

NICOLAI ILIE 313CB

#### TASK-1

Algoritmul convertește imaginea la double. Apoi determina matricele reduse și le calculează pe cele noi ( $U_k$ ,  $S_k$ ,  $V_k$ ). La final calculează aproximarea imaginii inițiale și o convertește la uint8 pentru a fi o imagine validă.

#### TASK-2

Algoritmul convertește imaginea la double. Se calculează media fiecărui rând iar apoi se normalizează matricea. Cu ajutorul matricei normalizate se construiește noua matrice  $Z$ . Se aplică algoritmul SVD matricei  $Z$ , se calculează matricele reduse, apoi se calculează matricele  $W$  și  $Y$ . Cu ajutorul acestora se construiește aproximația matricei inițiale care se convertește la final la uint8 pentru a fi o matrice validă.

#### TASK-3

Imaginea inițială este convertită la double și normalizată asemănător task-ului anterior. Apoi este calculată matricea de covarianță  $Z$ . Se determină valorile și vectorii proprii ai matricei  $Z$ , memorându-se valorile proprii în vectorul  $S$  (parametrul `vector` memorează valorile proprii sub formă de vector), iar vectorii proprii în matricea  $V$ . Se sortează descrescător vectorul  $S$ , cu memorarea pozițiilor elementelor din vectorul sortat (vectorul  $ind$ ). Apoi se utilizează o copie a matricei  $V$  ( $A$ ) pentru a pune vectorii în ordinea indicată de vectorul de indici înapoi în matricea  $V$ . Se calculează apoi matricea  $W$  și se schimbă baza matricei inițiale. La final se adună la loc media scăzută anterior și se trece la forma uint8.

In fiecare caz (task 1, 2 si 3) in funcție de numărul de componente principale, alese am observat ca la un număr mai mare de componente principale imaginea este mai clara iar cu cat descreștem numărul, detaliile din poza devin mai neclare.

#### TASK-4

##### 1. Funcția visualize\_image

Algoritmul pune in vectorul v linia number din matricea de antrenament, apoi transforma acest vector într-o matrice de 28x28 salvata in im. La final transforma matricea in uint8 pentru a fi o matrice valida.

##### 2. Funcția prepare\_data

Funcția încarcă datele din fișierul primit ca argument si determina datele de antrenament si etichetele. Apoi extrage informațiile cerute in variabilele corespunzătoare.

##### 3. Funcția magic\_with\_pca

Algoritmul convertește mai întâi matricea la double apoi scade din matricea inițială media fiecărei coloane. Se calculează matricea de covarianța Z si se aplica algoritmul de calculare a valorilor si vectorilor proprii asemănător task-ului 3, cu sortare asemănătoare. Apoi se calculeaza matricea Y schimbând baza matricii initiale si la final se determina matricea train.

#### 4. Funcția prepare\_photo

Funcția inversează pixelii matricei utilizând scăderea din `uint8(255)` a matricei inițiale. Apoi este transformata într-un vector utilizând funcția `reshape`.

#### 5. Funcția KNN

Funcția calculează distanța euclidiană față de vectorul de test memorând în vectorul `distance` fiecare distanță. Apoi este sortat vectorul cu reținerea indicilor și sunt memorați primii 5 indici. La final este calculată predicția determinând mediana celor 5 poziții.

#### 6. Funcția classifyImage

Algoritmul apelează funcția anterioară pe setul de date de antrenament. Scade din vectorul `image` media fiecărei coloane din `train_mat`. Apoi schimbă baza înmulțind cu `Vk`. La final este apelată funcția KNN pentru determinarea cifrei.