

Analisi Comparativa degli Spazi Colore RGB, LAB e L*

Studio Quantitativo delle Immagini, Clustering
tramite K-means, DBI e Percezione Visiva Umana

Bellesini Silvia 41946A
Conca Francesco 29404A
De Tommaso Simona 31916A



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Indice

Argomenti trattati

01 – Percezione del colore

02 – Confronto tra gli spazi colore

03 – Algoritmo K-Means e DBI

04 – Spiegazione del codice

05 – Conclusioni

Obiettivo del progetto

Capire in quale spazio colore l'algoritmo K-Means riesce a segmentare un'immagine in modo più simile alla percezione visiva umana.

Per rispondere a questa domanda, abbiamo utilizzato due strategie:

- 1 – Visiva: abbiamo osservato e confrontato le immagini segmentate nei diversi spazi colore, per vedere dove i risultati ci sembravano più naturali, più realistici e più omogenei
- 2 – Quantitativa: abbiamo usato una misura matematica chiamata Davies–Bouldin Index. Questo indice ci dice quanto i cluster sono compatti e ben separati: valori più bassi significano che i cluster sono migliori, quindi più omogenei al loro interno e più distanti dagli altri.



Perchè studiare la percezione del colore?

Capire la percezione del colore significa capire come rendere più umane le tecnologie che usiamo ogni giorno

Il progetto indaga quali spazi colore sono più coerenti con la percezione umana

- il colore è uno degli strumenti principali della comunicazione visiva
- la percezione del colore è un processo estremamente complesso e non lineare
- comprendere come percepiamo i colori è essenziale per sviluppare strumenti digitali più efficaci

Stephen Few e i principi della percezione nelle visualizzazioni

Il colore è relativo: lo percepiamo sempre in funzione del contesto (sfondo, colori vicini).

1 – Evitare sfondi complessi, gradienti, texture e 3D: creano interferenze percettive.

2 – Scegliere la palette in base allo scopo:

- Categorica → per distinguere categorie diverse
- Sequenziale → per valori che vanno da basso ad alto
- Divergente → quando i valori si muovono rispetto ad un punto centrale

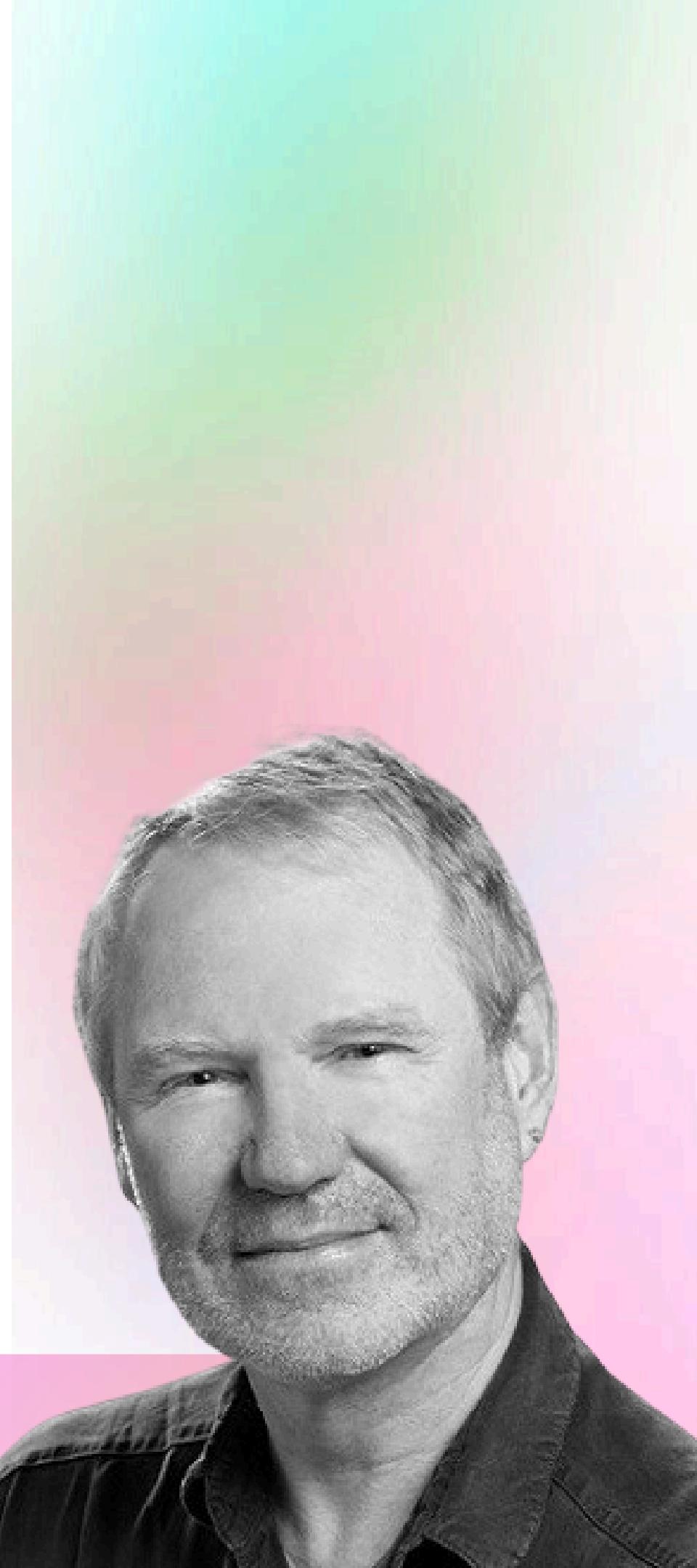
3 – Mantenere un equilibrio percettivo nelle palette: evitare colori che attirano l'attenzione

4 – Considerare l'accessibilità: evitare combinazioni problematiche per daltonismo.

Le sue regole nascono dall'idea che la visualizzazione non deve riflettere la matematica dei colori, ma la percezione umana

Tratto dall'articolo:

Practical Rules for Using Color in Charts

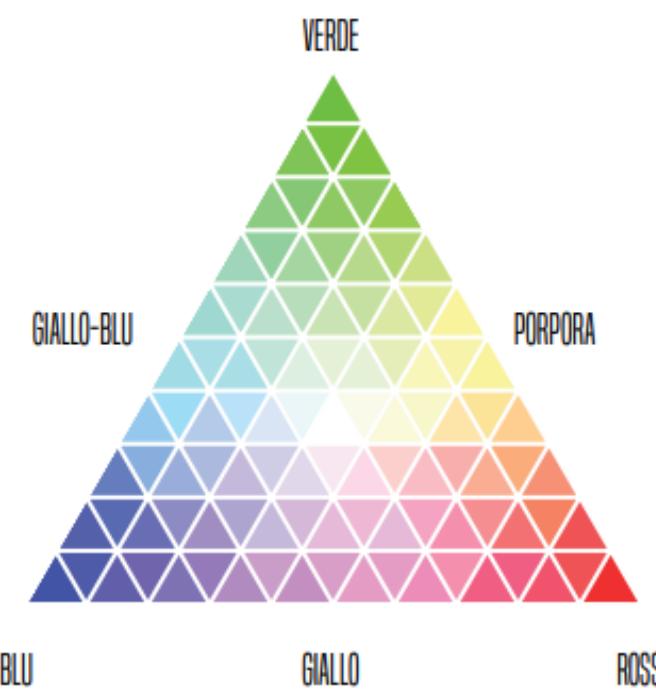
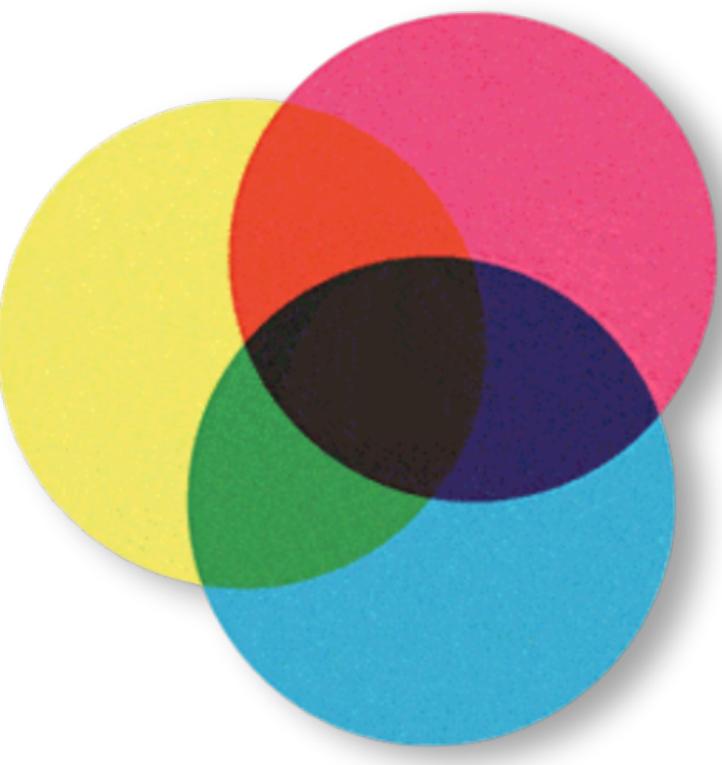


Spazio RGB

- RGB è uno spazio colore tecnico, non percettivo: le distanze numeriche tra colori non corrispondono alle distanze che noi percepiamo

- La sensibilità umana varia a seconda luminosità e tinta:
 - colori simili possono risultare lontani in RGB,
 - colori numericamente vicini possono apparire molto diversi.

- RGB non è uno spazio uniforme percettivamente
- Conseguenze pratiche: algoritmi come K-Means, che raggruppano i colori in base alla distanza, producono cluster che non riflettono la percezione umana.
- Da qui la necessità di spazi più percettivi come CIELAB e L*.



Spazio CIELAB

Per superare questi limiti, la CIE (Commissione Internazionale dell'Illuminazione) ha definito lo spazio CIELAB, progettato proprio per essere percettivamente uniforme

Le sue tre componenti sono:

- L^* = luminosità (lightness), da 0 a 100
- a^* = asse cromatico da verde a rosso
- b^* = asse cromatico da blu a giallo

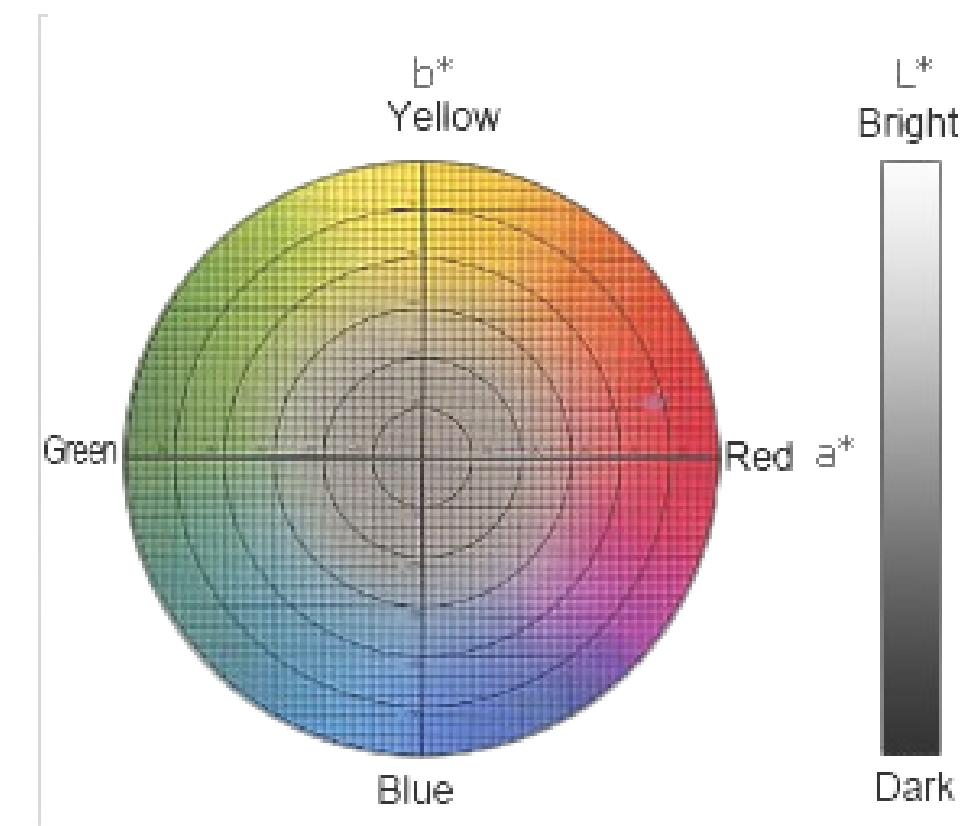
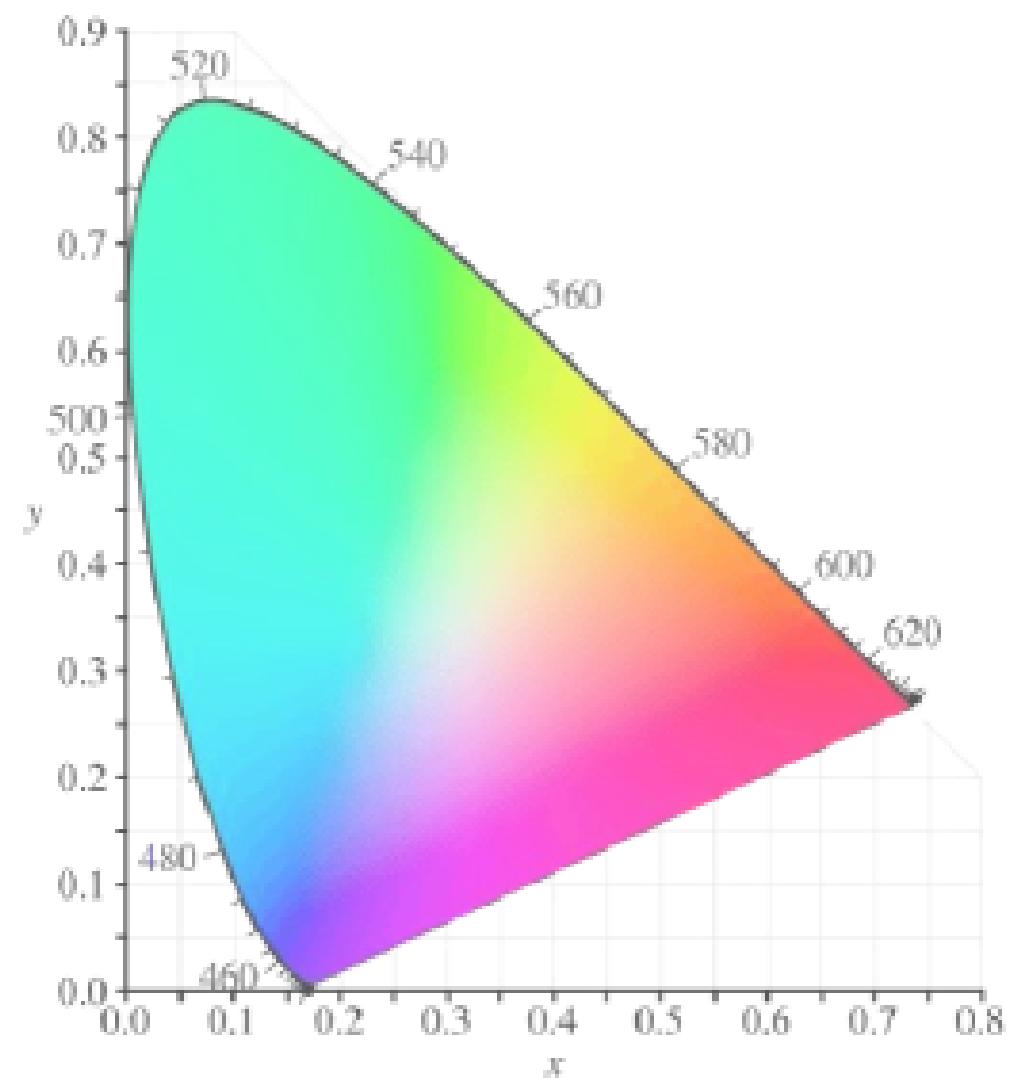
La distanza matematica tra due colori è molto più simile alla distanza percepita dal nostro occhio; per questo è uno degli spazi colore più usati in applicazioni di grafica, stampa, computer vision e per segmentazione basata sulla percezione umana.

Spazio L^*

Analizzata la luminosità.

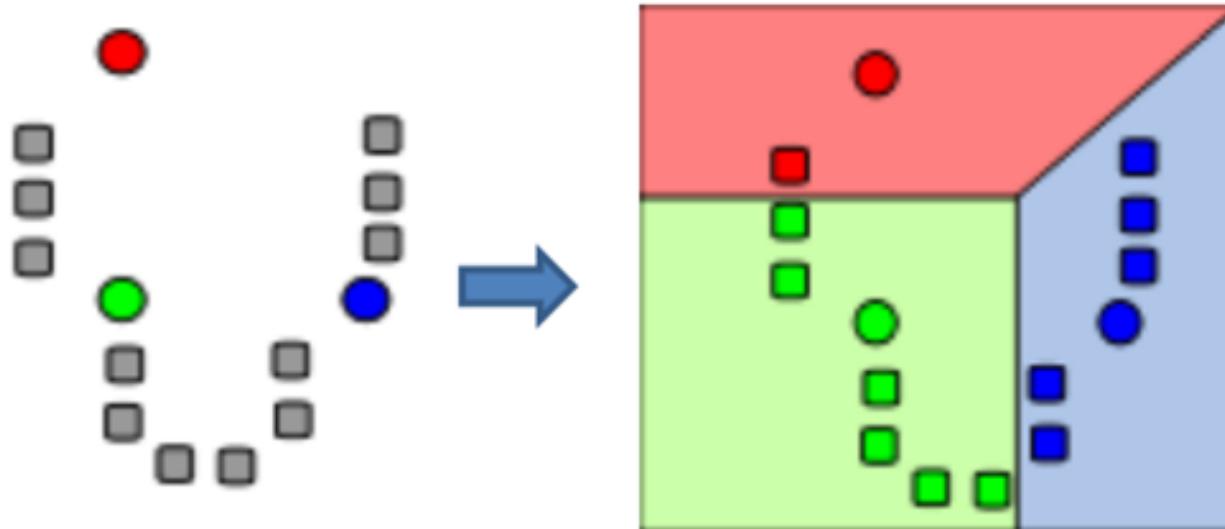
Il colore scompare completamente e l'immagine viene rappresentata in scala di grigi. Rimangono solo luce, ombra e contrasto.

Secondo molti studi percettivi, l'occhio umano è molto più sensibile alla luminosità che al colore. Quindi, anche se sembra di perdere informazione, in realtà si mette in evidenza la componente visiva più importante per il cervello umano, cioè la struttura luminosa dell'immagine.



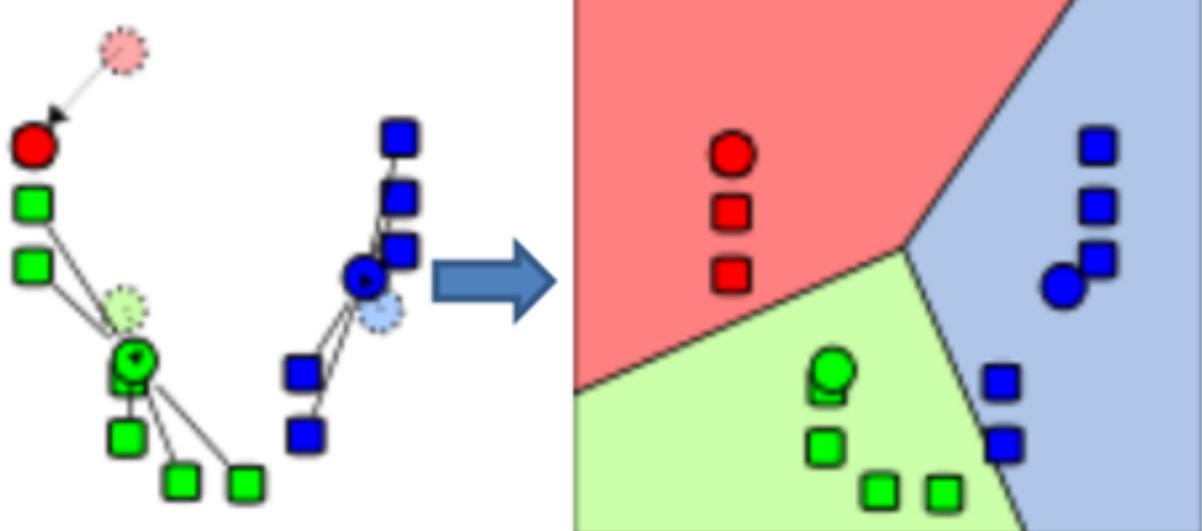
Algoritmo K-Means

Utilizzato per il clustering non supervisionato, cioè per raggruppare dati senza conoscere in anticipo le etichette o categorie. Il suo obiettivo è suddividere un insieme di dati in K gruppi (cluster), in modo tale da minimizzare la somma delle distanze tra ogni punto e il centro del cluster di appartenenza (detto centroide). La metrica più comune è la distanza euclidea.



Funzionamento dell'algoritmo

1. Inizializzazione: Si scelgono K centri iniziali, casuali o tramite metodi ottimizzati.
2. Assegnazione: Ogni punto viene assegnato al centro più vicino in base alla distanza.
3. Aggiornamento: Per ciascun cluster si ricalcola il nuovo centroide come media dei punti assegnati.
4. Iterazione: Assegnazione e aggiornamento si ripetono finché i centroidi non cambiano più o si raggiunge il limite di iterazioni.



Visualizzazione: Tutti i pixel appartenenti allo stesso cluster vengono mostrati con lo stesso colore, semplificando l'immagine.

SSE (Sum of Squared Errors)

K-Means mira a minimizzare la funzione SSE (Sum of Squared Errors).

La SSE è una misura che indica quanto i punti di un cluster sono vicini al loro centroide.

Si calcola sommando, per ogni punto, la distanza al quadrato dal centroide del cluster a cui appartiene.

Se la SSE è bassa, i punti sono ben raggruppati e il cluster è compatto

Se la SSE è alta, i punti sono sparsi e il cluster è poco definito

K-Means cerca di minimizzare la SSE, cioè di creare cluster più compatti e precisi.

formula calcolo SSE:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2$$

Dove:

x è un punto del cluster,
 μ_i è il centroide del cluster,
K è il numero di cluster.

Vantaggi e svantaggi dell'algoritmo

I vantaggi di questo algoritmo sono:

- è semplice e veloce, anche su dataset grandi
- è facile da implementare
- i risultati intuitivi e interpretabili.

Lo svantaggio è il dover decidere prima K, il numero di cluster; per scegliere il numero ottimale di K esistono diversi metodi. Nel progetto abbiamo utilizzato il Davies-Bouldin Index (DBI).

Davies-Bouldin Index (DBI)

E' un indice di valutazione del clustering che misura quanto i cluster sono compatti e quanto sono separati tra loro.

L'obiettivo del DBI è valutare quanto i cluster sono distinti e ben formati.

Un buon clustering dovrebbe avere:

- cluster compatti: i punti sono vicini tra loro e vicini al centroide
- cluster ben separati: lontani dai centroidi degli altri cluster

Minore è il DBI, migliore è il clustering.

Per ogni cluster, il DBI confronta:

- la dispersione interna al cluster (S_i): indica la coesione: quanto i punti di quel cluster sono vicini al suo centroide
- la distanza tra centroidi di cluster diversi (M_{ij}): indica la separabilità: quanto due cluster sono lontani tra loro.

Per ogni coppia di cluster i e j si calcola:

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}$$

Ovvero: più i cluster sono dispersi (grande S) e vicini (piccolo M), più il clustering è peggio.

Poi, per ogni cluster si prende il valore peggiore (massimo R_{ij}), e si fa la media di tutti questi valori:

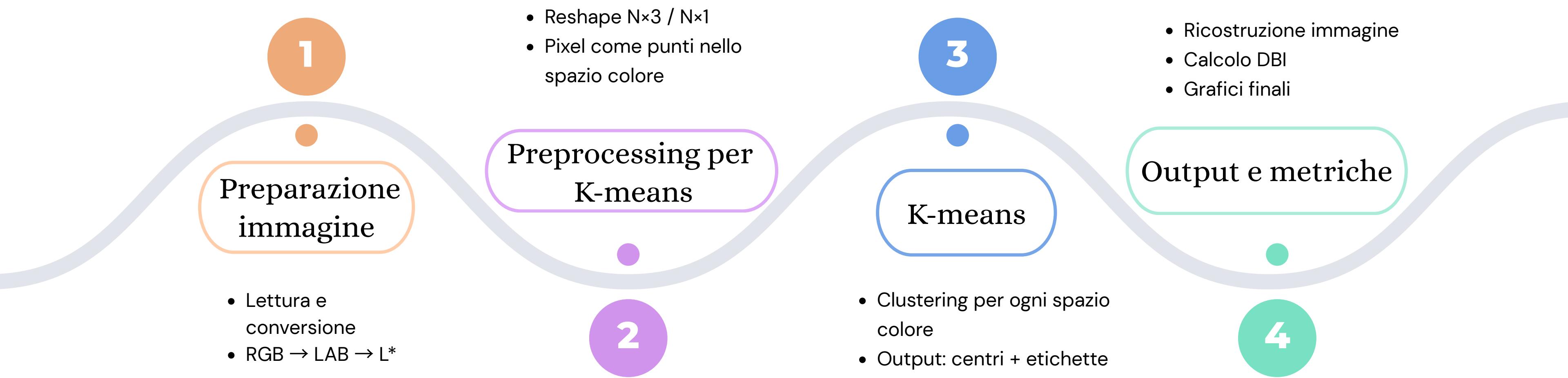
$$DBI = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} R_{ij}$$

Più basso è l'indice, migliore è il clustering.

Per scegliere K:

- eseguire K-Means con diversi valori di K (ad esempio: da 2 a 10)
- calcolare il DBI per ogni K
- scegliere il K con il DBI più basso

Il codice



Scelta del dataset

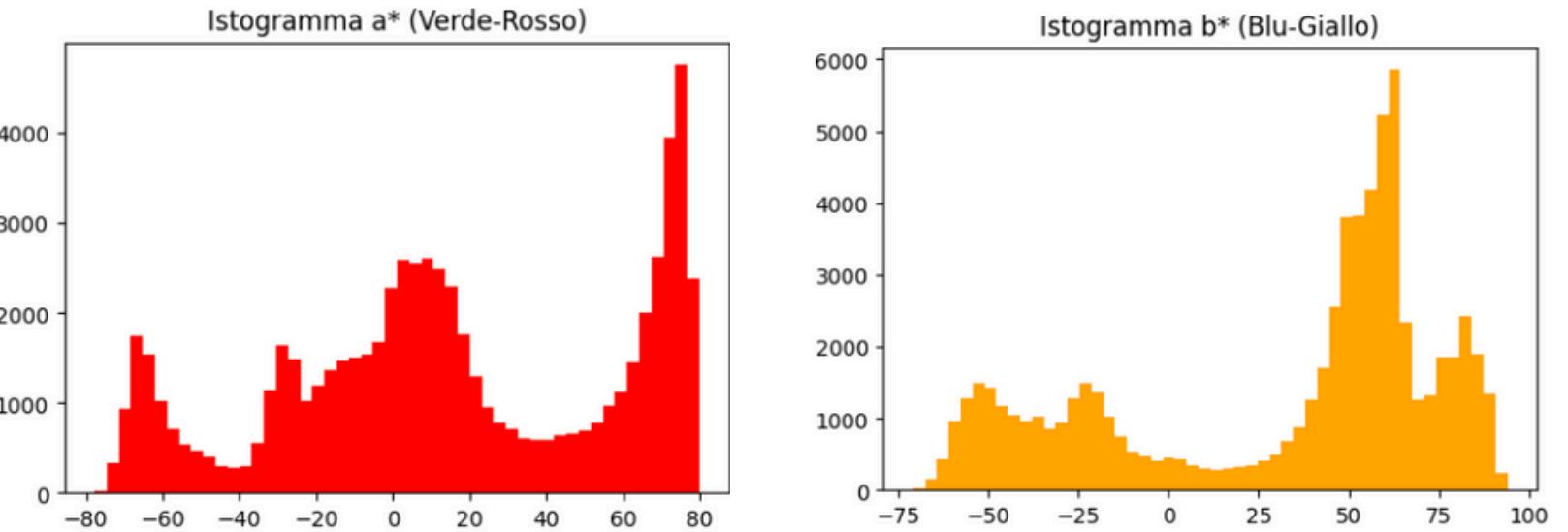
Perchè scegliere l'immagine balls1

- Colori saturi e distinti: le palline hanno tonalità molto diverse e ben separate
- Cluster compatti: ogni pallina ha una bassa varianza interna quindi il DBI può misurare correttamente la qualità dei cluster
- Istogrammi con picchi netti: colori separati chiaramente facilitano la segmentazione
- Ridotto rumore cromatico: non presenta ombre, texture complesse o illuminazioni variabili

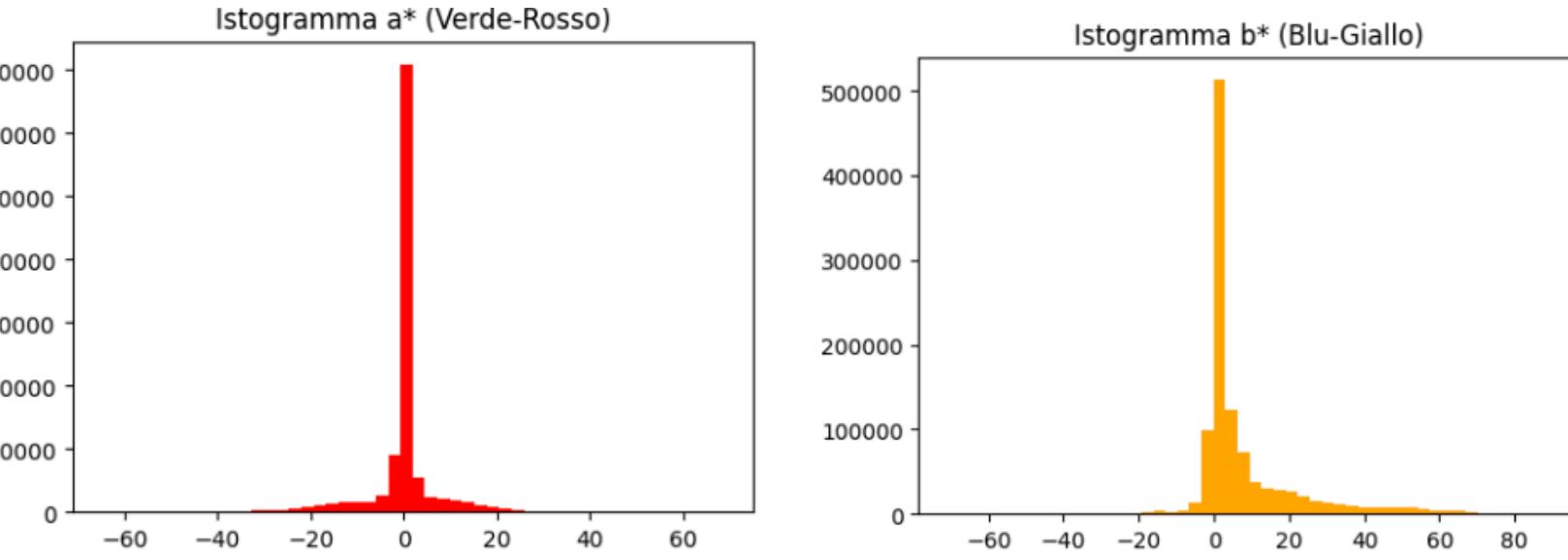
Immagine scelta: "balls1"



Distribuzione Originale dei Canali CIELAB



Esempio immagine scartata: "road"



Distribuzione dei canali CIELAB

A cosa serve

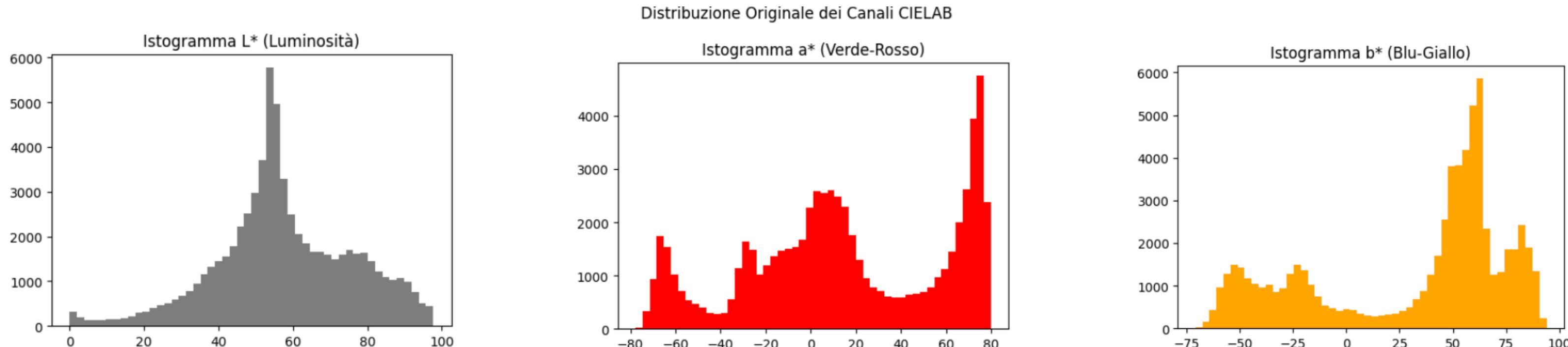
- Per capire come sono distribuiti i colori nei canali L^* , a^* , b^* .
- Per vedere se i pixel formano gruppi naturali utili al K-means.
- Per valutare la separabilità cromatica, che influisce sul DBI.

Cosa indica

- L^* : la maggior parte dei pixel ha luminosità media.
- a^* : picchi separati \rightarrow molti rossi e verdi distinti.
- b^* : picchi verso il giallo e il blu \rightarrow molti gialli e arancioni, colori molto saturi.

Perché è rilevante

- Picchi chiari e separati = cluster ben definiti nello spazio LAB \rightarrow segmentazione più efficace.



K-means

```
# --- FUNZIONE CLUSTERING E VALUTAZIONE ---
def cluster(data):
    """Esegue K-means e calcola il Davies-Bouldin Index (DBI)."""
    kmeans = KMeans(n_clusters=K, random_state=42, n_init="auto").fit(data)
    labels = kmeans.labels_
    centers = kmeans.cluster_centers_
    # DBI: indice di compattezza e separazione (meno è meglio)
    dbi = davies_bouldin_score(data, labels)
    return labels, centers, dbi

# --- ESECUZIONE CLUSTERING ---
results = {}

for name, data in {
    'RGB': pixels_rgb,
    'LAB': pixels_lab,
    'L*': pixels_l
}.items():
    labels, centers, dbi = cluster(data)
    results[name] = (labels, centers, dbi)
    print(f"Risultato {name}: DBI = {dbi:.4f}")
```

Calcoliamo K-means e DBI

- Applichiamo K-means ai pixel (in RGB, LAB o L*) per raggruppare i colori simili e ottenere una segmentazione dell'immagine.
- Usando esattamente lo stesso algoritmo e lo stesso K in tutti gli spazi colore, possiamo confrontare in modo corretto quale rappresentazione produce cluster più naturali e più separati.
- Ci aspettiamo un DBI più basso in CIELAB, con cluster più compatti e separati rispetto a RGB.

Distribuzione dei centri

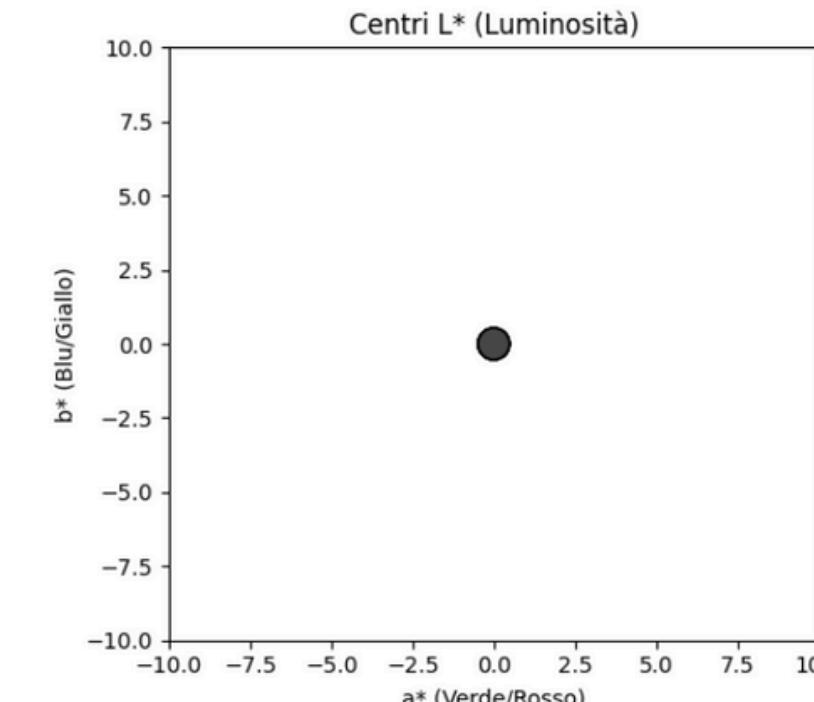
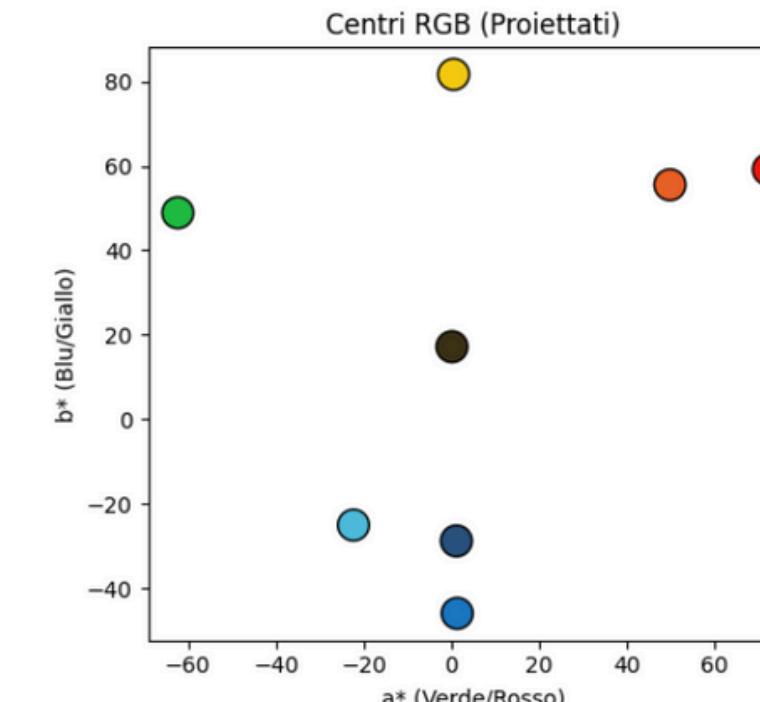
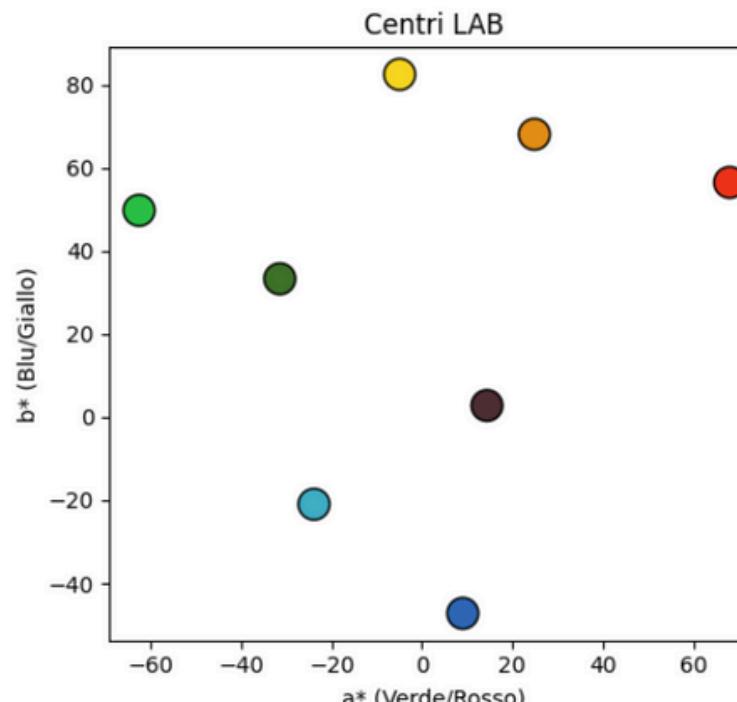
Perché visualizzarli

- Permettono di capire dove cadono i colori medi dei cluster nello spazio cromatico (a^*b^*)
- Aiutano a valutare la separazione cromatica tra i gruppi: cluster ben distanziati indicano colori nettamente diversi
- Offrono un supporto visivo alla qualità del clustering, complementare al DBI

Cosa indicano i centroidi

- In LAB: i centri sono su colori saturi ben distinti (rosso, verde, blu, scuro), mostrando cluster cromatici ben definiti
- In RGB (proiettati in LAB): i centri sono più raggruppati, indicando che alcune differenze cromatiche non sono ben separate quando l'algoritmo lavora in RGB
- In L*: tutti i centroidi cadono nell'origine a^*b^* , mostrando che il clustering usa solo la luminosità senza distinguere i colori

Distribuzione dei Centri dei Cluster (8) sul Piano Cromatico a^*b^*



Ricostruzione immagine

```
# --- FUNZIONE RICOSTRUZIONE IMMAGINE ---
def reconstruct(labels, centers, mode):
    """Ricostruisce l'immagine segmentata usando i centri dei cluster."""
    if mode == 'RGB':
        img_rec = centers[labels].reshape(H, W, C)
    elif mode == 'LAB':
        lab = centers[labels].reshape(H, W, C)
        img_rec = color.lab2rgb(lab)
    else: # L* (Scala di Grigi)
        lab = np.zeros(len(labels), 3)
        lab[:, 0] = centers[labels].flatten()
        img_rec = color.lab2rgb(lab.reshape(H, W, C))
    return (np.clip(img_rec, 0, 1) * 255).astype(np.uint8)
```

Scopo

- Dopo il clustering, ogni pixel ha un'etichetta che indica a quale cluster appartiene (labels).
- La ricostruzione serve a visualizzare la segmentazione, sostituendo ogni pixel con il colore medio del suo cluster (centers).

Come funziona

- RGB: ogni pixel viene sostituito dal centro RGB del cluster.
- LAB: si sostituisce ogni pixel con il centro LAB, poi si converte in RGB per visualizzazione.
- L*: si costruisce un LAB con $L^* = \text{centro del cluster}$ e $a^*=b^*=0 \rightarrow$ poi conversione in RGB \rightarrow immagine in scala di grigi.

Risultati

- L^* DBI = 0.5189 → ideale per segmentazione in scala di grigi.
- LAB DBI = 0.7286 → ideale per segmentazione basata sul colore.
- RGB DBI = 0.9557 → meno efficace per colori saturi e distinti.

Perché si usa RGB e non CIELAB nei computer

- Corrispondenza diretta con i display: RGB rappresenta direttamente i tre canali di luce (rosso, verde, blu) che uno schermo può emettere.
- Evita conversioni complesse: CIELAB è uno spazio percettivo non lineare e richiede trasformazioni per essere visualizzato su schermi digitali.

