

## Temă de casă: Algoritm PCA și aplicațiile sale în recunoașterea fețelor

### Aspecte teoretice

---

În literatură au fost propuse numeroase tehnici de recunoaștere/autentificare a fețelor, cu grade de complexitate, constrângeri, performanțe și arii de aplicabilitate foarte diverse. În mod special, identificarea, extragerea și ierarhizarea setului de trăsături semnificative care va constitui „semnătura” fiecărei fețe supuse analizei continuă să reprezinte un subiect de larg interes. Paleta foarte largă de opțiuni din această categorie poate fi clasificată ținând cont de o serie de criterii precum natura informațiilor extrase (distingem trăsături rezultate din geometria specifică fețelor, respectiv în urma aplicării unei proceduri de proiecție pe subspații liniare sau neliniare specifice), aria imaginii supuse procesării (vom regăsi metode, denumite *holistice*, care prelucrează întreaga imagine, respectiv soluții orientate pe zone limitate situate în jurul punctelor fiduciale), gradul de localizare spațială a trăsăturilor, natura reală (eventual binară) sau complexă a semnăturii, modularitatea arhitecturii, robustețea în raport cu numeroasele surse de variabilitate specifice acestor aplicații. De multe ori alegerea metodei de extragere a trăsăturilor semnificative este strâns legată de tipul clasificatorului ce urmează a fi utilizat în vederea furnizării deciziei. Mai mult, dimensiunea și particularitățile bazei de date disponibile impun constrângeri suplimentare în privința alegerii soluției adecvate.

Algoritm PCA a fost aplicat cu succes la recunoașterea fețelor, sub denumirea generică *Eigenfaces* și este descris pe larg în materialul pus la dispoziție. ”Materia primă” o constituie ansamblul imaginilor disponibile în baza de date, formate din matrici cu valori reale (eventual – binare). Fiecare astfel de matrice, presupusă de dimensiune (MxN), este mai întâi transformată într-un vector de aceeași lungime, prin concatenarea coloanelor corespunzătoare. Algoritmul de procesare presupune parcurgerea următorilor pași:

a) Se calculează valoarea medie a imaginilor care formează setul de antrenare (presupus a avea K fotografii):

$$\bar{I} = \frac{\sum_{j=1}^K I_j}{K}$$

și se „centrează” imaginile originale (se aduc la valoare medie nulă):

$$I_j^{centrat} = I_j - \bar{I}$$

b) Se calculează așa-numita *scatter matrix*, care reprezintă aproximarea matricii de covarianță a imaginilor din baza de date (aproximarea este cu atât mai bună cu cât avem mai multe imagini la dispoziție):

$$S = \frac{1}{K} A \cdot A^T$$

unde matricea A are pe coloane câte o fotografie centrată:

$$A = \begin{bmatrix} I_1^{centrat} & I_2^{centrat} & \dots & I_k^{centrat} \end{bmatrix}_{(M \times N) \times k}$$

Matricea S este simetrică și are dimensiuni (M\*N)x(M\*N).

c) Se calculează valorile și vectorii proprii ai matricii S (vectorii proprii ai matricii S în cazul lucrului cu imagini reprezentând fețe poartă denumirea *Eigenfaces*).

d) Se ordonează valorile proprii ale matricii S în sens descrescător. Se trasează un grafic care exprimă pierderea de informație în raport cu factorul de compresie. Astfel, dacă notăm cu  $\lambda_i$ ,  $i=1\dots(M \times N)$ , valorile proprii sortate ale matricii S, graficul anterior se referă la raportul (pe abscisă avem  $j=1\dots(M \times N)$ ):

$$\frac{\sum_{i=1}^j \lambda_i}{(M \times N)} \quad \sum_{j=1} \lambda_j$$

Graficul anterior permite estimarea numărului de valori și vectori proprii considerați semnificativi (adică aceia care păstrează cea mai mare parte din energia imaginilor originale).

e) Se proiectează imaginile (centrate) originale pe spațiul descris de vectorii proprii reprezentativi (tipic aceștia sunt în număr de 5-10% din numărul total). Proiecția constă de fapt în efectuarea produsului scalar dintre fiecare imagine originală și o matrice având drept coloane numai vectorii proprii semnificativi. Pentru fiecare imagine (centrată)  $I_j^{centrat}$  se obține proiecția pe baza relației:

$$W_j^T = I_j^{centratT} \cdot V_{PCA}$$

$$V_{PCA} = [E_1 E_2 E_3 \dots E_{N_{max}}]$$

unde  $N_{max}$  este numărul maxim de vectori proprii reținuți,  $E_j$  sunt vectorii proprii semnificativi, iar vectorii  $W_j$  au dimensiunea ( $N_{max} \times 1$ ) și pot fi priviți ca „semnăturile” asociate imaginilor originale.

f) Clasificarea imaginilor test presupune mai întâi determinarea „semnăturii” fiecărei imagini în raport cu subspațiul determinat anterior (și care depinde exclusiv de imaginile din setul de antrenare!) și găsirea acelei imagini din baza de date de antrenare a cărei semnătură este cea mai apropiată de semnătura imaginii de test. Aprecierea similitudinii dintre astfel de perechi de imagini se realizează folosind o metrică convenabil aleasă. Opțiunea uzuală este distanța Euclideană (L2), însă se pot utiliza și alte măsuri precum funcția de autocorelație sau cosinusul unghiului dintre 2 vectori.

## Cerințe

---

1. Baza de date utilizată este disponibilă în fișierul **data\_olivetti.asc**. Aceasta conține câte 10 fotografii ale unui grup de 40 de persoane. Fiecare fotografie are dimensiunea de 32x32 pixeli și este plasată pe câte o coloană a matricii de date menționate anterior (în acest scop, fiecare fotografie este transformată dintr-o matrice cu dimensiunile de 32x32 într-un vector cu dimensiunea 1024x1). Fiecare grup consecutiv de câte 10 fotografii corespunde unei anumite persoane. Afișați într-o imagine fotografiile corespunzătoare primei persoane (cele 10 fotografii vor fi plasate într-un singur grafic, pe 2 rânduri cu câte 5 poze).

*Indicație:* Modificarea dimensiunilor unei matrici (cu păstrarea numărului total de valori ale acesteia) se face apelând funcția **reshape.m**. Pentru a afișa o fotografie, transformați mai întâi valorile matricii 32x32 corespunzătoare în nivele de gri folosind funcția **mat2gray.m**, după care folosiți funcția **imshow.m**.

2. Efectuați experimente repetate pentru determinarea performanțelor de recunoaștere a fețelor utilizând algoritmul PCA. În acest scop, apălați fișierul de comenzi `test_eigenfaces.m`. Performanțele obținute sunt reflectate în valoarea parametrului `hit_rate_all`, care arată procentajul de recunoaștere corectă obținut într-un set de experimente succesive. Țineți cont de următoarele observații:
- pentru utilizarea algoritmului în acest gen de aplicații este necesar ca baza de date avută la dispoziție să fie împărțită în 2 subseturi: un set de antrenare (*Train*), care va fi utilizat pentru obținerea matricii  $V_{PCA}$  pe baza căreia se obțin apoi „semnăturile” corespunzătoare fiecărei fotografii, respectiv un set de testare (*Test*), care va permite evaluarea performanțelor algoritmului. În acest scop, cele 10 poze corespunzătoare fiecărei persoane vor fi alocate separat celor 2 subseturi de date, numărul de poze alocate setului de *Train* fiind definit de variabila `no_train` din fișierul `test_eigenfaces.m`.
  - numărul de vectori proprii semnificativi care vor intra în componența matricii  $V_{PCA}$  se alege cu ajutorul parametrului `thresh`. De exemplu, `thresh=0.05` semnifică faptul că numai 5% dintre vectorii proprii ai matricii  $S$  (definite în paragraful *Aspecte teoretice*) sunt păstrați.
  - fiecare dintre vectorii proprii semnificativi ai matricii  $S$  are dimensiunea de  $1024 \times 1$ . Pentru a putea fi interpretați, aceștia trebuie redimensionați sub forma unor matrici de dimensiune  $32 \times 32$ , apoi afișați urmând procedura descrisă la punctul 1.
  - există posibilitatea de a afișa imaginile approximate folosind un număr redus de vectori proprii (semnificativi) ai matricii  $S$ . Imaginile reconstruite sunt disponibile în variabilele denumite `data_train_reconstruct` și `data_test_reconstruct`.
- 2.1 Efectuați câte 10 experimente repetate (`no_trials=10`) alocând câte 5 fotografii ale fiecărei persoane setului de antrenare (`no_train=5`) și variind numărul de vectori proprii semnificativi păstrați. În acest scop, efectuați experimente pentru `thresh={0.05; 0.1; 0.15}`.
- 2.2 Efectuați câte 10 experimente repetate (`no_trials=10`) păstrând un număr fix de vectori proprii semnificativi (de exemplu, `thresh=0.05`) și variind numărul de poze ale fiecărei persoane alocate setului de *Train*. În acest scop alegeți `no_train={3, 5, 7}`.
- 2.3 Pentru unul dintre experimente (de exemplu, `thresh=0.05` și `no_train=5`) trasați graficul valorilor proprii sortate ale matricii  $S$  (adică graficul  $\sum_{i=1}^j \lambda_i / \sum_{j=1}^{(M \times N)} \lambda_j$ ).
- 2.4 Pentru unul dintre experimente, afișați într-un singur grafic imaginile reprezentând primii 10 vectori proprii semnificativi.