Metode Numerice

Tema 2

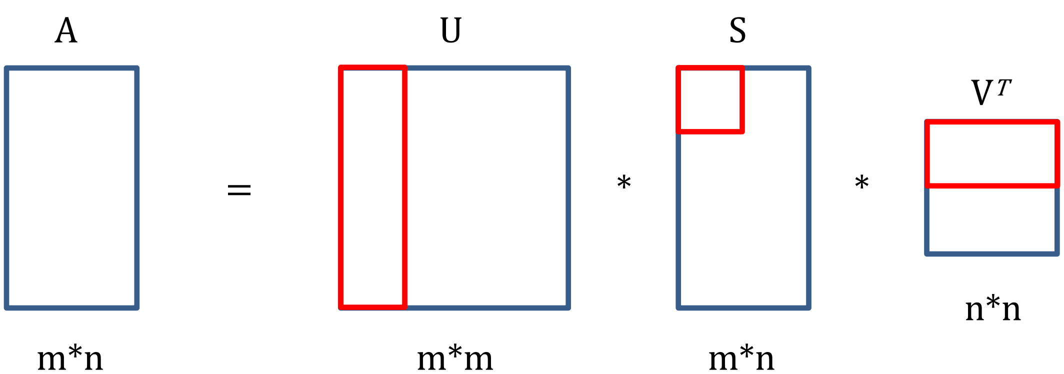
Bogdan-Andrei Buga,

grupa 312CB

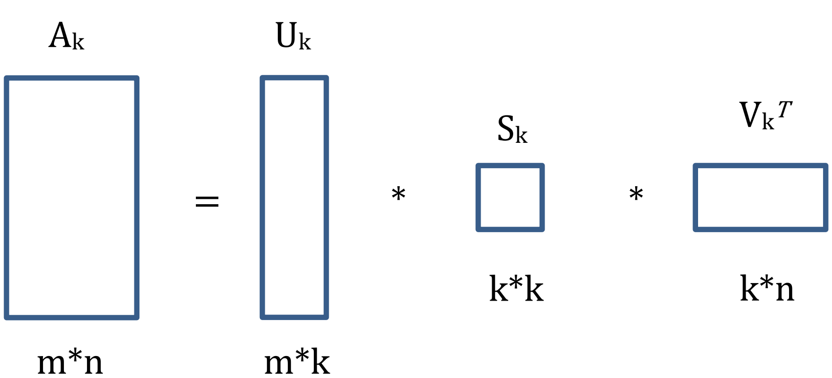
Task 1

Avand la dispozitie o matrice A din $\boldsymbol{R}^{m*n}$, o descompunem dupa valorile singulare astfel: $A = U * S * V^T$, unde U este din $\boldsymbol{R}^{m*m}$, S este din $\boldsymbol{R}^{m*n}$, iar V este din $\boldsymbol{R}^{n*n}$. Dupa efectuarea descompunerii si citirea parametrului k, generam matricele U_k, S_k si V_k (marcate cu rosu) astfel:

- $U_k(i, j) = U(i, j)$, (i, j) fiind din (1:m, 1:k);
- $S_k(i, j) = S(i, j)$, (i, j) fiind din (1:k, 1:k);
- $S_k(i, j) = S(i, j)$, (i, j) fiind din (1:n, 1:k);



Matricea returnata este $A_k = U_k * S_k * V_k^T$.



Task 2

(*OBSERVATIE: Imaginea testata in fisierul 'task2.m' este './in/images/image1.gif'*)

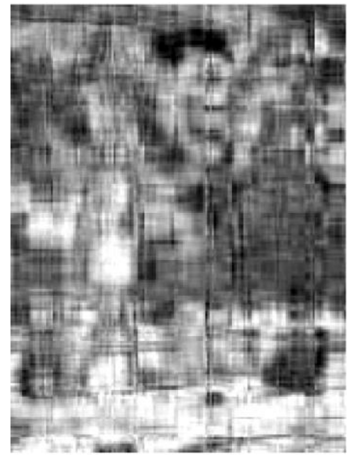
Dupa citirea matricei A corespunzatoare unei imagini citite in functie si *DVS*, alcatuim 4 grafice astfel incat pe axa Ox sa avem numere naturale de la 1 la min(m, n), iar pe axa Oy sa avem f(k) = :

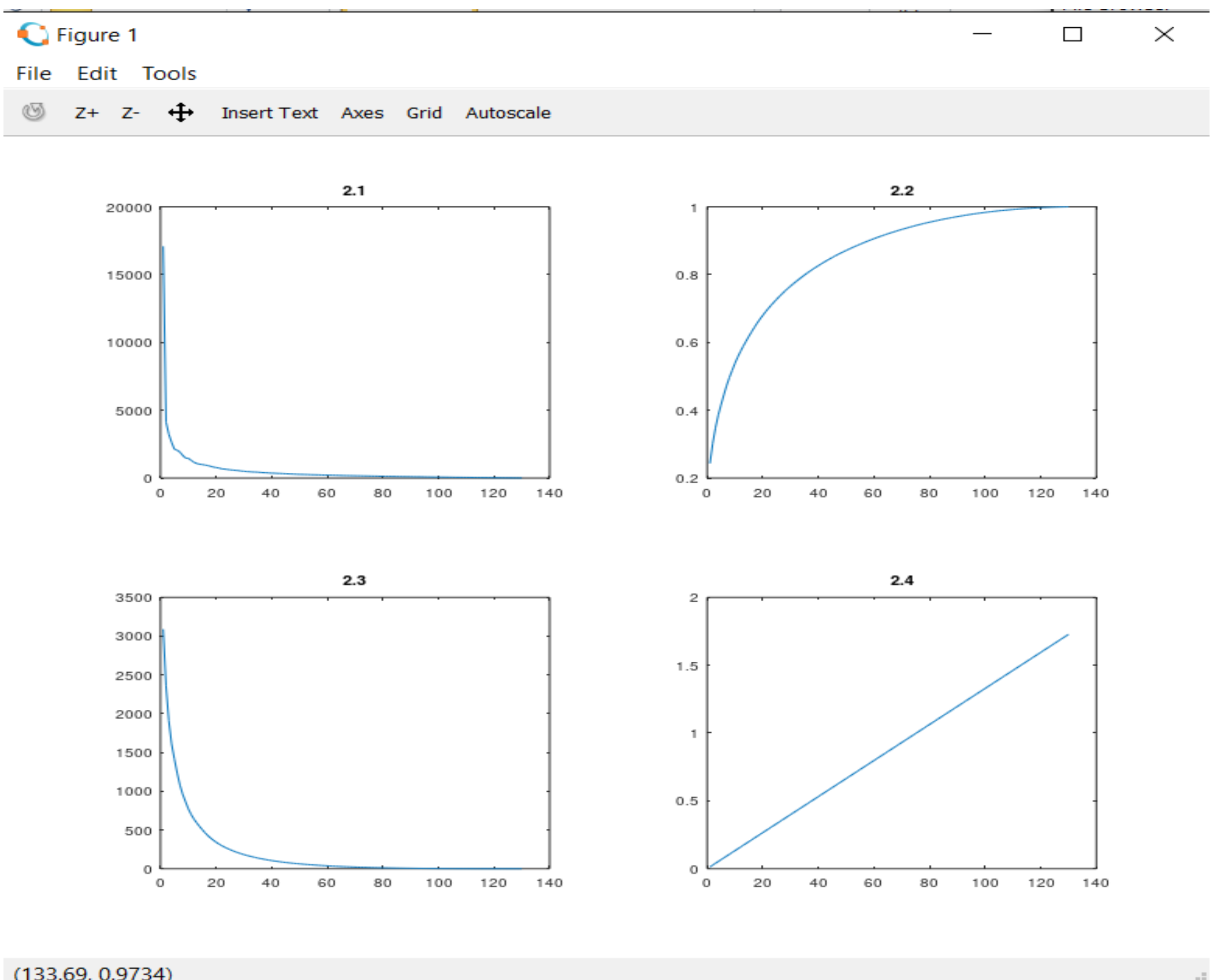
1. A k-a valoare singulara a lui A (S(k, k));
2. Raportul dintre urma matricei formata din primele k linii si k coloane ale matricei S (S(1:k, 1:k)) si urma matricei S;
3. Eroarea aproximarii lui A_k (calculata pentru orice k din 1:min(m, n)) la A, care este egala cu (suma patratelor diferentelor dintre A(i, j) si A_k(i, j)) / (m * n);
4. rata de compresie a datelor, care este determinata cu ajutorul formulei: $f(k) = \frac{(m+n+1)*k}{m*n}$.

Grafice task 2

Exemplul 1

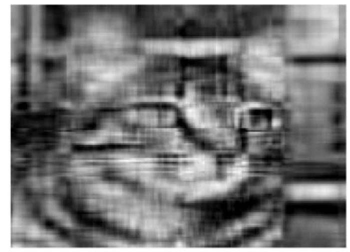
A_k = task1(‚./in/images/image4.gif’, 10)

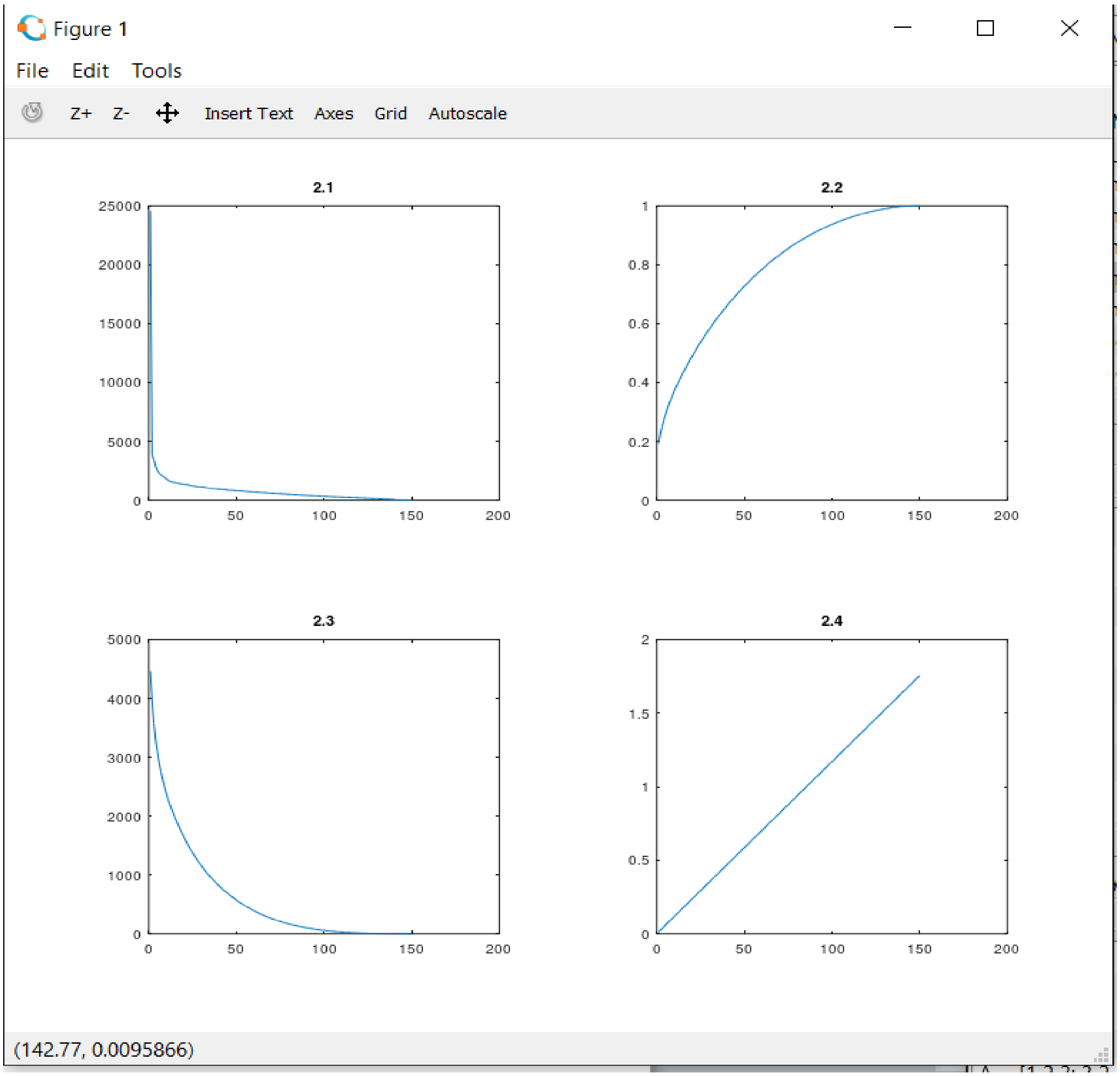




Exemplul 2

A_k = task1(‚./in/images/image3.gif’, 10)





Task 3

Avand la dispozitie o matrice A din $\boldsymbol{R}^{m*n}$, retinem intr-un vector coloana μ (notat in program cu „miu"), pe pozitia i a acestuia, media aritmetica a elementelor de pe linia i a lui A. Din fiecare linie a lui A scadem μ, iar matricea transpusa rezultata in urma scaderii mentionate o vom impatri la $\sqrt{n-1}$, rezultatul obtinut fiind memorat intr-o matrice Z.

Dupa *DVS*-ul matricei Z (Z = U * S * V^T), retinem primele k coloane ale matricei V intr-o matrice W, a carei transpusa va fi inmultita cu A la dreapta si retinuta intr-o alta matrice, notata Y.

Functia intoarce valorile singulare ale matricei Z si matricea A_k calculata astfel: A_k = W * Y + μ.

Task 4

Construim vectorul μ pe care il scadem la A, la fel ca la "Task 3". Ce difera la acesta cerinta fata de cerinta anterioara este felul in care generam matricea Z; Z este, de data asta, $\frac{A*A\prime}{n-1}$.

Matricea V va fi matricea vectorilor proprii ai lui Z si retinem primele k coloane ale matricei V intr-o matrice W, iar de aici, metoda de prelucare a rezultatelor care trebuie returnate este aceeasi ca la "Task 3".

Task 5

(*OBSERVATIE: Imaginea testata in fisierul 'task5.m' este './in/images/image1.gif'*)

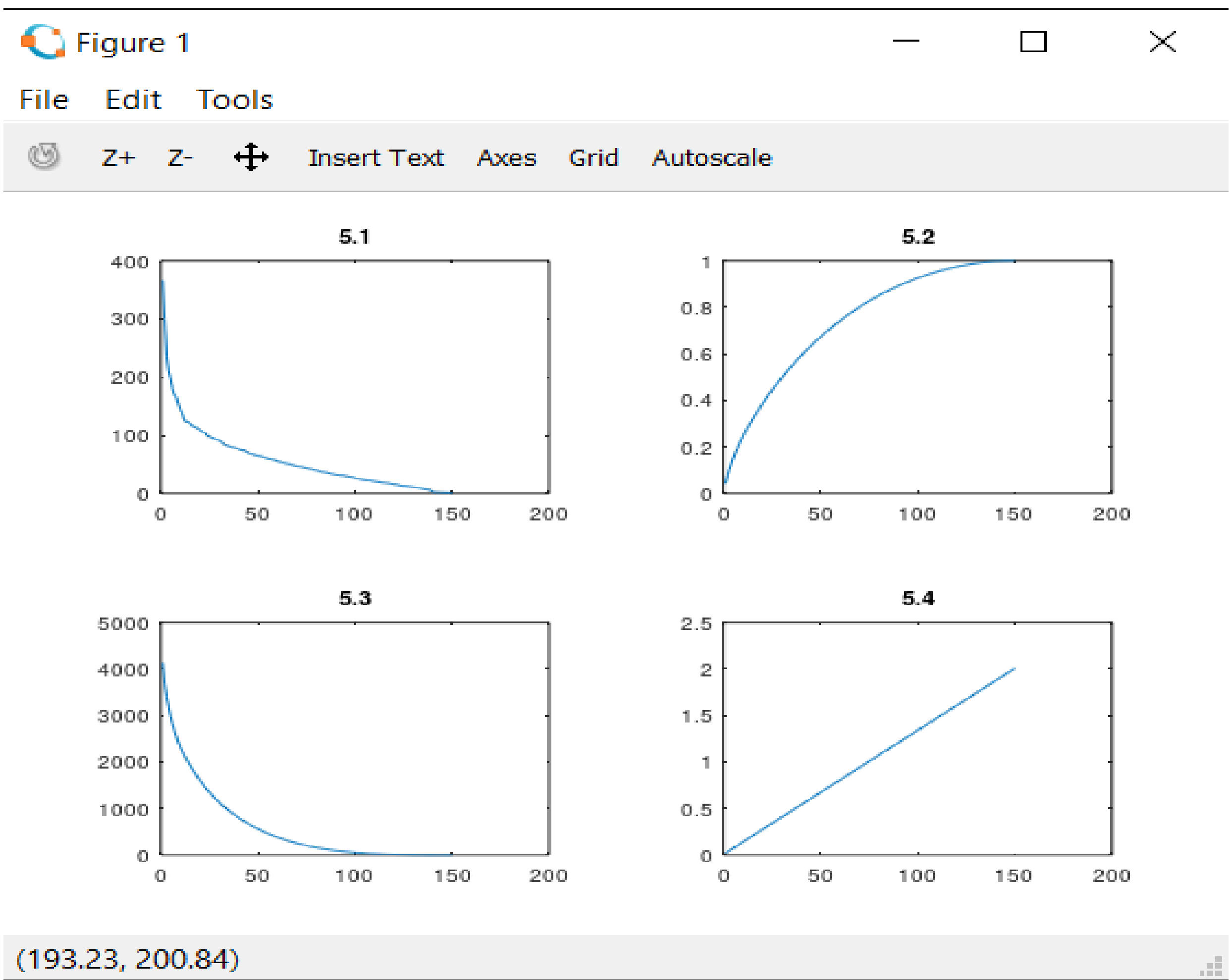
Dupa citirea matricei A corespunzatoare unei imagini citite in functie (notata cu img), alcatuim 4 grafice astfel incat pe axa Ox sa avem numere naturale de la 1 la min(m, n), iar pentru fiecare k din 1:min(m, n) sa generam matricele A_k si S returnate de functia task3(img, k) si sa avem pe axa Oy f(k) = :

1. S(k, k);
2. Raportul dintre urma matricei formata din primele k linii si k coloane ale matricei S (S(1:k, 1:k)) si urma matricei S;
3. Eroarea aproximarii lui A_k (calculata pentru orice k din 1:min(m, n)) la A, care este egala cu (suma patratelor diferentelor dintre A(i, j) si A_k(i, j)) / (m * n);
4. rata de compresie a datelor, care este determinata cu ajutorul formulei: $f(k) = \frac{2k+1}{n}$.

Grafice task 5

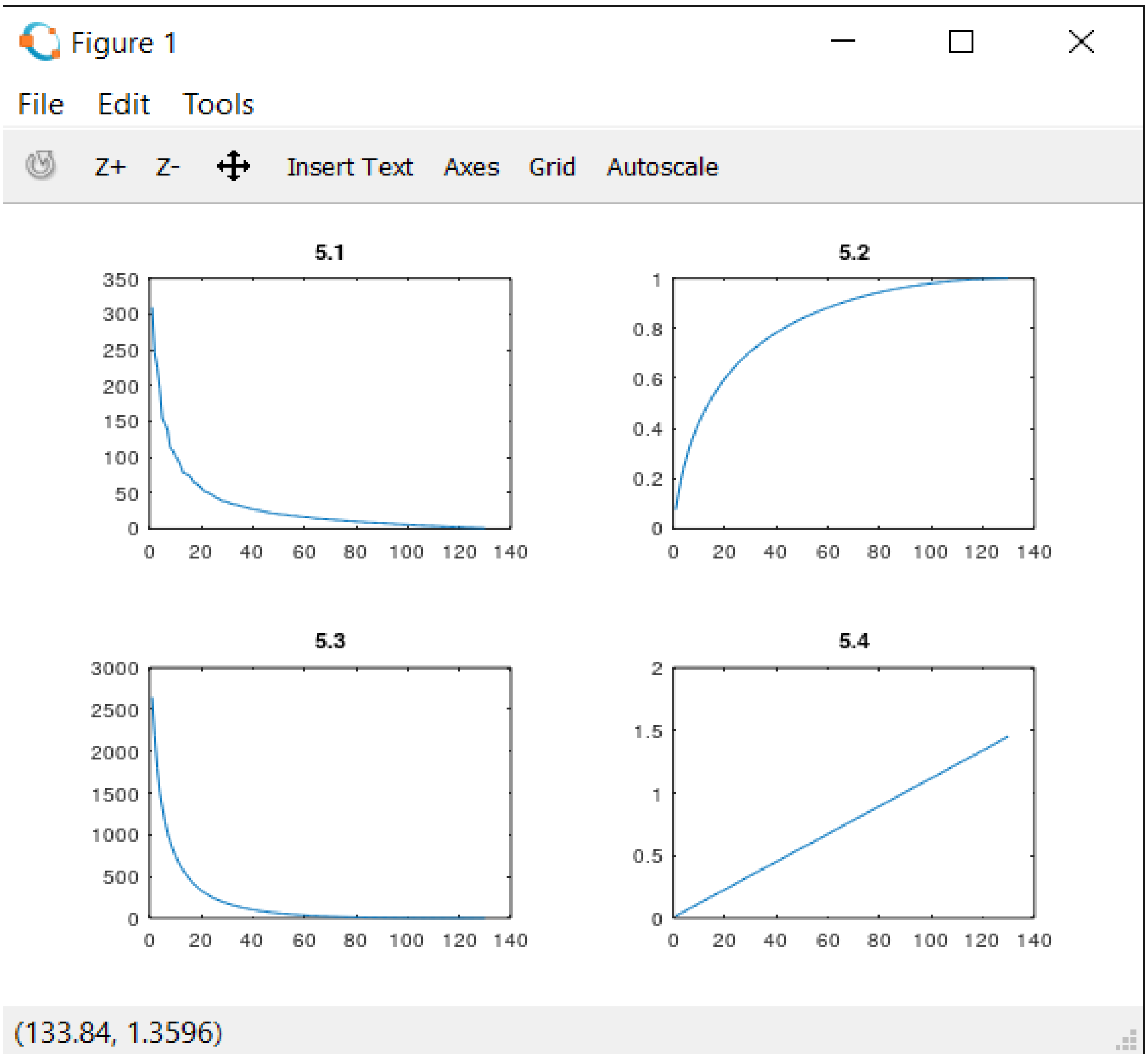
Exemplul 1

[A_k, S] = task3(‚./in/images/image4.gif', 10)



Exemplul 2

[A_k, S] = task3(,./in/images/image3.gif’, 10)



Task 6

- *Pasul 1 („./task6/eigenface_core.m")*

Memoram cele 10 matrice asociate celor 10 fete din directorul *./task6/dataset* intr-un vector de matrice img (dimensiune: 200 * 200 * 10) si vectorii coloana asociati acestor matrice intr-o matrice T (dimensiune: $200^2 * 10$). (OBSERVATIE: Imaginile sunt trecute din RGB in alb-negru inainte de atribuirea vectorului tridimensional img)

In vectorul coloana m (care va fi returnat), retinem pe pozitia i (i din 1:40000) media aritmetica a elementelor de pe linia i a matricei T, apoi matricea A, care va fi returnata de catre aceasta functie, va fi rezultatul scaderii fiecarei linii din T la vectorul m.

Avand matricea $C = A^T * A$, memoram in matricea V vectorii proprii ai lui C pentru care valorile proprii corespunzatoare sunt mai mari ca 1.

Celelalte matrice returnate sunt eigenfaces = A * V (fetele „proprii") si pr_img = $eigenfaces^T * A$ (proiectiile fiecarei fete).

- *Pasul 2 („./task6/face_recognition.m”)*

Dupa transformarea imaginii citite ca parametru intr-o matrice img, generam vectorul coloana T asociat acestei matrice din care scadem vectorul medie calculat la pasul anterior si setat ca parametru la aceasta etapa a rezolvarii temei. Proiectia acestei fete este pr_test_img = eigenfaces' * A2.

In vectorul linie d vom retine norma vectorului pr_test_img - pr_img(:,i) (i din 1 : nr.coloane(A)). Functia va returna atat valoarea minima a acestui vector, cat si indicele acestui minim (fața față de care este cea mai apropiata fața data ca parametru acestei functii).