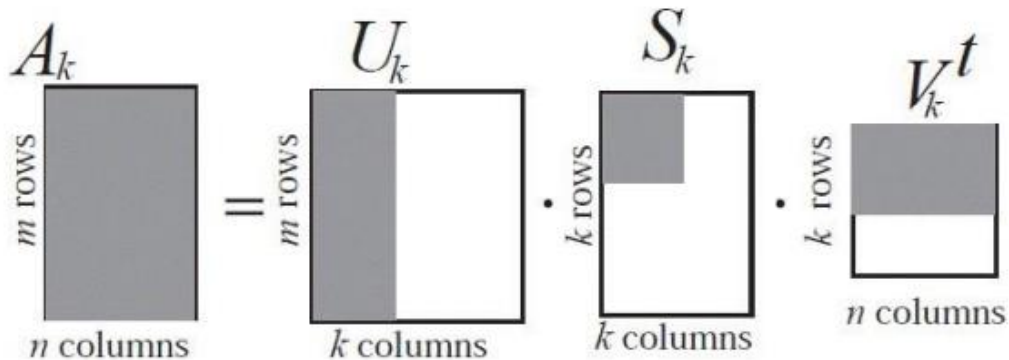


Tema 2

Task 1

După castarea la double a imaginii, am realizat descompunerea SVD a imaginii:
 $[U \ S \ V] = \text{svd}(\text{photo})$



Conform imaginii, am trunchiat rezultatele și am obținut $\text{new_X} = U * S * V'$ și am castat matricea la uint8.

Task 2

Am calculat media fiecărei linii a pozei și am scăzut vectorul coloană rezultat (notat μ) din fiecare coloană a pozei.

Am calculat Z conform formulei:

$$Z = \frac{A^T}{\sqrt{n-1}}$$

Am făcut descompunerea SVD a matricei Z și am păstrat din vectorul V doar primele k coloane, de unde rezultă matricea W .

Obținem matricea $Y = W' * A$, cu ajutorul căreia putem calcula o aproximare a matricei inițiale cu formula:

$$A_k = WY + \mu$$

Matricea rezultată este varianta compresată a imaginii date.

Task 3

Ca la taskul anterior, se calculează media liniilor și se scade vectorul coloană rezultat din fiecare coloană a matricei.

Se calculează matricea Z conform formulei:

$$Z = \frac{A * A^T}{n-1}$$

Calculăm valorile proprii ale matricei Z : $[V \ S] = \text{eig}(Z)$.

Se sortează valorile proprii descrescător și se obține matricea W , formată din primii k vectori proprii corespunzători valorilor proprii sortate.

Se calculează $Y = W' * A$, iar noul $A = WY + \mu$.

Task 4

- `prepare_data`:

Se extrag din datele din fișierul dat drept input primele `no_train_images` imagini.

- `visualise_image`

Funcție care ajută să vizualizăm inputul.

Transformă vectorul linie într-o matrice de 28×28 și dă display la imagine.

- `magic_with_pca`:

Calculează media fiecărei coloane și scade vectorul rezultat din fiecare linie a matricei.

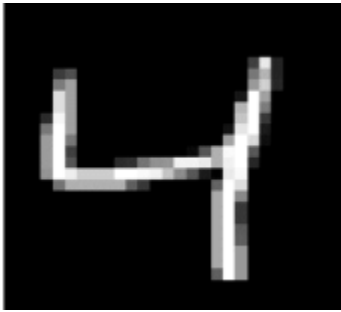
Calculează matricea de covarianță ca $X' * X / (m-1)$, matrice căreia îi calculează vectorii și valorile proprii.

Se sortează descrescător valorile proprii și se extrag în V_k primele `pcs` valori proprii corespunzătoare valorilor proprii sortate.

Se calculează $Y = X * V_k$ și se obține `train = Y * V_k'`.

- `prepare_photo`:

Transformă imaginile de test în vectori prelucrabili de restul funcțiilor.
Imaginile de test au fundalul negru, în timp ce cele de antrenament au fundal alb.



Pentru a realiza conversia, culoarea fiecărui pixel devine: 255 – culoare.
Imaginile sunt apoi transpuse și deșirate în vectori.

- KNN:

Calculăm distanța Euclidiană între vectorul test și fiecare rând al matricei Y.
Sortăm distanțele crescător și extragem în closest etichetele corespunzătoare primelor k cele mai apropiate linii.

Funcția returnează prediction, calculat ca mediana vectorului closest.

- `classifyImage`:

Apelăm `magic_with_pca`.

Scădem din fiecare linie a lui `im` media pe coloane și înmulțim `im` cu `Vk`
rezultat din `magic_with_pca`.

Calculăm prediction prin apelarea lui KNN.

Observatii

- Calitatea imaginii este influențată de numărul de componente principale alese. Cu cât numărul este mai mic, cu atât calitatea imaginii reconstruite scade.
- Creșterea numărului de componente principale duce la creșterea calității.
- Trebuie să existe un echilibru între numărul de componente alese și resursele calculatorului folosite și timpul de execuție.
- Numărul ales în temă de 23 de componente duce la o acuratețe de 93.3%.