

- Tema 2 MN -

Task 1: Functia primeste ca parametru cale catre imagine si numarul valorilor singular. Imaginea este citita si stocata sub forma de matrice. Se utilizeaza functia din Octave svd pentru determinarea matricilor U, S si v corespunzatoare imaginii citite. Utilizand formula data matricea initiala este aproximata de matricea A_k .

Task2:

- graficul 1: Se determina matricea diagonal S cu ajutorul functiei svd. Din matrice diagonal este convertita intr-un vector cu ajutorul functiei diag. In final este construit graficul cu functia plot.
- graficul 2: Se declara vectorul k asa cum a fost dat in cerinta si este calculate suma valorilor singulare din matricea S. Lungimea vectorului k, data de functia length, da numarul de interval ce vor fi parcurse pentru a se construi vectorul (valabil si pentru graficul 3, 4 si task-ul 5). Pentru fiecare k este calculata suma valorilor singular. In vectorul X este stocat raportul dintre suma specifica fiecarui k si suma tuturor valorilor singular. In final este construit graficul in functie de vectorii k si X.
- graficul 3: Asemănător algoritmul de la graficul 2, pentru fiecare element din vector este determinată matricea A_k cu ajutorul functiei de la task-ul 1. Este calculate suma si apoi raportul după formula data, rezultatul obținut la pentru fiecare element din k este stocat in vectorul X.
- graficul 4: Pentru fiecare k este calculate rata de compresie, rezultatul fiind stocat in vectorul X.
-

Task3, Task4: Functiile de la aceste doua task-uri sunt realizate si respecta pasii prezentati in cerinta.

Task5:

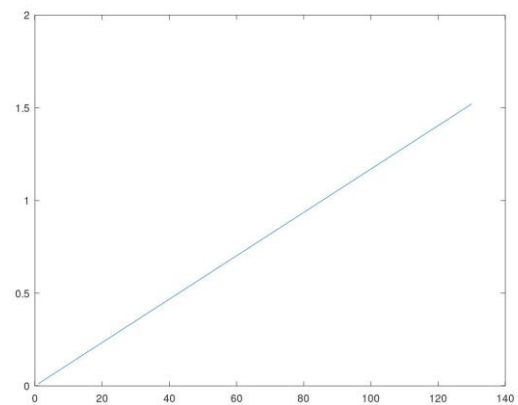
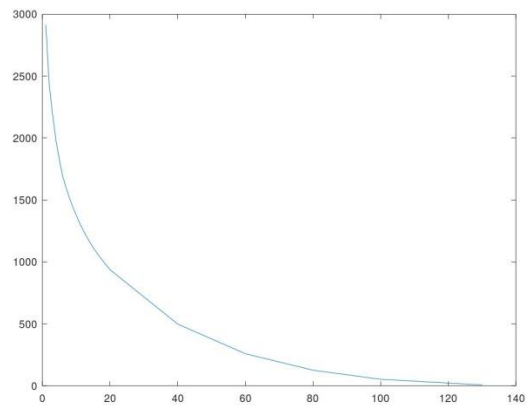
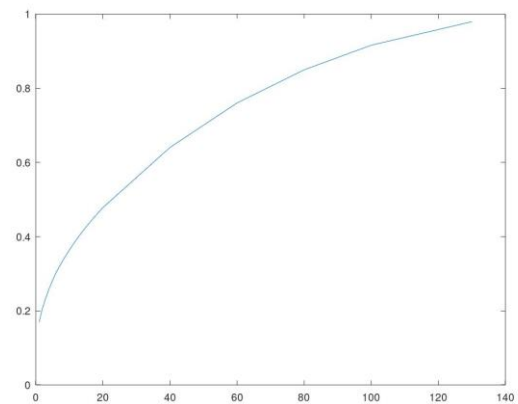
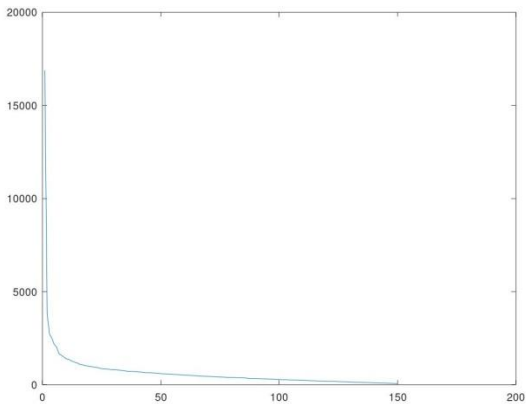
- graficul 1: Matricea diagonal S este obtinuta in urma apelarii task-ului 3, apoi transformata in vector coloana cu functia diag. In final este construit graficul in functie de vectorul rezultat.
- graficul 2 si 3 sunt construite la fel ca la taskul2, insa matricele S si A_k sunt obtinute cu ajutorul functiei de la task-ul 3.

- graficul 4: Pentru fiecare k este calculate raportul dat, rezultatul fiind stocat in vectorul X.

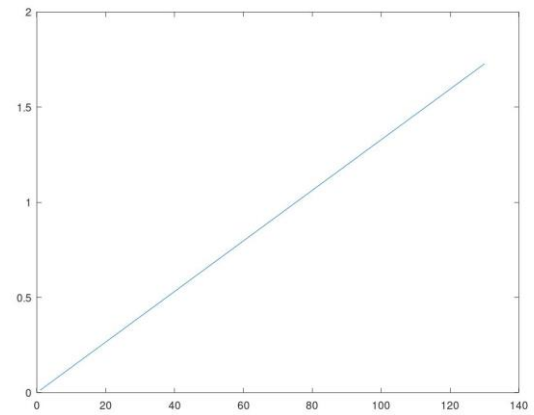
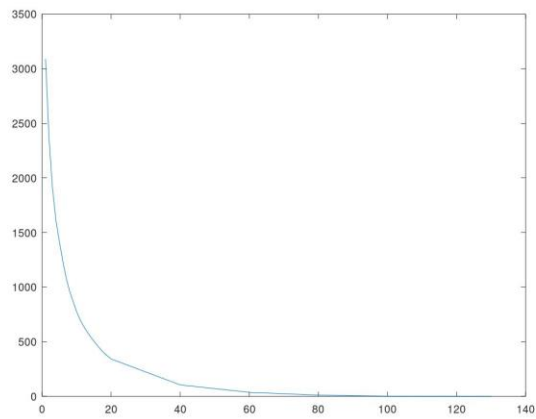
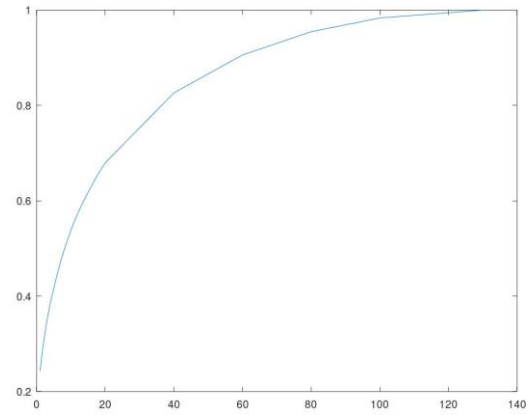
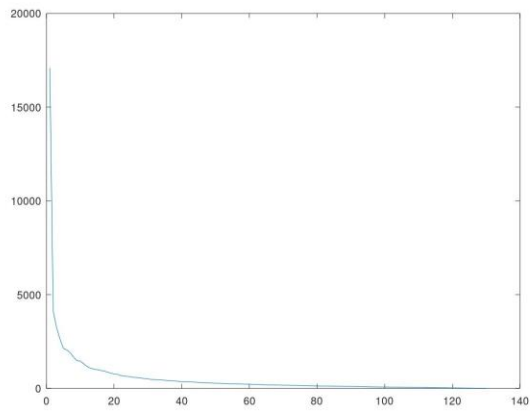
Task6: In functia `eigen_face_core` sunt citite pe rand cele 10 imagini din folder si transformate din matrici in vector coloana cu functia `reshape`. Cu acesti vectori coloana este construita matricea T. Pentru fiecare linie din matricea T se calculeaza media ce este retinuta in vectorul m. Pentru matricea Z create se calculeaza valorile si vectorii proprii. Din matricea V sunt eliminate acele coloane ce contin valorile proprii mai mici ca 1. In final sunt determinate matricele necesare. Functia `face_recognition` primeste ca parametru o imagine ce este citita si transformata in vector coloana din care este scazut vectorul mediilor m, calculat la functia anterioara. Se calculeaza proiectia imaginii test dupa cum este descrie in cerinta si mai apoi se determina distant minima dintre proiectia imaginii test si proiectiile imaginilor calculate la functia anterioara. Se retine indicele imaginii ce a indeplinit conditia ceruta.

- Grafice task2:

- Imaginea 2:

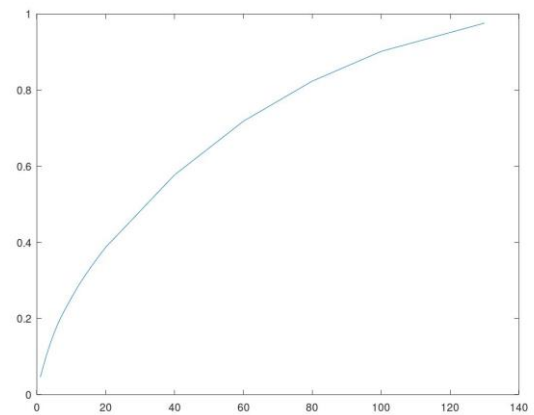
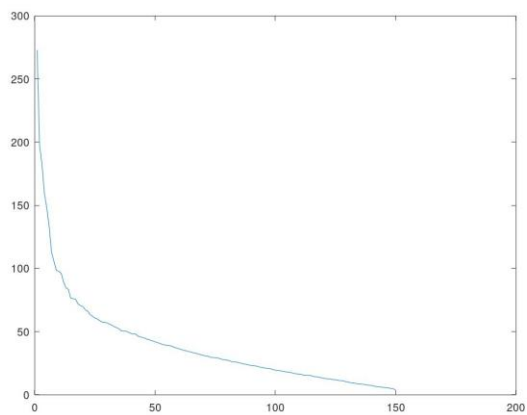


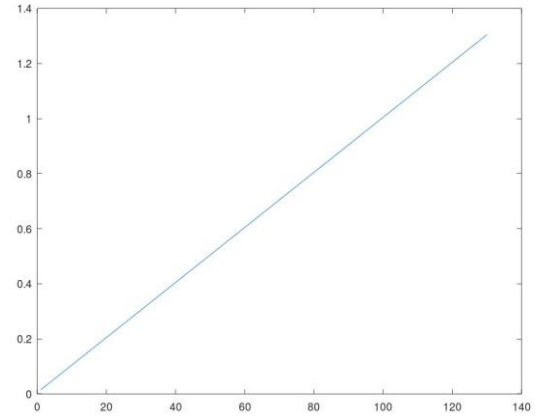
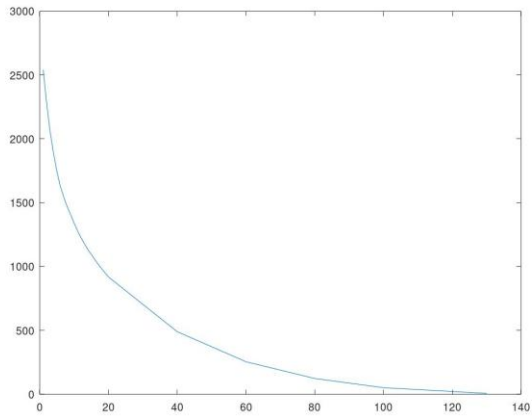
- Imaginea 3



• Grafice task5:

- Imaginea 2:





- Imaginea 3

