

README

TEMA 2

-METODE NUMERICE-

Florina-Cristina Drastaru

315CA, ACS, UPB

Contents

1	Task 1	3
2	Task 2	3
3	Task 3	5
4	Task 4	6
5	Task 5	6
6	Task 6	9
6.1	Eigenface	9
6.2	Face recognition	9

1 Task 1

Am citit imaginea data ca parametru cu ajutorul funcțiilor *imread* si *double*. De asemenea, am citit k - numarul de valori singulare.

Cu ajutorul functiei *svd*, am descompus matricea citita in 3 matrici: U (m linii si m coloane), S (m linii si n coloane) si V (n linii si n coloane).

Am realizat descompunerea valorilor singulare, eliminand linii si coloane din matricile obtinute, in conformitate cu numarul de valori singulare, noile matrici obtinute fiind U_k (m linii, k coloane), S_k (k linii, k coloane), V_k (n linii, k coloane).

Noua matrice, A_k , am obtinut-o prin inmultirea matricilor U_k , S_k si trasnpusa matricei V_k .

2 Task 2

Am testat cerintele de la acest task pentru imaginile *image2.gif* si *image3.gif*.
Ordinea rezolvarii cerintelor este urmatoarea:

- Cerinta 1 - *Figure 1*, stanga-sus
- Cerinta 2 - *Figure 2*, dreapta-sus
- Cerinta 3 - *Figure 3*, stanga-jos
- Cerinta 4 - *Figure 4*, dreapta-jos

Pentru a rezolva **Cerinta 1**, am aplicat functia *svd* asupra matricei A , in urma careia am obtinut matricile U , S si V . Stiind ca matricea S contine pe diagonala valorile singulare, am aplicat functia *plot* asupra diagonalei matricei S ($diag(S)$) pentru a reprezenta grafic acele valori.

Pentru a rezolva **Cerinta 2**, am calculat informatia(vectorul *info*) data de primele k valori singulare rezultate din descompunerea valorilor singulare ale matricei A , calculand raportul dintre suma primelor k valori singulare si suma totala.

Pentru a rezolva **Cerinta 3**, am calculat eroarea aproximarii cu ajutorul functiei *mean*.

Pentru a rezolva **Cerinta 4**, am calculat rata de compresie a datelor cu ajutorul formulei oferite in cerinta.

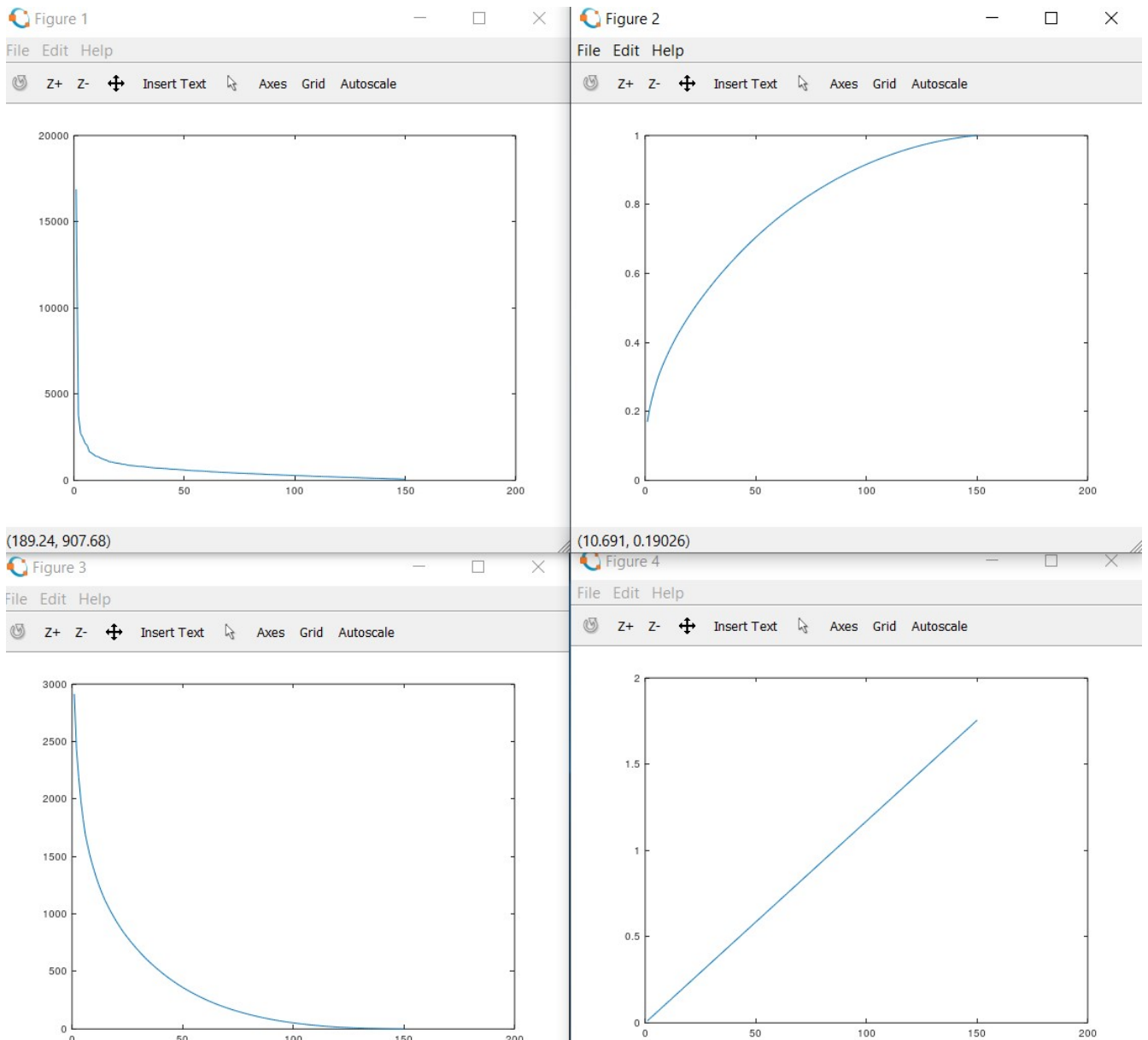


Figure 1: image2

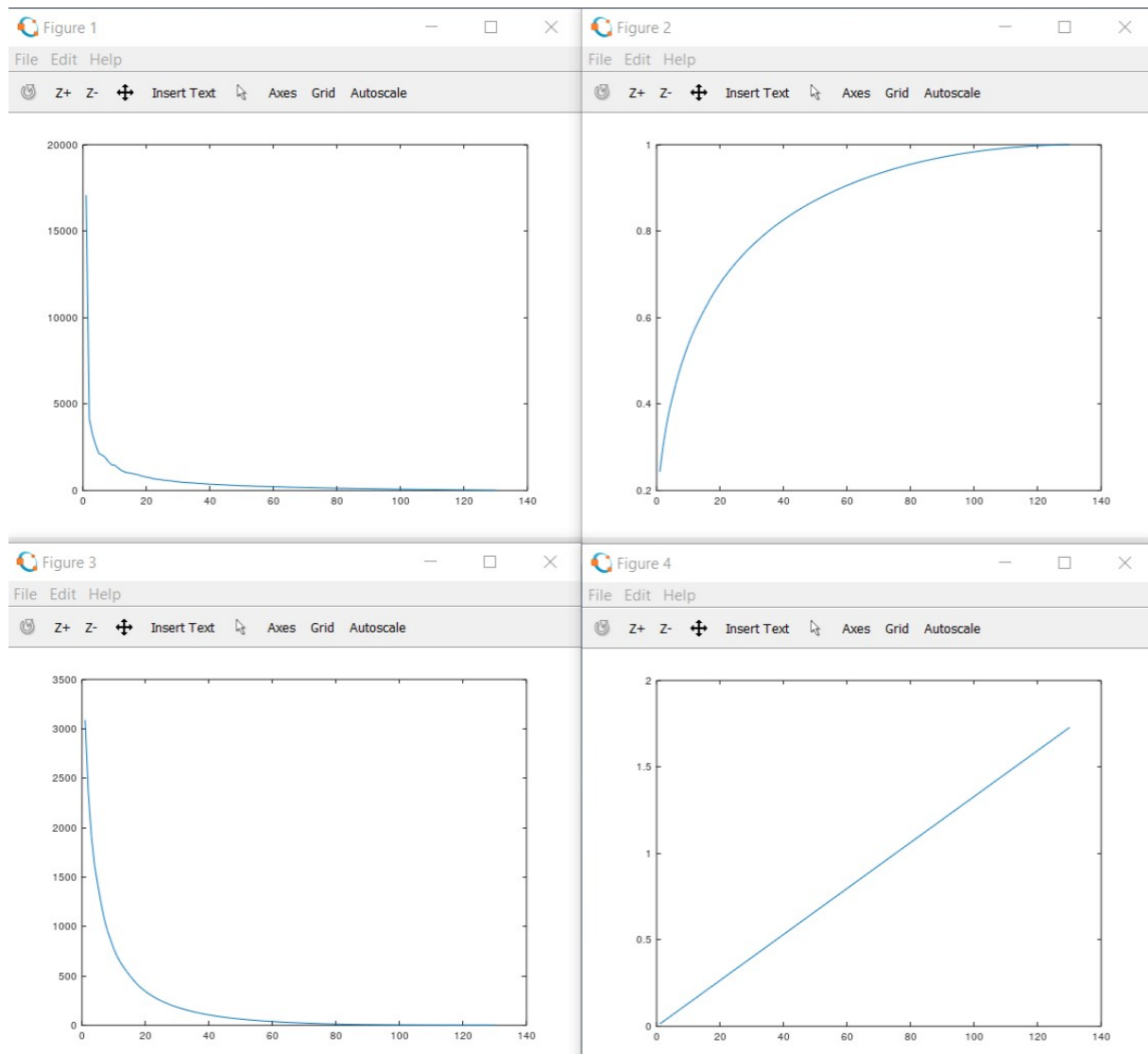


Figure 2: image3

3 Task 3

Am citit imaginea data ca parametru ca matrice.

Am calculat media pentru fiecare linie, adunand toate elementele de pe o linie si impartinand la numarul de coloane ale matricei.

Am actualizat matricea A si am construit matricea Z conform indicatiilor.

Am facut descompunerea valorilor singulare pentru matricea Z, folosind acelasi algoritm de la task1.

Am construit matricea W, pastrand doar primele k coloane din matricea V.

Am calculat proiectia lui A in spatiul componentelor principale(matricea Y).

Am aproximat matricea initiala.

4 Task 4

Aplicand acelasi algoritm de la task3, am calculat media pentru fiecare linie a matricei A, am stocat-o in vectorul niu, am actualizat matricea A si am calculat matricea Z dupa formula oferita in indicatii.

Am aflat vectorii proprii si valorile proprii ale matricei Z cu ajutorul functiei *eig*.

Am construit matricea W, pastrand doar primele k coloane din matricea V si am calculat proiectia lui Z in spatiul componentelor principale(W).

Am aproximat matricea initiala.

5 Task 5

Am testat cerintele de la acest task pentru imaginile *image2.gif* si *image3.gif*.

Ordinea rezolvarii cerintelor este urmatoarea:

- Cerinta 1 - *Figure 1*, stanga-sus
- Cerinta 2 - *Figure 4*, dreapta-jos
- Cerinta 3 - *Figure 2*, dreapta-sus
- Cerinta 4 - *Figure 3*, stanga-jos

Rezolvarile acestor cerinte au fost similare cu cele de la Task-ul 2, presupunand utilizarea formulelor oferite ca indrumare.

La Cerinta 1 am reprezentat grafic primele k valori singulare.

La Cerinta 2 am reprezentat informatia data de primele k valori singulare rezultate din descompunerea valorilor singulare ale matricei Z, dupa formula data.

La Cerinta 3 am reprezentat grafic eroarea aproximarii pentru matricea A.

La Cerinta 4 am reprezentat rata de compresie a datelor.

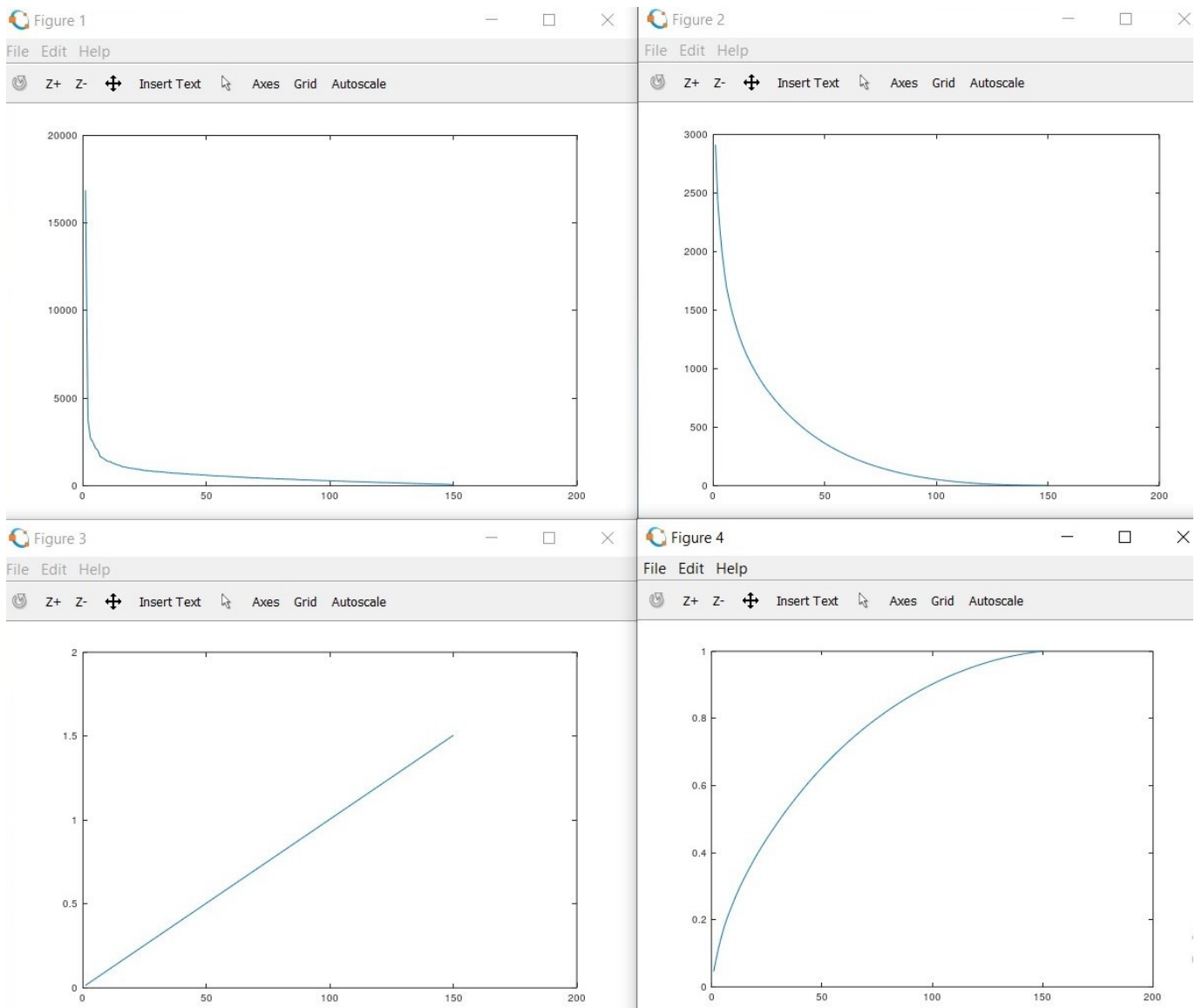


Figure 3: image3

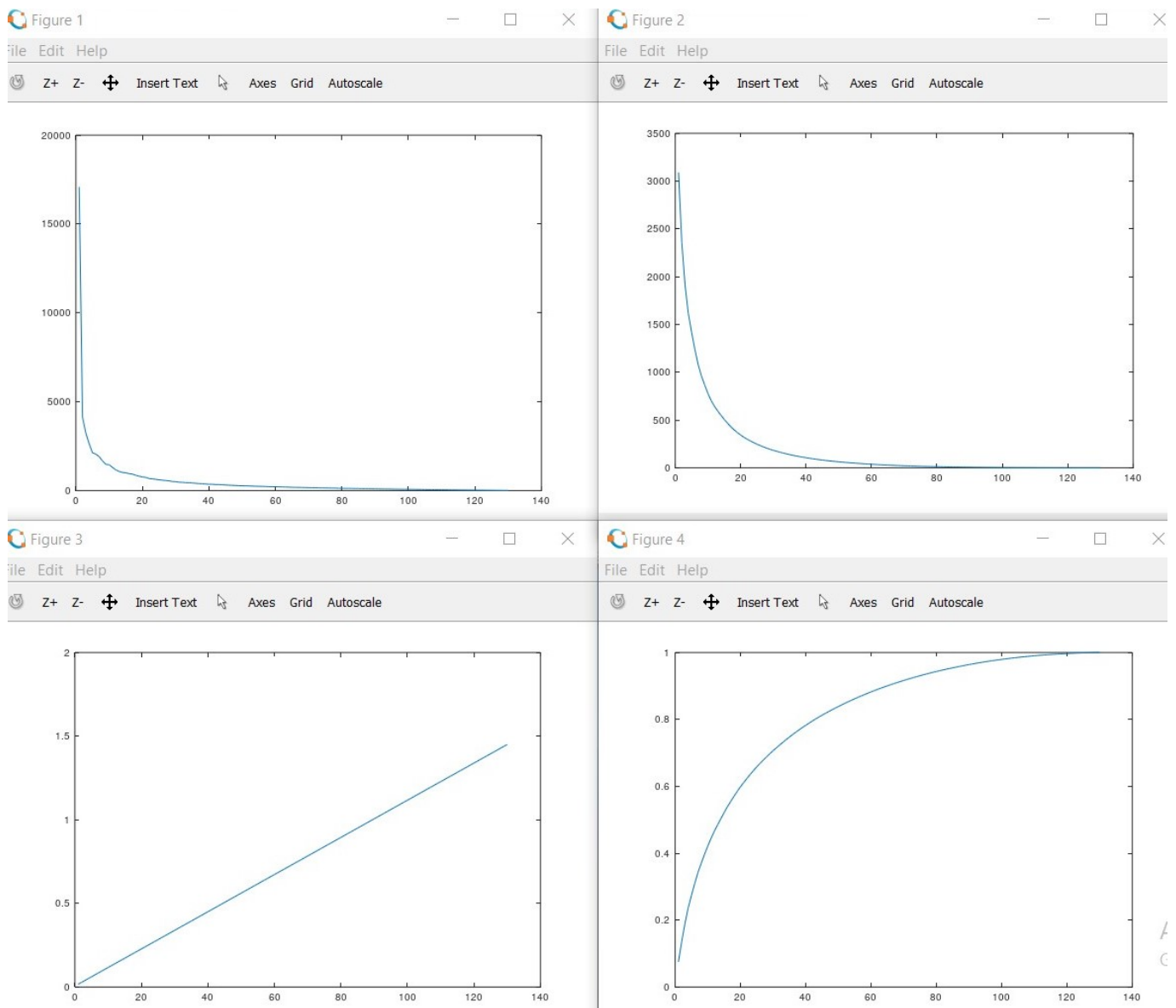


Figure 4: image3

6 Task 6

6.1 Eigenface

Am citit fiecare imagine(matricea X) din directorul *dataset* si am transformat-o intr-un vector coloana - *vc*. Apoi, fiecare vector coloana obtinut l-am concatenat la matricea T.

Am aflat media pentru fiecare linie a matricei T, am stocat-o in vectorul *m*, apoi am actualizat matricea T.

Am calculat matricea M, obtinuta din produsul dintre transpusa matricei A si matricea A si am aflat, cu ajutorul functiei *eig*, vectorii proprii(matricea V) si valorile proprii(matricea S) ale matricei M.

Am cautat in matricea S valorile proprii mai mari decat 1 si am concatenat vectorii proprii corespunzatori acestora la matricea *MEig*.

Am calculat matricea cu fetele proprii(*eigenfaces*), prin inmultirea matricei A cu matricea *MEig*.

Am calculat proiectia fiecarei matrice (*pr*) si am salvat-o in vectorul *pr_img*.

6.2 Face recognition

Am citit imaginea de test data ca parametru si am facut-o alb-negru, am stocat-o in matricea A, apoi am transformat-o in vectorul coloana *vc*.

Am actualizat matricea A, folosindu-ma de vectorul *vc* si de media trimisa ca parametru - *m*.

Am calculat proiectia imaginii de test in spatiul fetelor.

Am calculat, cu ajutorul functiei *norm*, distanta dintre proiectia imaginii de test si proiectiile obtinute in functia *eigenface_core*, trimise ca parametru in aceasta functie. Fiecare distanta - *d* am salvat-o in vectorul de distante - *dist*.

Cu ajutorul functiei *min*, aplicata vectorului *dist*, am obtinut distanta minima - *min_dist* si indexul - *output_img_index*.