

# Clasificarea cifrelor scrise de mână

Bolohan Marian Cristian  
321AC

December 2023

# Descrierea aplicației

Această aplicație a fost realizată cu scopul de a clasifica în mod automat anumite imagini ce conțin cifre scrise de mână.

Datele sunt reprezentate de imagini cu cifre scrise de mână, având dimensiuni de 16x16 pixeli.

Pentru reducerea dimensiunii datelor am folosit *Principal Component Analysis* (PCA), care se bazează pe *Descompunerea valorilor singulare* (DVS).

# Procesarea datasetului

Datele au fost preluate dintr-un dataset ce conține 1593 de imagini cu cifre scrise de mână de 80 de persoane. Imaginile au fost ulterior încadrate într-o dimensiune de 16x16 pixeli, fiecare pixel având o valoare de 0 sau 1.

	0	1	2	3	4	5	6	7	8	9	...	256	257	258	259	260	261	262	263	264	265
0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	...	1	0	0	0	0	0	0	0	0	0
1	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	...	1	0	0	0	0	0	0	0	0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1	0	0	0	0	0	0	0	0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	...	1	0	0	0	0	0	0	0	0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	1	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1588	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	0	0	0	0	0	0	0	0	0	1
1589	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	0	0	0	0	0	0	0	0	0	1
1590	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	0	0	0	0	0	0	0	0	0	1
1591	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	0	0	0	0	0	0	0	0	0	1
1592	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	...	0	0	0	0	0	0	0	0	0	1

1593 rows x 266 columns

# Procesarea datasetului

Pe fiecare linie a matricei de date, primele 256 de valori corespund imaginii propriu-zise, iar ultimele 10 valori reprezintă un label ce clasifică imaginea. Acest label poate fi definit astfel:

$$cifra = i \quad \text{dacă } label_i = 1, i = 0 : 9$$

unde *label* este vectorul ce corespunde ultimelor 10 valori dintr-o linie a matricei de date.

# Procesarea datasetului

Matricea de date poate fi reprezentată astfel:

$$Data = [X \quad Y]$$

unde:

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_{1593}^T \end{bmatrix} \in \mathbb{R}^{1593 \times 256} \text{ și } Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_{1593}^T \end{bmatrix} \in \mathbb{R}^{1593 \times 10}$$

$X$  reprezintă matricea imaginilor și  $Y$  reprezintă matricea label-urilor. Matricea  $Y$  a fost ulterior transformată într-un vector  $y$  ce conține pozițiile de pe coloane unde valoarea este egală cu 1, după regula din slide-ul anterior.

$$y \in \mathbb{R}^{1593}$$

În aplicație, matricea  $X$  a fost denumită *data\_numbers*, iar vectorul  $y$ , *data\_labels*.

# Procesarea datasetului

Matricea  $X$  a fost ulterior sortată pe linii după valorile din  $y$  în mod crescător, pentru a avea toate cifrele în ordine crescătoare.

După afișarea numărului de imagini pentru fiecare cifră, se observă că există imagini lipsă, nu toate cifrele au același număr de imagini, de exemplu, pentru cifra 0 au fost găsite 161 de imagini, iar pentru cifra 8 au fost găsite doar 155.

Așadar, am selectat pentru fiecare cifră primele 150 de imagini, pentru a asigura o egalitate a datelor.

Astfel:

$$X \in \mathbb{R}^{1500 \times 256} \text{ și } y \in \mathbb{R}^{1500}$$

## Procesarea datasetului

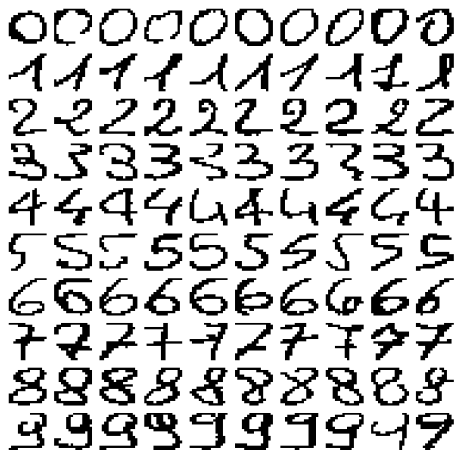


Figura: Primele 10 imagini cu fiecare cifră din baza de date

# Împărțirea datelor

În continuare, baza de date va fi împărțită pentru antrenare și testare. Pentru antrenare am alocat 80% din toată baza de date, ce reprezintă câte 120 de imagini pentru fiecare cifră. Prin urmare, matricea de lucru:

$$X_{train} \in \mathbb{R}^{1200 \times 256} \text{ și } y_{train} \in \mathbb{R}^{1200}$$

În aplicatie, datele de antrenare au fost denumite *data\_numbers\_train*, respectiv *data\_labels\_train*, iar datele de testare au fost denumite *data\_numbers\_test*, respectiv *data\_labels\_test*.



# PCA

Calculăm DVS-ul lui  $X_{train} = U\Sigma V^T$ , unde  $U \in \mathbb{R}^{1200 \times 1200}$  și  $V \in \mathbb{R}^{256 \times 256}$  sunt matrici ortogonale, iar  $\Sigma$  este o matrice diagonală ce conține valorile singulare ordonate crescător.

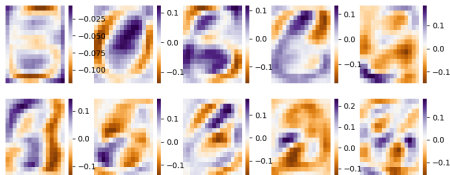
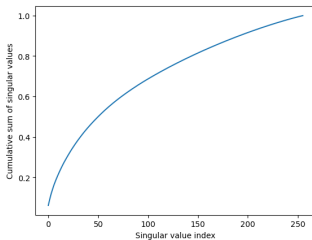


Figura: Primele 10 coloane ale matricii  $V$

# PCA

Pentru a reduce datele, calculăm suma cumulativă a valorilor singulare:



Folosind graficul sumei cumulative a valorilor singulare, alegem numărul de componente principale pentru reducerea datelor. În acest caz alegem  $r = 150$  de componente, din care recuperăm 81,3% din informație. Astfel, putem calcula matricea datelor reduse  $X_{train\_red}$ :

$$X_{train} = U\Sigma V^T \Rightarrow X_{train\_red} = X_{train} V_r = U_r \Sigma_r \in \mathbb{R}^{1200 \times 150}$$

Pentru eficiență se face calculul  $X_{train\_red} = U_r \Sigma_r$ .

În aplicație, matricea  $X_{train\_red}$  a fost numită *data\_numbers\_train\_reduced*.

# PCA

Din graficul primelor 3 componente principale, se observă o separare a cifrelor în diferite cluster, însă separația nu este foarte clară folosind un număr mic de componente principale.

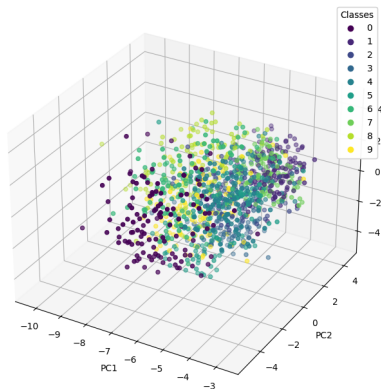


Figura: Graficul primelor 3 componente principale

## Etapa de testare

Pentru recunoașterea cifrei din imagine, se caută cea mai asemănătoare imagine din datele de antrenare, după care imaginea se clasifică conform label-ului imaginii găsite.

Imaginea din baza de testare se proiectează pe spațiul datelor reduse:

$$testImg_{red} = testImg * V_r$$

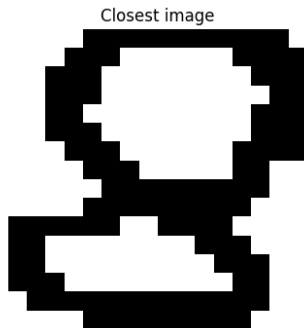
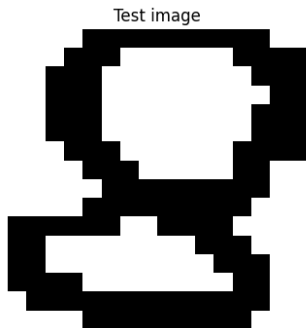
Pentru a găsi imaginea cea mai asemănătoare, ne folosim de distanța Euclidiană pentru a găsi cea mai apropiată imagine din baza de antrenare în noul spațiu.

$$d = \min_{i=1:1200} \|testImg_{red} - X_{train\_red}[i]\|$$

## Etapa de testare

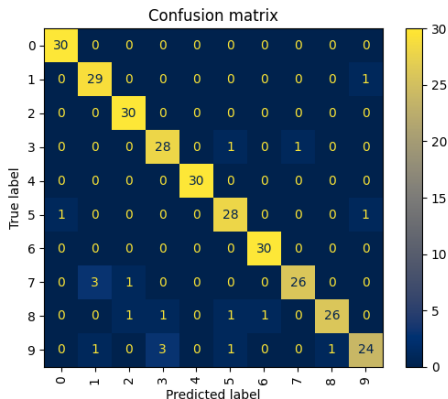
În acest exemplu, cea mai apropiată imagine a fost găsită la o distanță de 1.39 față de imaginea de testare.

Imaginea a fost clasificată după label-ul imaginii găsite, fiind un 8.



# Matricea de confuzie

Matricea de confuzie ne arată dacă datele prezise de clasificator corespund cu datele reale.



# Matricea de confuzie

Folosind matricea de confuzie, putem calcula acuratețea clasificatorului:

$$accuracy = \frac{\sum_{i=0}^9 A_{ii}}{\sum_{i=0}^9 \sum_{j=0}^9 A_{ij}} \%$$

unde  $A$  este matricea de confuzie și această formulă semnifică numărul de predicții corecte raportate la numărul total de predicții.

Pentru această aplicație, acuratețea clasificatorului este de 93.67%.

# Referințe

- 1 Curs MN
- 2 Dataset
- 3 Matrice de confuzie