x	x	x	3
x	x	x	x	x
x	x	x	x	x
x	x	x	x	x
4	x	x	x	5

4x4			
2	x	x	3
x	x	x	x
x	x	x	x
4	x	x	5

Che valore assumono i nuovi pixel (x) dell'immagine?

Ho varie possibilità: replicazione, interpolazione, ...

Resizing dell'immagine (4)

2	3
4	5

Ho un'immagine di dimensioni 2x2 (2 righe e 2 colonne) la debbo portare a dimensione 4x4 (4 righe e 4 colonne) o 5x5, ... , che cosa faccio?

5x5				
2	2	3	3	3
2	2	3	3	3
2	2	3	3	3
4	4	4	5	5
4	4	4	5	5

Inputo i valori dei pixel mancanti con la tecnica di replicazione

4x4			
2	2	3	3
2	2	3	3
4	4	5	5
4	4	5	5

Che valore assumono i nuovi pixel (x) dell'immagine?

Ho varie possibilità: replicazione, interpolazione, ...