

Università di Parma
Corso di Laurea Magistrale in Ingegneria Informatica
Fondamenti di Visione Artificiale
a.a. 2018/19

PROVA PRATICA 19-06-2019

NOME:

COGNOME:

MATRICOLA:

WORKSTATION N°:

Non è consentito scambiarsi materiale via rete (ovviamente).

E' consentito l'uso di funzioni OpenCv.

Salvare l'esame in un file COGNOME_MATRICOLA.zip.

FIRMA

ES1

Vi viene fornita una cartella "images" contenente un piccolo database di immagini.

Scrivere un codice che legga un'immagine dal database (immagine *originale*) e individui quella piu' simile tra tutte le altre presenti nel database, utilizzando **K-Means**

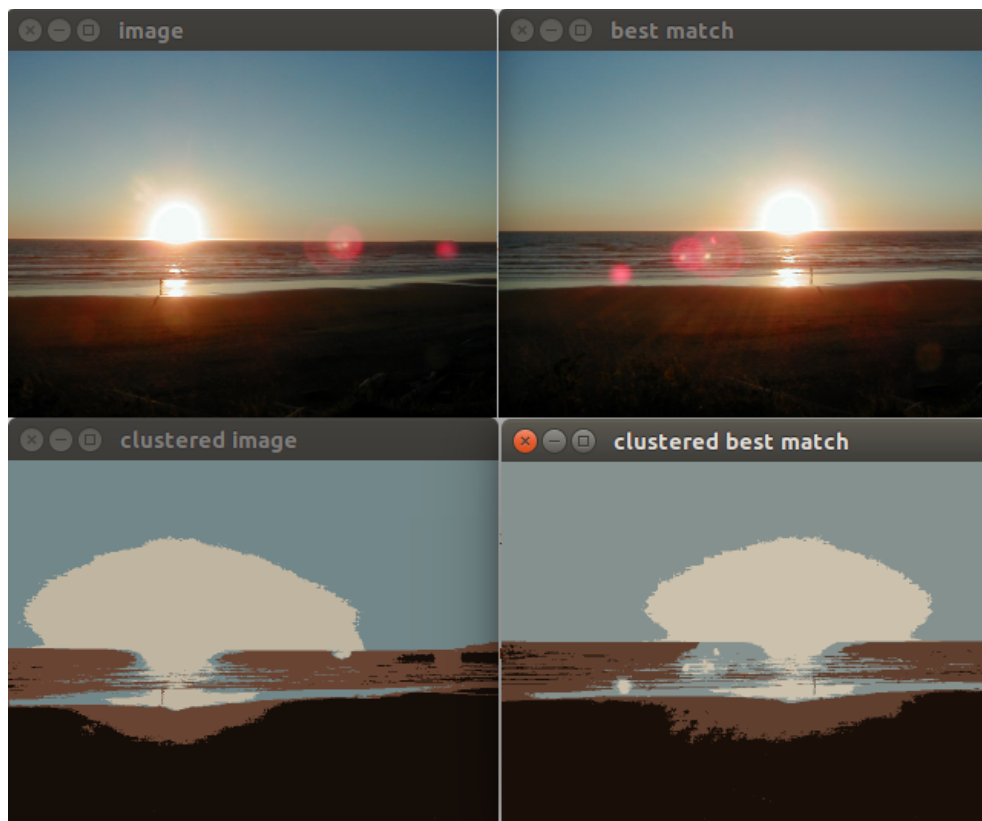
APPROCCIO

Possiamo dividere l'algoritmo in due distinte fasi:

1. calcolare i cluster dell'immagine originale tramite K-Means
2. per ogni immagine nel database (esclusa ovviamente quella originale), calcolare i corrispondenti cluster e confrontarli con quelli dell'immagine originale
3. stabilire quale immagine database assomiglia di piu' a quella originale sulla base del confronto tra cluster

SUGGERIMENTI

1. E' consentito e suggerito di utilizzare K-Means di **OpenCV** (si veda l'esempio ad esercitazione)
2. In cosa consistono i cluster ottenuti da K-Means? Trovare una **SEMPLICE** metrica per confrontare i cluster.
3. Potrebbe essere utile confrontare cluster di dimensioni simili.
4. E' fornita una struttura *cluster* che puo' essere utile per raccogliere le informazioni dei vari cluster ed eventualmente fare ordinamenti o altre operazioni



NOTE E COMMENTI

Questo approccio non funziona molto bene per la ricerca di immagini, per varie ragioni.

Prima di tutto K-Means genera cluster di pixel non adiacenti. Il fatto che in un'immagine ci possano essere, poniamo, 100 pixel molto bianchi non ci da nessun tipo di informazione sul contenuto di questa immagine. E' la vela di barca? E' neve su una montagna? E' semplicemente una zona satura dell'immagine? Sono tante piccole patch molto chiare, o una grande patch contigua?

Solo utilizzando K-Means e' impossibile definire queste caratteristiche, di fatto i cluster generati sono molto poco caratterizzanti.

Quale poteva essere un metodo forse piu' robusto? Magari utilizzare dei key-point, per ottenere delle feature piu' robuste? Forse, ma avremmo perso un po' il concetto di "classe" di immagini (spiaggia, lago, barca, etc.), focalizzandoci su alcuni punti salienti di una specifica immagine.

Questo e' un problema che va affrontato con tecniche di MACHINE LEARNING per essere risolto correttamente in un caso pratico, ma esula decisamente dal corso di quest'anno.

SOLUZIONE E NOTE

Come detto, si trattava di applicare K-Means all'immagine originale e a tutte le rimanenti nel database, confrontando poi le caratteristiche dei cluster.

K-Means fornisce il centro di massa per ogni cluster, in questo caso 3 valori corrispondenti alla media di R,G e B, essendo le immagini a colori. Una semplice somma delle differenze in valore assoluto dei centri di massa era una metrica già sufficiente per ottenere risultati ragionevoli nei casi semplici.

K-Means sceglie i centri dei cluster a caso, quindi alla fine potremmo avere la configurazione indicata in figura:



E' evidente che in questo caso non sarebbe sufficiente confrontare i cluster nel loro ordine così come vengono generati da K-Means, perché otterremmo errori molto grandi anche in presenza di immagini simili, andando a confrontare cluster incoerenti.

La soluzione più corretta sarebbe confrontare tutti con tutti, e scegliere la configurazione di associazioni che porta all'errore minore. In questo caso, ad esempio sarebbe (0→1 1→0 2→3 3→2 4→3). Questo tipo di associazione ottima tra due insiemi di elementi per i quali esiste una funzione di confronto si può realizzare con l'algoritmo Ungherese, ma esula decisamente dall'ambito dell'esame. (https://it.wikipedia.org/wiki/Algoritmo_ungherese)

Una soluzione intermedia (che poteva valere la lode) era semplicemente quella di ordinare i cluster per dimensione, in modo da minimizzare la probabilità di confronti incoerenti. Ad esempio, il cluster 0 della prima immagine è circa la metà del cluster 0 della seconda immagine. Un semplice ordinamento li avrebbe probabilmente scambiati nel vettore dei cluster, rendendo più probabile ottenere un errore basso.