

**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**

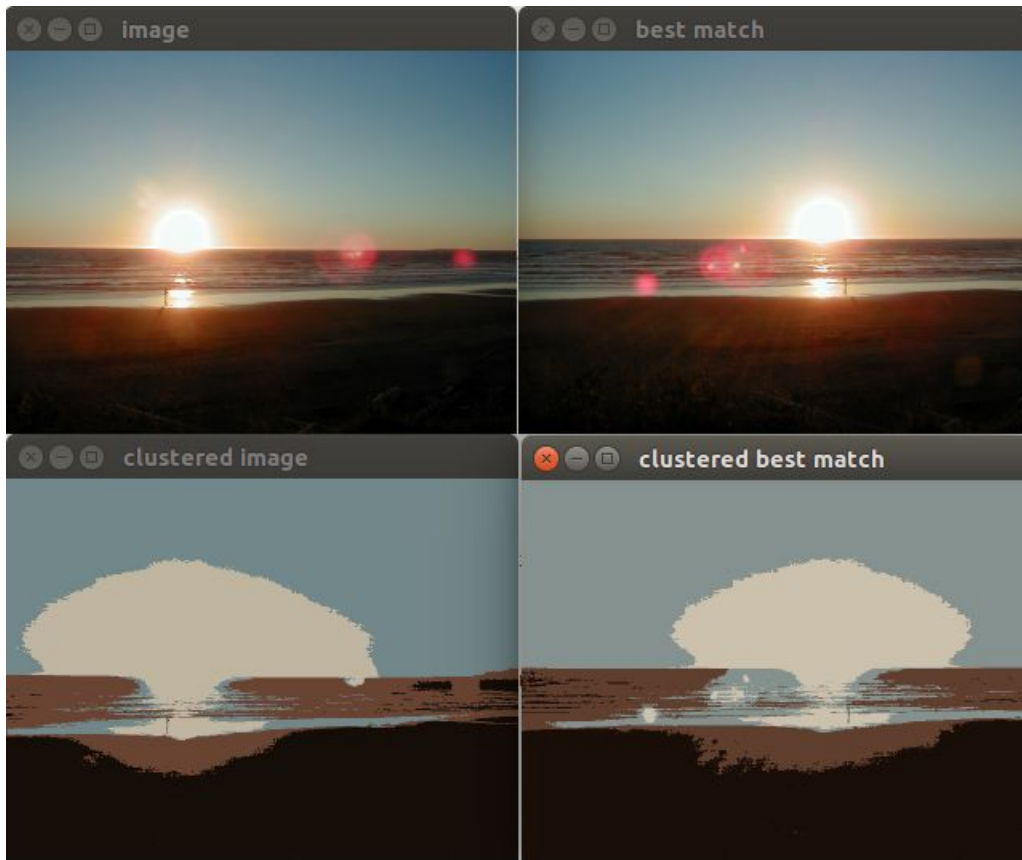
### APPROCCIO

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.