

K-Means Clustering

Descriere proiect

Scopul acestui proiect este realizarea segmentarii unei imagini si clasificarea acesteia in mai multe multe grupuri utilizand metoda K-Means clustering. Noi trebuie sa realizam clusterizarea imaginilor color utilizand mai multe metriki (euclidiana, cosinus, L1 - Manhattan).

Segmentarea imaginii este procesul de partitie a unei imagini digitale in mai multe regiuni distincte care contin fiecare pixel (seturi de pixeli, cunoscute si sub numele de superpixeli) cu atribute similare. Rolul segmentarii imaginii este de a schimba reprezentarea acesteia in ceva mai semnificativ si mai usor de analizat. Acesta este procesul de atribuire a unei etichete fiecarui pixel dintr-o imagine, astfel incat pixelii cu aceeași etichetă au anumite caracteristici.

Algoritmul de clustering K-Means este un algoritm nesupraveghet și este utilizat pentru a segmenta zona de interes de pe fundal. Acesta grupează sau compartează datele date în K-grupuri sau părți bazate pe K-centroids.

Pași în algoritmul K-Means:

1. Alegem numărul de clustere K
2. Selectam la puncte K aleatorii, centroizii
3. Alocam fiecare punct de date celui mai apropiat centroid → care formează clusterul K
4. Calculam și așezam noul centroid al fiecarui cluster
5. Reasignam fiecare punct de date către noul centroid cel mai apropiat. Daca a avut loc vreo reasignare, ne intoarcem la pasul 4, in caz contrar, modelul este gata.

Am construit o structura Punct ce contine un vector pentru a reprezenta coordonatele x, y, R,G,B, H,S,V dar si un cluster intreg care este o modalitate de reprezenta ca un punct apartine unui cluster specific.

Functia extractFeatures are rolul de a extrage int-un vector, toate caracteristicile punctelor din imagine (x, y, R,G,B, H,S,V). Functia kMeansClustering realizeaza algoritmul K-means si are ca parametrii vectorul de puncte determinat anterior, numarul de clustere, numarul de repetitii, weight-ul ales, eroarea si distanta pe care o vom aplica(Euclidian, Cosinus sau Manhattan). Cu cat numarul de repetitii este mai mare, solutia finala va fi mai buna. Daca distanta dintre un punct si un cluster curent este mai mica decat distanta dintre acest punct si clusterul anterior, actualizam punctul pentru a face parte din clusterul current.

Dupa prima iteratie punctele nu vor fi distribuite in mod egal fiecarui grup, asadar trebuie sa existe o sau doua parte in care se calculeaza noii centroizi, realizat prin apelarea metodei computeCentroids. ComputeCentroids itereaza peste puncte pentru a face legatura intre date(puncte) si centroizi si construieste noii centrozi.

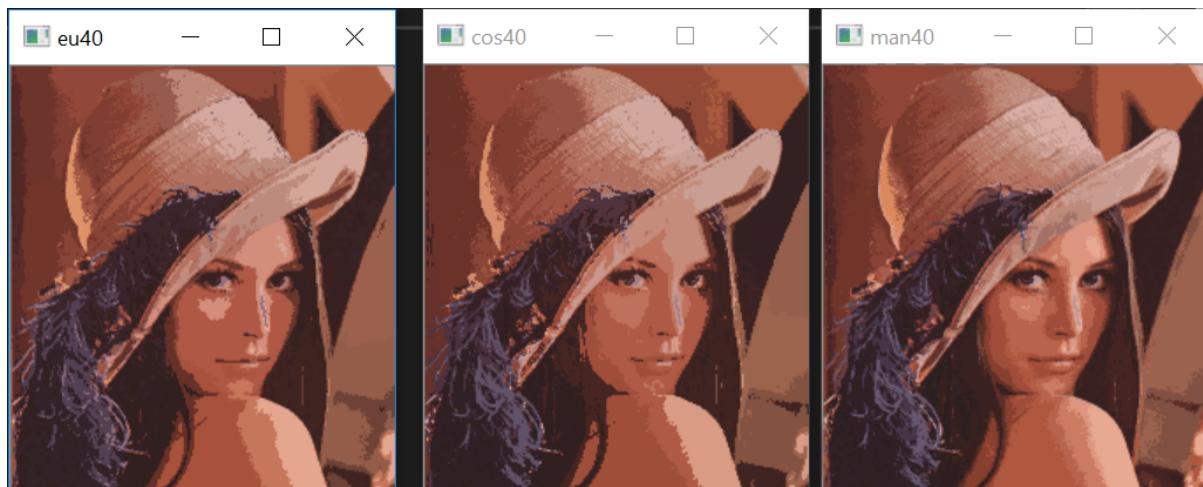
Pentru afisarea rezultatului folosim functia generateKMeansResult si ne vom folosi de imaginea originala, de vectorul de puncte cu caracteristicile si de un vector de puncte cu centroizii calculate cu kMeansClustering(). Aici vom construi imaginea destinatie si o vom afisa in terminal. Pentru calcularea distantei vom folosi 3 distante: Euclidian, Manhattan si Cosinus. Toate functiile vor avea nevoie de 2 puncte si de vectorul de ponderi constante.

Distanta Euclidiană măsoară lungimea unui segment care leagă cele două puncte. Distanta Manhattan reprezintă distanța dintre două puncte dintr-o grilă bazată pe o cale strict orizontală și / sau verticală (adică de-a lungul liniilor de grilă). Similaritate cosinus este folosită în general ca metrică pentru măsurarea distanței atunci când magnitudinea vectorilor nu contează. Aceasta are valori între -1 și 1, -1 însemnând că nu există similaritate între cei 2 vectori, 1 însemnând că sunt exact la fel.

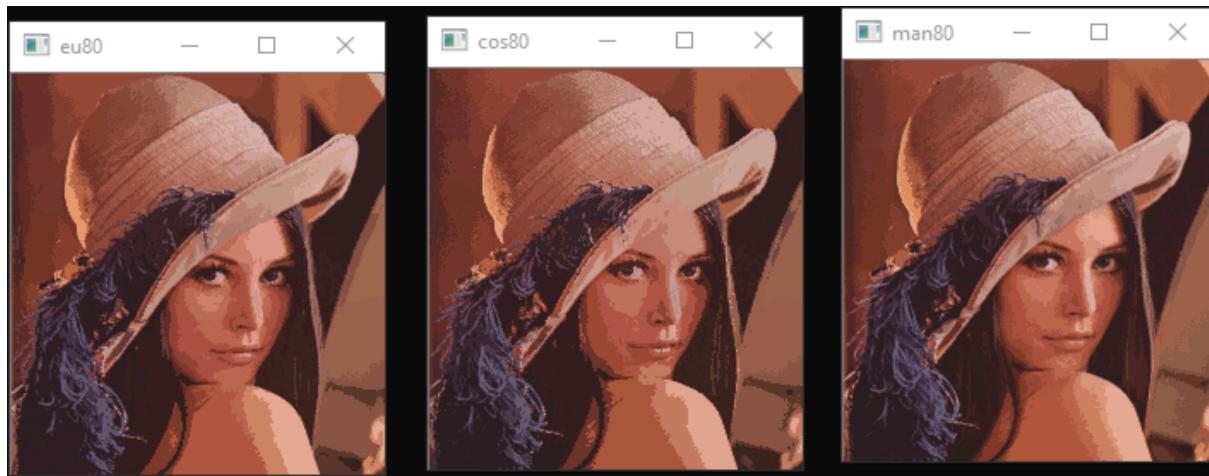
Imagine sursă pe care am facut testele:



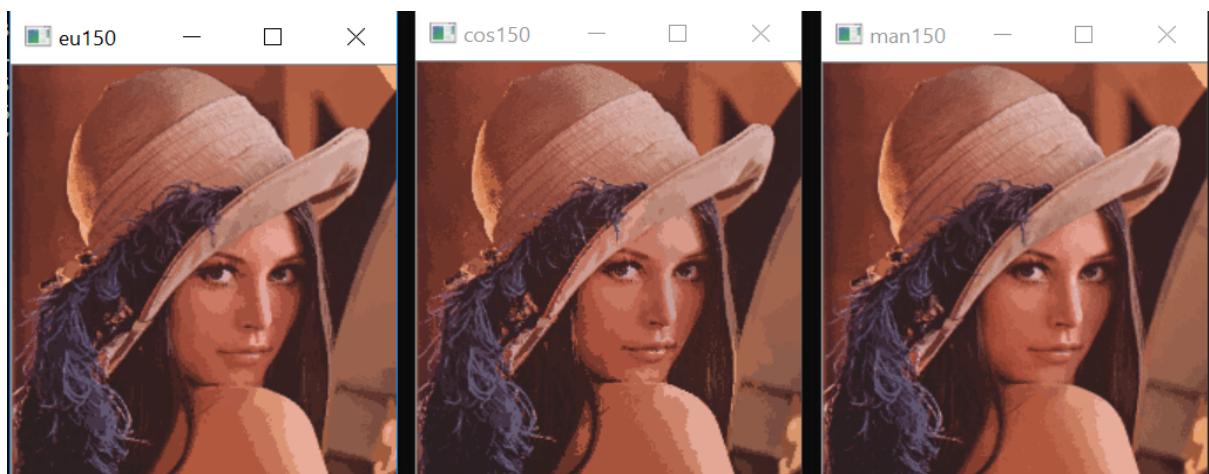
Imaginiile rezultate, corespunzătoare distantei euclidiene, cosine similarity și distantei Manhattan, pentru K=40 clustere:



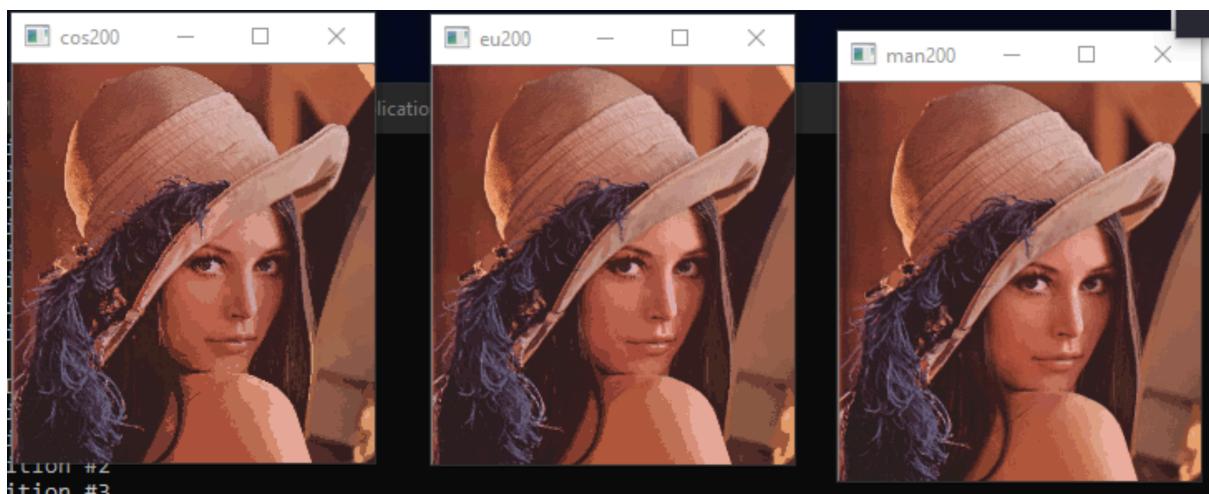
Imaginiile rezultate, corespunzatoare distantei euclidiene, cosine similarity si distantei Manhattan, pentru K=80 clustere:



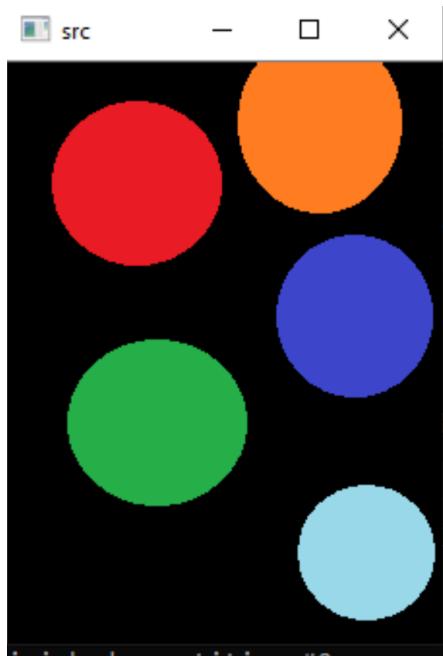
Imaginiile rezultate, corespunzatoare distantei euclidiene, cosine similarity si distantei Manhattan, pentru K=150 clustere:



Imaginiile rezultate, corespunzatoare distantei euclidiene, cosine similarity si distantei Manhattan, pentru K=200 clustere:



Imagine sursa pe care am facut testele:



Imaginiile rezultate, corespunzatoare distantei euclidiene, cosine similarity si distantei Manhattan, pentru K=5 clustere:

