

README

-Task 1-

Funcția task1 primește o imagine reprezentată sub forma unei matrice și un parametru k. Scopul funcției este să aproximeze această imagine folosind descompunerea în valori singulare (SVD). Procesul constă în convertirea matricei în tipul double, aplicarea SVD pentru a obține matricile U, S și V, selecția primelor k coloane din aceste matrici pentru a obține matricile reduse U_red, S_red și V_red, și apoi calcularea unei aproximări a matricei inițiale folosind aceste matrici reduse. Imaginea rezultată este returnată sub forma unei matrice de tip uint8.

-Task 2-

Funcția task2 primește o imagine reprezentată sub forma unei matrice și un parametru pcs. Scopul funcției este să aproximeze această imagine folosind descompunerea în valori singulare (SVD) și reducerea dimensionalității. Procesul constă în convertirea matricei în tipul double, calcularea mediei pentru fiecare vector al imaginii, normalizarea imaginii prin scăderea mediei fiecărui rând, construirea matricei Z, aplicarea SVD asupra matricei Z pentru a obține matricile U, S și V, selectarea primelor pcs coloane ale matricei V pentru a obține matricea W, calcularea matricei Y prin înmulțirea transpusă a lui W cu matricea normalizată, aproximarea matricei inițiale folosind produsul dintre W și Y, și transformarea matricei rezultate în tipul uint8. Imaginea rezultată este returnată sub forma unei matrice de tip uint8.

-Task 3-

Funcția `task3` primește o imagine reprezentată sub forma unei matrice și un parametru `pcs`. Scopul funcției este să aproximeze această imagine folosind descompunerea în vectori și valorile proprii ale matricei de covarianță. Procesul constă în convertirea matricei în tipul `double`, calcularea mediei pentru fiecare rând al imaginii, normalizarea imaginii prin scăderea mediei fiecărui rând, calcularea matricei de covarianță utilizând produsul dintre matricea normalizată și transpusa sa, calcularea vectorilor și valorilor proprii ale matricei de covarianță, ordonarea descrescătoare a valorilor proprii și crearea matricei `V`, păstrarea doar a primelor `pcs` coloane din matricea `V`, calcularea matricei `Y` prin înmulțirea transpusă a lui `W` cu matricea normalizată, aproximarea matricei inițiale folosind produsul dintre `W` și `Y`, și transformarea matricei rezultate în tipul `uint8`. Imaginea rezultată este returnată sub forma unei matrice de tip `uint8`.

-Task 4-

- Funcția **`prepare_data(name, no_train_images)`** primește un nume de fișier și numărul de imagini de antrenament `no_train_images`. Scopul funcției este să pregătească setul de date pentru antrenament, extrăgând matricea de antrenament și vectorul de etichete asociate. Procesul constă în inițializarea matricei `train_mat` și vectorului `train_val`, încărcarea datelor din fișierul specificat, salvarea în matricea `train_mat` primelor `no_train_images` linii din tabelul de imagini de antrenament, și salvarea în vectorul `train_val` primelor `no_train_images` valori ale vectorului de etichete. Matricea `train_mat` și vectorul `train_val` sunt returnate ca rezultat al funcției.

- Funcția **visualise_image(train_mat, number)** primește o matrice de antrenament `train_mat` și un număr `number` care reprezintă linia din matricea de antrenament pe care se dorește vizualizarea imaginii. Scopul funcției este să extragă linia corespunzătoare numărului specificat, să o transforme într-o matrice de dimensiune 28x28 și să o vizualizeze ca o imagine. Procesul constă în inițializarea matricei finale, citirea liniei cu numărul `number` din matricea de antrenament, transformarea liniei într-o matrice 28x28 prin `reshaping` și transpunere, transformarea matricei rezultate în formatul `uint8` pentru a reprezenta o imagine validă, afișarea imaginii folosind funcția `imshow`, și returnarea matricei finale.
- Funcția **prepare_photo(im)** primește o imagine `im` și are scopul de a pregăti această imagine pentru a fi utilizată în algoritmi de procesare a datelor. Procesul implică inițializarea unui șir gol de lungime 784, inversarea pixelilor imaginii prin scăderea fiecărui pixel de la valoarea maximă (255), transpunerea imaginii și transformarea acesteia într-un vector linie prin aplicarea funcției `reshape`. Astfel, rezultatul final este un șir de pixeli reprezentând imaginea într-o formă liniară, gata pentru a fi utilizat în etapele următoare ale algoritmului.
- Funcția **magic_with_pca(train_mat, pcs)** implementează algoritmul PCA pentru prelucrarea datelor de antrenament. În primul rând, se calculează media și matricea de covarianță. Apoi, se determină vectorii și valorile proprii și se selectează componentele principale. Utilizând aceste componente, se proiectează datele într-un nou spațiu și se realizează o aproximare a datelor inițiale. Astfel, funcția realizează o transformare a datelor de antrenament folosind PCA.

- Funcția **KNN(labels, Y, test, k)** implementează algoritmul KNN (K-nearest neighbors). În primul rând, se inițializează variabilele necesare. Apoi, se calculează distanța Euclidiană între fiecare rând din matricea Y și vectorul de test primit ca argument. Se ordonează distanțele în ordine crescătoare și se selectează primele k valori care reprezintă etichetele corespunzătoare celor mai apropiate imagini. În final, se calculează predicția folosind mediana acestor k valori, care va fi rezultatul funcției. Astfel, funcția implementează algoritmul KNN pentru clasificarea datelor de test.
- Funcția **classifyImage(im,train_mat, train_val, pcs)** implementează clasificarea imaginii de intrare utilizând PCA și algoritmul KNN. În primul rând, se inițializează variabila pentru predicție. Imaginea de intrare "im" este convertită în tipul de date double. Apoi se aplică funcția "magic_with_pca" asupra setului de date de antrenament "train_mat" pentru a obține matricea "Y". Se scade media fiecărei coloane din imaginea "im" utilizând media calculată pe setul de date de antrenament. Se efectuează transformarea în baza PCA, înmulțind imaginea "im" cu matricea "Vk". În final, se calculează predicția folosind algoritmul KNN, cu $k = 5$, utilizând etichetele de antrenament "train_val" și matricea "Y". Predicția rezultată este returnată ca rezultat al funcției.