

# Recunoaștere Facială cu PCA și SVM

Aplicații ale Diagonalizării și Descompunerii  
în Valori Singulare

WAGNER Ștefan-Daniel

OLTEAN Dan-Gabriel    MATVEEV Victor-Nicolae

Facultatea de Științe Aplicate

Teoria Codării și Stocării Informației

Universitatea Națională de Știință și Tehnologie POLITEHNICA București (UNSTPB)

*Coordonatori:* Conf. dr. Alina Claudia Petrescu-Niță

Drd. Aurelia-Mihaela Bodeanu

Matematici Aplicate în Realitatea Virtuală, 2025–2026

## Rezumat

Lucrarea prezintă un sistem de recunoaștere facială bazat pe Analiza Componentelor Principale (PCA) și clasificare cu Support Vector Machines (SVM). Pornind de la metoda clasică *Eigenfaces* (Turk și Pentland, 1991), care identifică persoanele prin distanța euclidiană în spațiul componentelor principale, înlocuim clasificarea cu un SVM cu nucleu RBF. Sunt tratate în detaliu fundamentele matematice: descompunerea spectrală, SVD, formularea PCA ca problemă de valori proprii, precum și derivarea SVM prin multiplicatori Lagrange. Implementarea practică folosește datasetul *Labeled Faces in the Wild* și este însoțită de o aplicație web demonstrativă și de o prezentare cu animații realizată în Manim.

# Cuprins

<b>Glosar de termeni</b>	<b>3</b>
<b>1 Introducere</b>	<b>5</b>
1.1 Structura lucrării	5
<b>2 Fundamente Matematice</b>	<b>6</b>
2.1 Spații vectoriale și produse scalare	6
2.2 Valori și vectori proprii	6
2.3 Diagonalizare	6
2.4 Descompunerea în Valori Singulare (SVD)	8
<b>3 Analiza Componentelor Principale (PCA)</b>	<b>9</b>
3.1 Necesitatea reducerii dimensionalității	9
3.2 Matricea de covarianță	9
3.3 PCA ca problemă de valori proprii	10
3.4 Calculul practic prin SVD	11
3.5 Eigenfaces	11
3.6 Alegerea numărului de componente	11
<b>4 Support Vector Machines (SVM)</b>	<b>12</b>
4.1 Clasificare binară liniară	12
4.2 SVM cu separare strictă ( <i>hard-margin</i> )	13
4.2.1 Formularea primală	13
4.2.2 Formularea duală prin multiplicatori Lagrange	13
4.3 SVM cu penalizare ( <i>soft-margin</i> )	14
4.4 Funcții nucleu ( <i>kernel trick</i> )	14
4.5 Extensia la mai multe clase	15
<b>5 Aplicație: Recunoaștere Facială cu PCA + SVM</b>	<b>16</b>
5.1 Arhitectura sistemului	16
5.2 Datasetul LFW	16
5.3 Preprocesare	17
5.4 Configurația modelului	17
5.5 Rezultate experimentale	18
5.6 Comparatie cu metoda clasică Eigenfaces	20
5.7 Analiza rezultatelor	21
<b>6 Concluzii</b>	<b>22</b>
<b>Declarație privind utilizarea instrumentelor de IA</b>	<b>23</b>
<b>Bibliografie</b>	<b>23</b>

# Glosar de termeni

Termeni recurenți, cu trimiteri la secțiunile sau documentațiile relevante.

## Concepte

**CLAHE** - *Contrast Limited Adaptive Histogram Equalization*. Egalizare locală a histogramei cu limitarea amplificării zgomotului; Secțiunea 5.

**Computer vision** - vedere artificială; domeniu care procesează informația vizuală.

**Curse of dimensionality** - blestemul dimensionalității; în spații de dimensiune mare, distanțele euclidiene devin aproape uniforme și clasificarea bazată pe distanță își pierde eficacitatea.

**Eigenface** - vector propriu al matricei de covarianță, reformatat ca matrice de pixeli; Secțiunea 3.

**Grayscale** - tonuri de gri; reprezentarea imaginii ca matrice scalară, fiecare pixel având o intensitate în  $[0, 255]$ .

**Kernel trick** - aplicarea unui algoritm liniar într-un spațiu transformat fără calculul explicit al transformării; Secțiunea 4.

**KKT** - condițiile Karush–Kuhn–Tucker, condiții necesare de optimalitate pentru probleme cu constrângeri de inegalitate.

**LFW** - *Labeled Faces in the Wild*; set de date public cu fotografii de fețe etichetate [7].

**Overfitting** - supraantrenare; modelul învață particularități ale datelor de antrenare și nu generalizează pe date noi.

**PCA** - *Principal Component Analysis*, reducerea dimensionalității prin proiecție pe direcțiile de varianță maximă; Secțiunea 3.

**RBF** - *Radial Basis Function*; nucleu gaussian  $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$ .

**SVD** - *Singular Value Decomposition*, descompunerea în valori singulare; Teorema 2.4.

**SVM** - *Support Vector Machine*, clasificator care caută hiperplanul cu margine maximă; Secțiunea 4.

**Whitening** - normalizarea componentelor PCA prin împărțire la  $\sqrt{\lambda_i}$ , astfel încât toate să aibă varianță unitară.

## Evaluare și metrici

**Cross-validation** - validare încrucișată; setul de antrenare se împarte în  $k$  părți disjuncte, modelul se antrenează pe  $k - 1$  părți și se evaluează pe partea rămasă, repetând de  $k$  ori. Scorul raportat este media celor  $k$  valori obținute.

**TP, FN, FP, TN** - *true positives, false negatives, false positives, true negatives*; cadranele matricei de confuzie pentru o clasă fixată; Secțiunea 5.

**Precizie** - *precision*;  $P = TP/(TP + FP)$ .

**Recall** -  $R = TP/(TP + FN)$ .

**F1-score** - media armonică între precizie și recall:  $F_1 = 2PR/(P + R)$ .

**Suport** - *support*; numărul de exemple dintr-o clasă în setul de evaluare.

## Biblioteci software

**Manim** - bibliotecă pentru animații matematice; am folosit-o pentru prezentarea proiectului. <https://docs.manim.community/>

**Matplotlib** - bibliotecă pentru generarea graficelor. <https://matplotlib.org/stable/>

**NumPy** - bibliotecă pentru tablouri n-dimensionale și algebră liniară. <https://numpy.org/doc/stable/>

**OpenCV** - bibliotecă pentru procesarea imaginilor. <https://docs.opencv.org/4.x/>

**scikit-learn** - bibliotecă pentru învățare automată (PCA, SVM, validare încrucișată). <https://scikit-learn.org/stable/>

# 1 Introducere

Recunoașterea facială automată este o problemă clasică în *computer vision*, cu aplicații în autentificare, indexare de imagini și supraveghere. Formularea matematică este directă: o imagine *grayscale* cu dimensiunile  $h \times w$  este reprezentată ca un vector în  $\mathbb{R}^{hw}$ , iar identificarea unei persoane revine la a decide cărei clase îi aparține acest vector. La rezoluția folosită în lucrarea de față ( $50 \times 37$  pixeli), fiecare fotografie devine un punct în  $\mathbb{R}^{1850}$ .

Lucrul direct într-un spațiu cu atâtea dimensiuni ridică două dificultăți. În primul rând, distanța euclidiană dintre vectori devine puțin discriminativă, fenomen cunoscut drept *curse of dimensionality* [1]. În al doilea rând, antrenarea unui clasificator pe atâtea trăsături este costisitoare și expune modelul la *overfitting*.

Soluția standard, propusă de Sirovich și Kirby [14] și popularizată de Turk și Pentland prin metoda *Eigenfaces* [15], este reducerea dimensionalității prin Analiza Componentelor Principale (PCA): fiecare fotografie este proiectată într-un subspațiu de dimensiune mult mai mică, iar clasificarea se realizează în acest spațiu redus. Lucrarea de față păstrează proiecția PCA, dar înlocuiește clasificarea originală prin distanță euclidiană cu un clasificator *Support Vector Machine* (SVM) cu nucleu RBF [4]. Comparația empirică a celor două abordări este prezentată în Secțiunea 5.

Combinând cele două tehnici, lucrarea ilustrează două teme matematice fundamentale: diagonalizarea matricelor simetrice (prin matricea de covarianță a setului de date) și optimizarea convexă cu constrângeri (prin formularea duală a SVM, derivată cu multiplicatori Lagrange).

## 1.1 Structura lucrării

Secțiunea 2 prezintă fundamentele matematice necesare: valori și vectori proprii, diagonalizare ortogonală și descompunerea în valori singulare. Secțiunea 3 tratează în detaliu PCA, cu demonstrația că direcția de varianță maximă coincide cu vectorul propriu dominant al matricei de covarianță. Secțiunea 4 formulează SVM, derivând problema duală prin multiplicatori Lagrange și extinzând clasificatorul prin funcții nucleu. Secțiunea 5 descrie implementarea și rezultatele experimentale obținute pe datasetul *Labeled Faces in the Wild* [7], inclusiv comparația cu metoda clasică a distanței euclidiene în spațiul PCA.

## 2 Fundamente Matematice

### 2.1 Spații vectoriale și produse scalare

Spațiul euclidian  $\mathbb{R}^n$  este echipat cu produsul scalar standard

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^n x_i y_i \quad (2.1)$$

și norma asociată  $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ . O mulțime de vectori  $\{u_1, \dots, u_k\}$  este *ortogonală* dacă  $\langle u_i, u_j \rangle = 0$  pentru  $i \neq j$ , și *ortonormată* dacă, în plus,  $\|u_i\| = 1$  pentru toți  $i$ .

### 2.2 Valori și vectori proprii

**Definiție 2.1.** Fie  $A \in \mathbb{R}^{n \times n}$ . Un scalar  $\lambda \in \mathbb{R}$  este *valoare proprie* a lui  $A$  dacă există un vector nenul  $v \in \mathbb{R}^n$  (numit *vector propriu*) astfel încât

$$Av = \lambda v. \quad (2.2)$$

Valorile proprii se obțin ca rădăcini ale ecuației caracteristice  $\det(A - \lambda I) = 0$ , polinom de grad  $n$  în  $\lambda$ .

**Teorema 2.2** (Teorema spectrală pentru matrice simetrice reale). Fie  $A \in \mathbb{R}^{n \times n}$  o matrice simetrică ( $A = A^\top$ ). Atunci:

1. Toate valorile proprii ale lui  $A$  sunt reale.
2. Vectorii proprii corespunzători unor valori proprii distincte sunt ortogonali.
3.  $A$  admite diagonalizare ortogonală: există o matrice ortogonală  $Q$  (cu  $Q^\top Q = I$ ) și o matrice diagonală  $\Lambda$  astfel încât

$$A = Q \Lambda Q^\top. \quad (2.3)$$

*Demonstrație (punctul 1).* Fie  $\lambda \in \mathbb{C}$  o valoare proprie și  $v \in \mathbb{C}^n$  vectorul propriu asociat, cu  $Av = \lambda v$ . Aplicând conjugarea complexă și folosind faptul că  $A$  are elemente reale, rezultă  $A\bar{v} = \bar{\lambda}\bar{v}$ . Atunci

$$\bar{v}^\top Av = \bar{v}^\top (\lambda v) = \lambda \|v\|^2, \quad (2.4)$$

iar pe de altă parte, folosind  $A = A^\top$ ,

$$\bar{v}^\top Av = (A^\top \bar{v})^\top v = (A\bar{v})^\top v = (\bar{\lambda}\bar{v})^\top v = \bar{\lambda} \|v\|^2. \quad (2.5)$$

Egalitatea celor două expresii implică  $\lambda = \bar{\lambda}$ , deci  $\lambda \in \mathbb{R}$ . □

### 2.3 Diagonalizare

**Definiție 2.3.** O matrice  $A \in \mathbb{R}^{n \times n}$  este *diagonalizabilă* dacă există o matrice inversabilă  $P$  și o matrice diagonală  $D$  astfel încât  $A = PDP^{-1}$ . Coloanele lui  $P$  sunt vectorii proprii ai lui  $A$ , iar intrările diagonale ale lui  $D$  sunt valorile proprii corespunzătoare.

În cazul matricelor simetrice (Teorema 2.2),  $P$  poate fi aleasă ortogonală ( $P^{-1} = P^T$ ), ceea ce simplifică atât calculele, cât și interpretarea geometrică: transformarea  $A$  se reduce la o scalare de-a lungul unor axe ortogonale. Această formă intervine direct în PCA, unde matricea de covarianță este simetrică.

Pentru a face concretă această interpretare, se pleacă de la matricea simetrică

$$A = \begin{pmatrix} 2,5 & 0,5 \\ 0,5 & 2,5 \end{pmatrix}.$$

Polinomul caracteristic  $\det(A - \lambda I) = (\lambda - 2,5)^2 - 0,25$  are rădăcinile  $\lambda_1 = 3$  și  $\lambda_2 = 2$ , cu vectorii proprii normăți  $v_1 = \frac{1}{\sqrt{2}}(1, 1)^T$  și  $v_2 = \frac{1}{\sqrt{2}}(-1, 1)^T$ . Matricea ortogonală formată cu ei drept coloane este rotația cu  $45^\circ$ ,  $Q = [v_1 \ v_2] = R(45^\circ)$ , iar cu  $\Lambda = \text{diag}(3, 2)$  se obține descompunerea  $A = Q\Lambda Q^T$ . Acțiunea ei pe planul euclidian se citește în patru cadre, urmărind vectorii proprii  $v_1$  și  $v_2$  (Figura 2.1).

$A = Q\Lambda Q^T$  applied to the eigenvectors  $v_1, v_2$

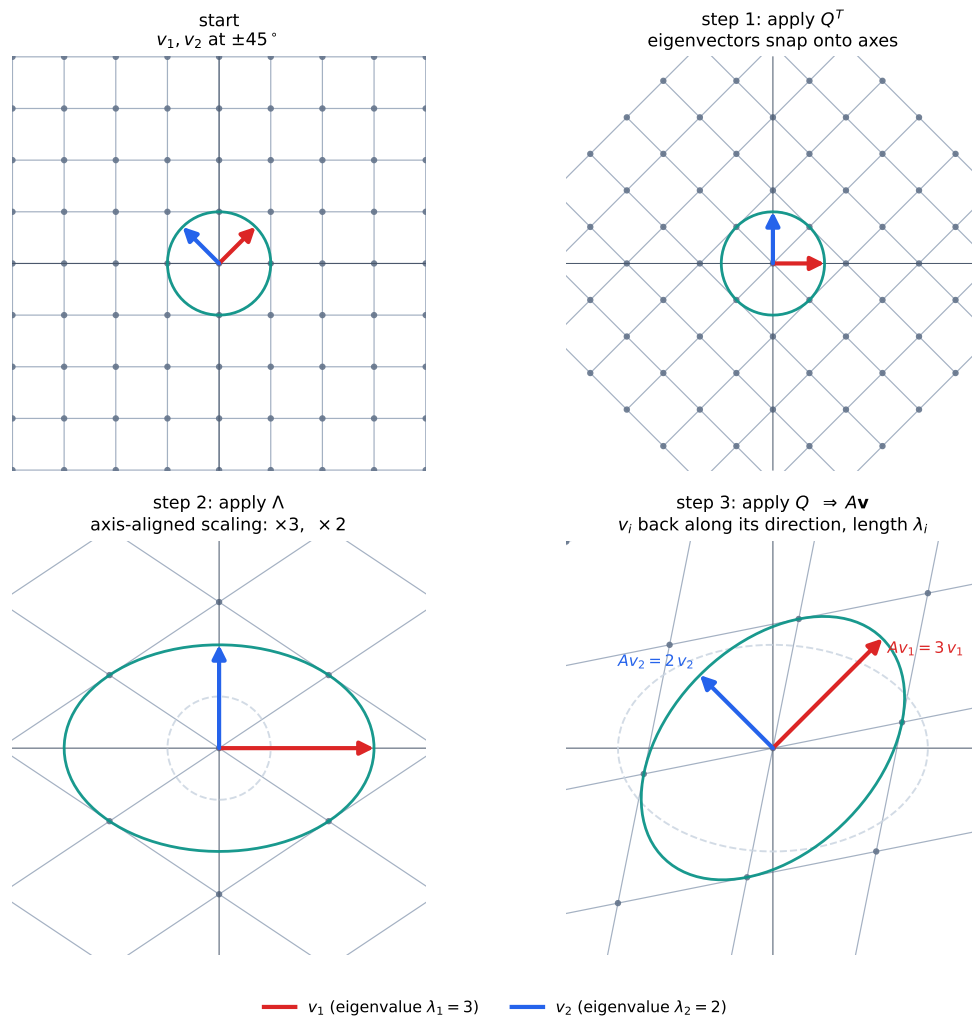


Figura 2.1: Interpretarea geometrică a diagonalizării ortogonale  $A = Q\Lambda Q^T$ : rotație în baza vectorilor proprii ( $Q^T$ ), scalare pe axe ( $\Lambda$ ), rotație inversă ( $Q$ ). Vectorii proprii revin pe direcțiile inițiale cu lungimile  $\lambda_i$ , ilustrând  $Av_i = \lambda_i v_i$ .

## 2.4 Descompunerea în Valori Singulare (SVD)

SVD generalizează ideea de diagonalizare la matrice dreptunghiulare.

**Teorema 2.4** (Descompunerea în valori singulare). *Fie  $X \in \mathbb{R}^{m \times n}$ . Atunci  $X$  admite descompunerea*

$$X = U \Sigma V^T, \quad (2.6)$$

unde:

- $U \in \mathbb{R}^{m \times m}$  este ortogonală; coloanele sale se numesc vectori singulari la stânga;
- $\Sigma \in \mathbb{R}^{m \times n}$  are valori nenegative pe diagonala principală, ordonate descrescător:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0,$$

numite valori singulare;

- $V \in \mathbb{R}^{n \times n}$  este ortogonală; coloanele sale se numesc vectori singulari la dreapta.

**Legătura cu descompunerea spectrală.** Înmulțind  $X^T X$  și  $XX^T$  se obțin

$$X^T X = V (\Sigma^T \Sigma) V^T, \quad (2.7)$$

$$XX^T = U (\Sigma \Sigma^T) U^T. \quad (2.8)$$

Coloanele lui  $V$  sunt vectorii proprii ai lui  $X^T X$ , coloanele lui  $U$  sunt vectorii proprii ai lui  $XX^T$ , iar valorile singulare și valorile proprii sunt legate prin  $\sigma_i^2 = \lambda_i(X^T X)$ .

**Aproximare de rang redus.** Teorema lui Eckart și Young [5] afirmă că, dintre toate matricele de rang cel mult  $k$ , cea mai bună aproximare a lui  $X$  în norma Frobenius este

$$X_k = \sum_{i=1}^k \sigma_i u_i v_i^T, \quad (2.9)$$

cu eroarea

$$\|X - X_k\|_F^2 = \sum_{i=k+1}^{\min(m,n)} \sigma_i^2. \quad (2.10)$$

Pe această proprietate se bazează reducerea de dimensionalitate din PCA.

### 3 Analiza Componentelor Principale (PCA)

PCA este una dintre cele mai vechi tehnici de reducere a dimensionalității, formulată independent de Pearson [10] și Hotelling [6] și sistematizată ulterior în monografia lui Jolliffe [8].

#### 3.1 Necesitatea reducerii dimensionalității

O imagine în *grayscale* de  $h \times w$  pixeli este reprezentată ca un vector  $\mathbf{x} \in \mathbb{R}^d$ , cu  $d = h \cdot w$ . Pentru imaginile din datasetul LFW folosit în această lucrare,  $d = 50 \times 37 = 1850$ .

În spații de dimensiune mare, datele devin rare: volumul crește exponențial cu dimensiunea, iar distanța dintre oricare două puncte tinde să devină uniformă [1]. În consecință, clasificarea directă pe toți cei 1850 de pixeli este atât costisitoare computațional, cât și expusă la *overfitting*.

PCA abordează această problemă prin identificarea unui subspațiu de dimensiune  $k \ll d$  care păstrează cât mai multă variabilitate a datelor. Geometric, se caută direcțiile de-a lungul cărora fețele variază cel mai mult unele față de altele.

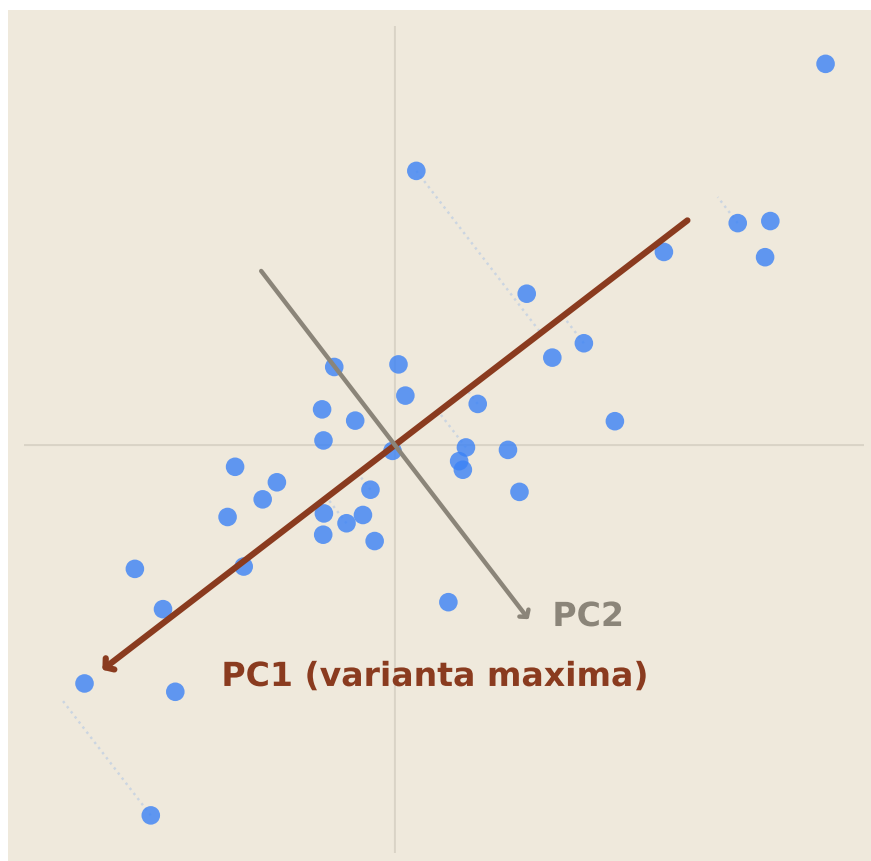


Figura 3.1: Intuiția geometrică a PCA: proiecția pe direcția de varianță maximă (PC1)

#### 3.2 Matricea de covarianță

Fie  $X \in \mathbb{R}^{n \times d}$  matricea de date, fiecare rând fiind o imagine *flattened*: matricea de pixeli rearanjată într-un vector linie de  $d$  componente, prin parcurgerea pixelilor pe rânduri.

Notăm cu  $n$  numărul de imagini și cu  $d$  numărul de pixeli. Datele se centrează prin scăderea mediei:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad \tilde{X} = X - \mathbf{1} \boldsymbol{\mu}^\top. \quad (3.1)$$

**Definiție 3.1.** Matricea de covarianță empirică a setului de date este

$$C = \frac{1}{n} \tilde{X}^\top \tilde{X} \in \mathbb{R}^{d \times d}. \quad (3.2)$$

Intrarea  $C_{ij}$  măsoară covarianța dintre intensitățile pixelilor  $i$  și  $j$  estimată pe setul de imagini. Următoarea proprietate este esențială pentru aplicarea diagonalizării.

**Propoziție 3.2.**  $C$  este simetrică și pozitiv semidefinită.

*Demonstrație.*

*Simetria.*

$$C^\top = \frac{1}{n} (\tilde{X}^\top \tilde{X})^\top = \frac{1}{n} \tilde{X}^\top \tilde{X} = C. \quad (3.3)$$

*Pozitiv semidefinită.* Pentru orice  $\mathbf{v} \in \mathbb{R}^d$ ,

$$\mathbf{v}^\top C \mathbf{v} = \frac{1}{n} \mathbf{v}^\top \tilde{X}^\top \tilde{X} \mathbf{v} = \frac{1}{n} \|\tilde{X} \mathbf{v}\|^2 \geq 0. \quad \square$$

Fiind simetrică,  $C$  admite diagonalizare ortogonală (Teorema 2.2):

$$C = Q \Lambda Q^\top, \quad (3.4)$$

cu  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_d)$  și  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$ .

### 3.3 PCA ca problemă de valori proprii

**Teorema 3.3** (Direcția de varianță maximă). Direcția unitară  $\mathbf{u}_1 \in \mathbb{R}^d$  care maximizează varianța proiectată,

$$\mathbf{u}_1 = \arg \max_{\|\mathbf{u}\|=1} \mathbf{u}^\top C \mathbf{u}, \quad (3.5)$$

coincide cu vectorul propriu al lui  $C$  asociat celei mai mari valori proprii.

*Demonstrație.* Lagrangianul asociat constrângerii  $\|\mathbf{u}\|^2 = 1$  este

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{u}^\top C \mathbf{u} - \lambda (\|\mathbf{u}\|^2 - 1). \quad (3.6)$$

Condiția de staționaritate  $\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \mathbf{0}$  conduce la

$$C \mathbf{u} = \lambda \mathbf{u}, \quad (3.7)$$

adică ecuația de valori proprii. În punctul de optim, varianța proiectată devine  $\mathbf{u}^\top C \mathbf{u} = \lambda \|\mathbf{u}\|^2 = \lambda$ , iar maximul se atinge pentru  $\lambda = \lambda_1$ , cea mai mare valoare proprie a lui  $C$ .  $\square$

*Observație 3.4.* Prin inducție, a  $k$ -a componentă principală este vectorul propriu corespunzător valorii proprii  $\lambda_k$ , sub constrângerea suplimentară de ortogonalitate față de primele  $k - 1$  componente.

Primii  $k$  vectori proprii  $\mathbf{u}_1, \dots, \mathbf{u}_k$  definesc matricea de proiecție  $U_k = [\mathbf{u}_1 \mid \dots \mid \mathbf{u}_k] \in \mathbb{R}^{d \times k}$ . Coordonatele unei imagini  $\mathbf{x}$  în spațiul redus sunt

$$\mathbf{z} = U_k^\top (\mathbf{x} - \boldsymbol{\mu}) \in \mathbb{R}^k. \quad (3.8)$$

### 3.4 Calculul practic prin SVD

Diagonalizarea directă a matricei  $C \in \mathbb{R}^{d \times d}$  este costisitoare când  $d$  este mare. Călea practică este SVD-ul matricei de date centrate  $\tilde{X} \in \mathbb{R}^{n \times d}$ :

$$\tilde{X} = U \Sigma V^\top. \quad (3.9)$$

Din relația (2.7) rezultă

$$\frac{1}{n} \tilde{X}^\top \tilde{X} = V \frac{\Sigma^2}{n} V^\top, \quad (3.10)$$

deci coloanele lui  $V$  sunt vectorii proprii ai matricei de covarianță, iar  $\lambda_i = \sigma_i^2/n$ . Pentru  $n < d$ , situație tipică în recunoașterea facială (în experimentele de față  $n = 973$  și  $d = 1850$ ), SVD-ul trunchiat este semnificativ mai eficient decât diagonalizarea directă a lui  $C$ .

### 3.5 Eigenfaces

În contextul recunoașterii faciale, componentele principale ale spațiului imaginilor poartă numele de *eigenfaces* [14, 15]. Fiecare eigenface  $\mathbf{u}_i \in \mathbb{R}^d$ , reformatat ca matrice de  $h \times w$  pixeli, are aspectul unei imagini de față cu nuanțe palide, codificând o direcție specifică de variație în populație: iluminare globală, poziția ochilor, conturul feței.

Orice imagine  $\mathbf{x}$  admite o reconstrucție aproximativă din  $k$  eigenfaces:

$$\mathbf{x} \approx \boldsymbol{\mu} + \sum_{i=1}^k \alpha_i \mathbf{u}_i, \quad \alpha_i = \mathbf{u}_i^\top (\mathbf{x} - \boldsymbol{\mu}). \quad (3.11)$$

Vectorul  $(\alpha_1, \dots, \alpha_k) \in \mathbb{R}^k$  este reprezentarea feței în spațiul redus și constituie intrarea pentru clasificatorul SVM.

### 3.6 Alegerea numărului de componente

Raportul de *varianță explicată cumulativă* al primelor  $k$  componente este

$$\text{VEC}(k) = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^d \lambda_i}. \quad (3.12)$$

În practică,  $k$  se alege astfel încât  $\text{VEC}(k) \geq 0,95$ , adică primele  $k$  componente captează cel puțin 95% din variabilitatea datelor. Graficul  $\text{VEC}(k)$  în funcție de  $k$  (*scree plot*) permite identificarea punctului dincolo de care componentele suplimentare aduc câștiguri neglijabile.

**Normalizarea variantei (*whitening*).** După proiecție, fiecare coordonată poate fi normalizată prin împărțire la  $\sqrt{\lambda_i}$ :

$$\tilde{\alpha}_i = \frac{\alpha_i}{\sqrt{\lambda_i}}. \quad (3.13)$$

Această operație, numită *whitening*, face ca toate componentele să aibă varianță unitară. Normalizarea este importantă pentru clasificatorii SVM cu nucleu RBF, sensibili la diferențele de scală dintre trăsături [13].

## 4 Support Vector Machines (SVM)

SVM este un clasificator introdus de Boser, Guyon și Vapnik [2] și extins la varianta *soft margin* de Cortes și Vapnik [4]. Cadrul teoretic complet apare în [16, Cap. 5]; pentru o expunere algoritmică și extensii prin nuclee, a se vedea [13].

### 4.1 Clasificare binară liniară

Fie un set de antrenare  $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$  cu  $\mathbf{x}_i \in \mathbb{R}^k$  și etichete  $y_i \in \{-1, +1\}$ . Se caută un hiperplan

$$\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^k : \mathbf{w}^\top \mathbf{x} + b = 0\}, \quad (4.1)$$

cu vectorul normal  $\mathbf{w} \in \mathbb{R}^k$  și termenul liber  $b \in \mathbb{R}$ , care să separe corect cele două clase. Regula de clasificare este

$$\hat{y}(\mathbf{x}) = \text{sgn}(\mathbf{w}^\top \mathbf{x} + b). \quad (4.2)$$

**Marginea de separare.** Distanța cu semn a unui punct  $\mathbf{x}_i$  față de hiperplan este  $\frac{y_i (\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}$ . *Marginea geometrică* a clasificatorului este distanța de la cel mai apropiat punct de antrenare la hiperplan:

$$\gamma = \min_{i=1, \dots, n} \frac{y_i (\mathbf{w}^\top \mathbf{x}_i + b)}{\|\mathbf{w}\|}. \quad (4.3)$$

Un clasificator cu marginea mai mare este mai robust la perturbații ale datelor; teoria învățării statistice oferă, în plus, garanții formale care leagă marginea de eroarea așteptată pe date noi [16].

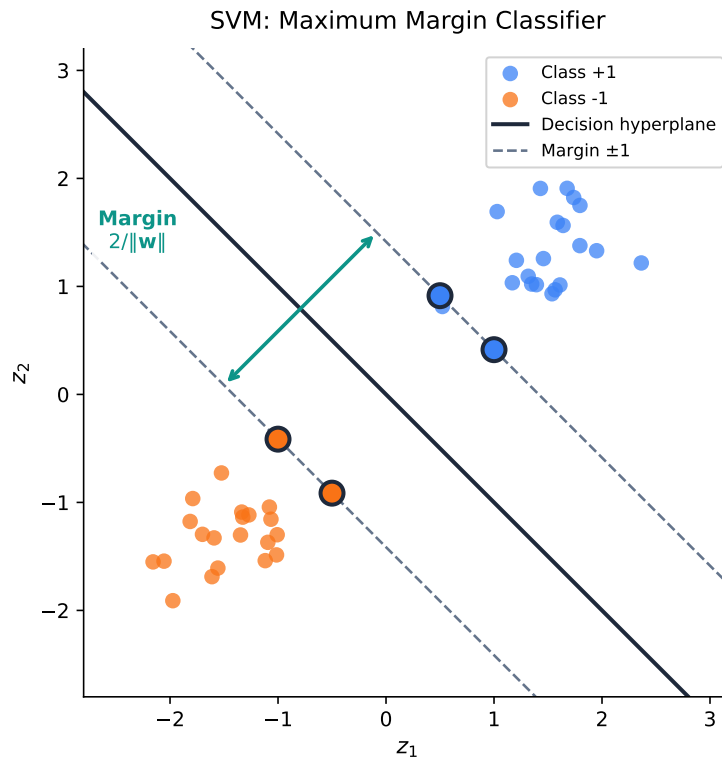


Figura 4.1: Hiperplanul optim de separare și vectorii suport (SVM hard-margin)

## 4.2 SVM cu separare strictă (*hard-margin*)

### 4.2.1 Formularea primală

Se presupune că datele sunt liniar separabile. Prin scalarea adecvată a lui  $\mathbf{w}$  și  $b$ , punctele cele mai apropiate de hiperplan satisfac  $y_i (\mathbf{w}^\top \mathbf{x}_i + b) = 1$ . În această convenție, marginea geometrică devine  $\frac{2}{\|\mathbf{w}\|}$ , iar condiția de clasificare corectă se scrie

$$y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, n. \quad (4.4)$$

Maximizarea marginii este echivalentă cu minimizarea lui  $\frac{1}{2} \|\mathbf{w}\|^2$ , conducând la problema

$$\boxed{\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{astfel încât} \quad y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, n.} \quad (4.5)$$

Forma minimizată este o funcție pătratică, iar constrângerile sunt liniare; problema este, prin urmare, o problemă de optimizare convexă cu soluție unică.

### 4.2.2 Formularea duală prin multiplicatori Lagrange

Atașând fiecărei constrângeri (4.4) un multiplicator Lagrange  $\alpha_i \geq 0$ , se obține lagrangianul

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \left[ y_i (\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right]. \quad (4.6)$$

**Condițiile KKT.** Condițiile de optimalitate Karush–Kuhn–Tucker [3] conduc la

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0} \quad \Longrightarrow \quad \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \quad (4.7)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{i=1}^n \alpha_i y_i = 0, \quad (4.8)$$

$$\alpha_i \left[ y_i (\mathbf{w}^\top \mathbf{x}_i + b) - 1 \right] = 0, \quad \forall i \quad (\text{complementaritate}). \quad (4.9)$$

**Derivarea problemei duale.** Substituind relația (4.7) în lagrangian, termenul cu  $b$  dispare datorită constrângerii (4.8), iar ce rămâne se simplifică la

$$\mathcal{L} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \quad (4.10)$$

Problema duală este

$$\boxed{\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad \text{cu} \quad \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i y_i = 0.} \quad (4.11)$$

**Vectori suport.** Din condiția de complementaritate (4.9),  $\alpha_i > 0$  dacă și numai dacă  $y_i (\mathbf{w}^\top \mathbf{x}_i + b) = 1$ , adică punctul  $\mathbf{x}_i$  se află chiar pe una din cele două hiperplane de suport. Aceste puncte se numesc *vectori suport* și sunt singurele care contribuie la soluție:

$$\mathbf{w} = \sum_{\substack{i=1 \\ \alpha_i > 0}}^n \alpha_i y_i \mathbf{x}_i. \quad (4.12)$$

*Observatie 4.1.* Problema duală (4.11) depinde de datele de antrenare exclusiv prin produsele scalare  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ . Această observație, datorată lui Boser, Guyon și Vapnik [2], este punctul de plecare pentru extensia cu funcții nucleu.

### 4.3 SVM cu penalizare (*soft-margin*)

În practică, datele rareori sunt perfect separabile liniar. Cortes și Vapnik [4] relaxează constrângerile prin introducerea variabilelor de abatere (*slack variables*)  $\xi_i \geq 0$ , care permit anumitor puncte să nu respecte marginea:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad \text{cu} \quad y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \quad (4.13)$$

Parametrul de regularizare  $C > 0$  controlează compromisul între două obiective concurente:

- **$C$  mic:** marginea este largă, dar se tolerează mai multe clasificări greșite (regularizare puternică, risc de subantrenare).
- **$C$  mare:** marginea se îngustează pentru a reduce erorile pe datele de antrenare (risc de overfitting).

În formularea duală, singura modificare față de cazul cu separare strictă este apariția unei constrângeri superioare pe multiplicatori:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad \text{cu} \quad 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^n \alpha_i y_i = 0. \quad (4.14)$$

### 4.4 Funcții nucleu (*kernel trick*)

Formularea duală (4.11) depinde de date doar prin produse scalare  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ . Această observație permite înlocuirea produsului scalar cu o *funcție nucleu*  $K(\mathbf{x}_i, \mathbf{x}_j)$  care calculează implicit produsul scalar într-un spațiu de trăsături de dimensiune mai mare [2]:

$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle, \quad (4.15)$$

unde  $\phi : \mathbb{R}^k \rightarrow \mathcal{F}$  este o transformare (posibil într-un spațiu de dimensiune infinită). Calculul explicit al lui  $\phi$  nu este necesar; este suficientă evaluarea lui  $K$ .

**Nucleul RBF (Radial Basis Function).** Folosit în această lucrare:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2), \quad \gamma > 0. \quad (4.16)$$

Parametrul  $\gamma$  controlează „raza de influență” a fiecărui punct de antrenare. Nucleul RBF corespunde unui spațiu de trăsături de dimensiune infinită și poate genera granițe de decizie complexe, în funcție de valorile  $C$  și  $\gamma$  [13].

Kernel Trick —  $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$

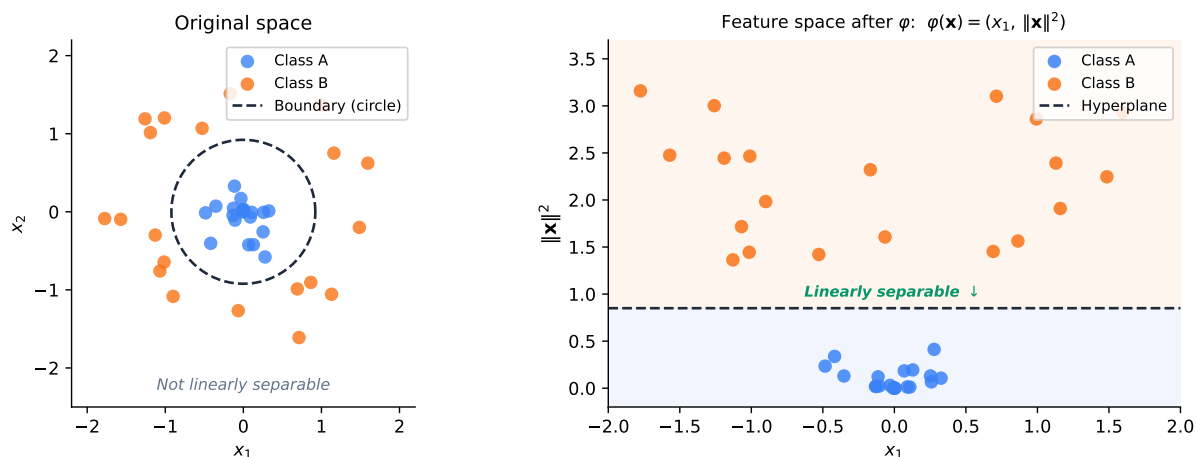


Figura 4.2: Efectul funcției nucleu: separare nelineară prin transformarea implicită  $\varphi$

**Teorema lui Mercer** [9] garantează că o funcție simetrică  $K$  corespunde unui produs scalar valid într-un spațiu Hilbert dacă și numai dacă matricea Gram  $[K(\mathbf{x}_i, \mathbf{x}_j)]_{ij}$  este pozitiv semidefinită pentru orice alegere finită de puncte.

## 4.5 Extensia la mai multe clase

SVM este, prin construcție, un clasificator binar. Pentru  $m > 2$  clase se folosește strategia *One-vs-One*: se antrenează câte un clasificator binar pentru fiecare pereche de clase (în total  $\binom{m}{2}$  clasificatori), iar predicția finală rezultă din votul majoritar.

Pentru obținerea unor probabilități de clasificare, scorul SVM se calibrează prin metoda lui Platt [12]: o regresie logistică suplimentară transformă scorurile brute în probabilități  $P(y = c | \mathbf{x})$ .

## 5 Aplicație: Recunoaștere Facială cu PCA + SVM

### 5.1 Arhitectura sistemului

Sistemul implementat parcurge următorii pași, de la imaginea brută la identificarea persoanei:



Figura 5.1: Arhitectura sistemului de recunoaștere facială

1. **Detecție și decupare:** fața este localizată în imagine printr-un clasificator Haar cascade [17].
2. **Preprocesare:** imaginea este convertită în *grayscale*, redimensionată la  $50 \times 37$  pixeli, egalizată cu CLAHE [18] și normalizată în intervalul  $[0, 1]$ .
3. **Reducerea dimensionalității:** vectorul de 1850 de pixeli este proiectat în spațiul PCA cu 150 de componente, cu *whitening*.
4. **Clasificare:** coordonatele PCA sunt clasificate de un SVM cu nucleu RBF, care returnează identitatea prezisă și probabilitățile asociate fiecărei clase.

### 5.2 Datasetul LFW

*Labeled Faces in the Wild* [7] este un set de date public ce conține fotografii ale unor persoane publice, realizate în condiții necontrolate (unghiuri, iluminări și expresii faciale diferite). Imaginile originale au rezoluția  $250 \times 250$  pixeli, color, în format JPEG. Am folosit varianta *funneled*, aliniată automat, și am aplicat un factor de redimensionare 0,4 prin `sklearn.datasets.fetch_lfw_people`, obținând imagini grayscale la  $50 \times 37$  pixeli. Am păstrat varianta filtrată cu minimum 70 de imagini per persoană, rezultând 1217 imagini distribuite între 6 persoane, și am împărțit datele stratificat în 973 de imagini pentru antrenare și 244 pentru test.

Persoană	Număr de imagini
George W. Bush	530
Colin Powell	236
Tony Blair	144
Donald Rumsfeld	121
Gerhard Schroeder	109
Ariel Sharon	77
<b>Total</b>	<b>1217</b>

Tabelul 5.1: Distribuția claselor în datasetul LFW filtrat

Se observă un dezechilibru pronunțat: George W. Bush are de aproape 7 ori mai multe imagini decât Ariel Sharon. Pentru a compensa, SVM este configurat cu ponderi de clasă proporționale cu inversul frecvenței (`class_weight='balanced'` în `scikit-learn`).

### 5.3 Preprocesare

1. **Grayscale.** Imaginile RGB de  $250 \times 250$  sunt convertite în tonuri de gri, reducând tensorul  $250 \times 250 \times 3$  la o matrice scalară  $250 \times 250$ .
2. **Redimensionare.** Factor 0,4 aplicat pe ambele axe, rezultând imagini de  $50 \times 37$  pixeli (`sklearn` ajustează ușor proporțiile pentru a se potrivi cu fețele aliniate).
3. **CLAHE [18].** Egalizarea adaptivă locală a histogramei reduce variațiile de iluminare fără a amplifica zgomotul. Implementarea folosește OpenCV cu `clipLimit=2.0` și `tileGridSize=(4,4)`: imaginea este împărțită în  $4 \times 4 = 16$  regiuni; histograma fiecărei regiuni este egalizată local, iar `clipLimit` limitează panta funcției de transfer pentru a evita amplificarea zgomotului în regiunile cu contrast scăzut.
4. **Normalizare.** Valorile pixelilor sunt scalate din  $[0, 255]$  în  $[0, 1]$ .
5. **Vectorizare.** Matricea  $50 \times 37$  este transformată într-un vector de  $d = 1850$  componente prin parcurgerea pixelilor pe rânduri (*flattened*; a se vedea Secțiunea 3).

### 5.4 Configurația modelului

Implementarea folosește biblioteca `scikit-learn` [11].

**PCA:**  $k = 150$  componente principale, cu *whitening* activat.

**SVM:** nucleu RBF, ponderi de clasă echilibrate, probabilități activate prin metoda Platt. Hiperparametrii  $C$  și  $\gamma$  au fost optimizați prin căutare pe grilă cu validare încrucișată pe 5 părți, explorând  $C \in \{10^2, 10^3, 5 \cdot 10^3\}$  și  $\gamma \in \{10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}\}$ . Valorile optime:  $C = 100$ ,  $\gamma = 0,005$ , cu un scor de validare încrucișată de 85,6%.

**Pipeline:** PCA și SVM sunt înlănțuite într-un obiect `Pipeline` din `scikit-learn`, ceea ce garantează că aceleași transformări sunt aplicate atât la antrenare, cât și la inferență.

**Nota privind validarea încrucișată.** *Cross-validation* pe 5 părți (5-fold CV) împarte setul de antrenare în 5 părți disjuncte de dimensiune aproximativ egală. Pentru fiecare combinație de hiperparametri, modelul este antrenat de 5 ori, folosind 4 părți pentru antrenare și 1 pentru evaluare; scorul raportat este media celor 5 valori de acuratețe. Procedura oferă o estimare mai stabilă a performanței decât o singură împărțire, reducând varianța datorată alegerii setului de validare.

## 5.5 Rezultate experimentale

	Precizie	Recall	F1	Suport
Ariel Sharon	1,00	0,62	0,77	16
Colin Powell	0,81	0,91	0,86	47
Donald Rumsfeld	0,90	0,79	0,84	24
George W. Bush	0,88	0,97	0,92	106
Gerhard Schroeder	0,94	0,68	0,79	22
Tony Blair	0,85	0,79	0,82	29
<b>Acuratețe globală</b>	<b>87,3%</b>			244
Media macro	0,90	0,80	0,83	244
Media ponderată	0,88	0,87	0,87	244

Tabelul 5.2: Rezultatele clasificării pe setul de test ( $C = 100$ ,  $\gamma = 0,005$ ,  $k = 150$  componente PCA)

**Matricea de confuzie.** Pentru o problemă cu  $m$  clase, *matricea de confuzie*  $M \in \mathbb{N}^{m \times m}$  contorizează predicțiile pe setul de test:  $M_{ij}$  este numărul de exemple din clasa adevărată  $i$  care au fost prezise ca aparținând clasei  $j$ . Diagonala ( $i = j$ ) conține clasificările corecte, iar elementele extra-diagonale sunt erorile.

În cazul binar (sau pentru o clasă fixată  $c$  analizată *one-vs-rest*), notațiile standard sunt cele din Figura 5.2a, care corespund celor patru cadrane ale matricei de confuzie schematizate în Figura 5.2b.

Notăție	Semnificație	Prezis	
		$c$	$\neg c$
TP	exemple din $c$ prezise ca $c$	<b>TP</b> <i>true positives</i>	<b>FN</b> <i>false negatives</i>
FN	exemple din $c$ prezise ca $\neg c$	<b>FP</b> <i>false positives</i>	<b>TN</b> <i>true negatives</i>
FP	exemple din $\neg c$ prezise ca $c$		
TN	exemple din $\neg c$ prezise ca $\neg c$		

(a) Definiții.

(b) Cadranele matricei de confuzie.

Figura 5.2: Notațiile binare TP, FN, FP, TN și poziționarea lor în matricea de confuzie pentru clasa  $c$ .

Metricile per clasă din Tabelul 5.2 se exprimă în acești termeni:

$$\text{Precizie} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F_1 = \frac{2 \cdot \text{Precizie} \cdot \text{Recall}}{\text{Precizie} + \text{Recall}}.$$

Pentru clasa  $c$ ,  $TP = M_{cc}$ ,  $FN = \sum_{j=1, j \neq c}^m M_{cj}$  (suma pe linia  $c$ , fără diagonală) și  $FP = \sum_{i=1, i \neq c}^m M_{ic}$  (suma pe coloana  $c$ , fără diagonală).

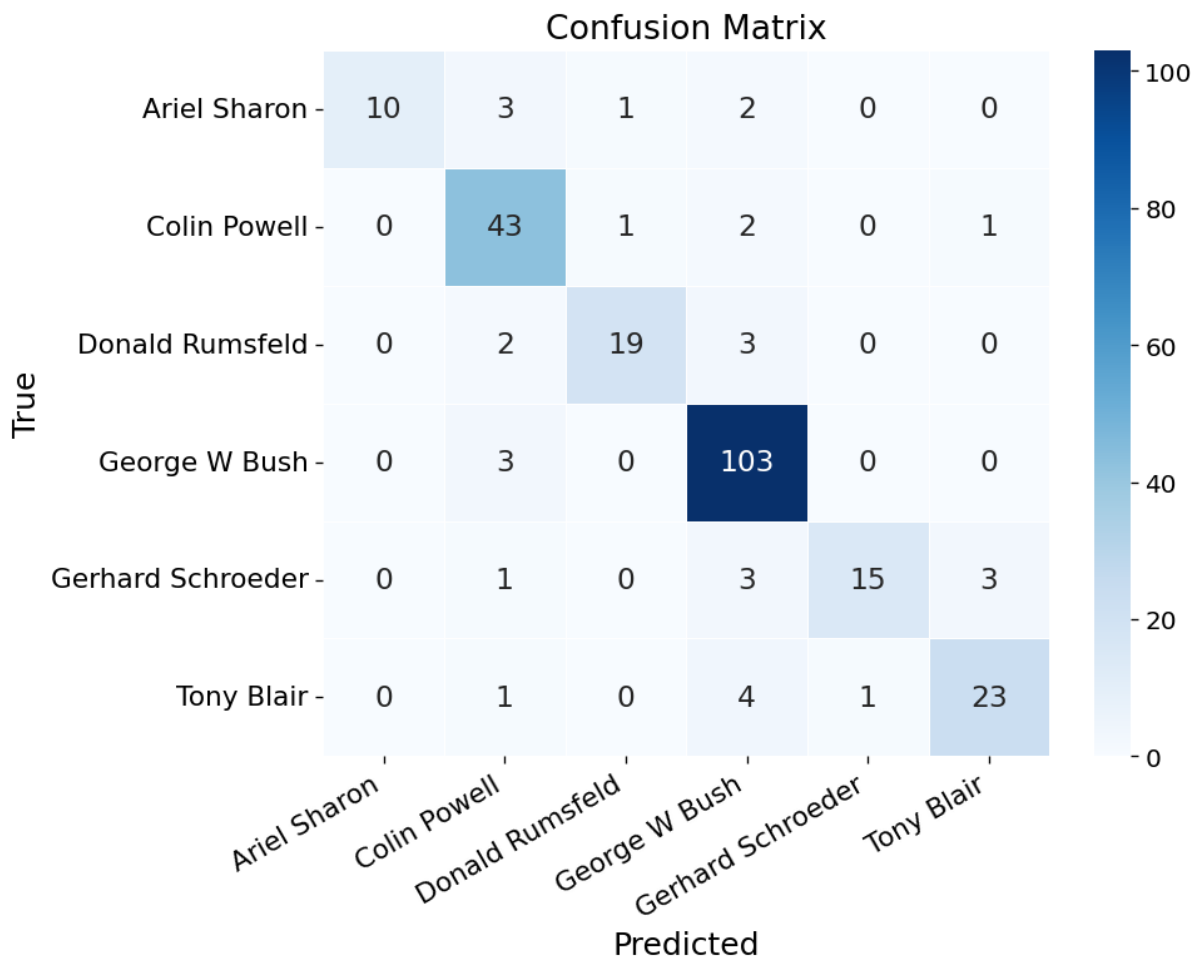


Figura 5.3: Matricea de confuzie pe setul de test. Liniile reprezintă clasa adevărată, coloanele clasa prezisă; valorile diagonale sunt clasificările corecte.

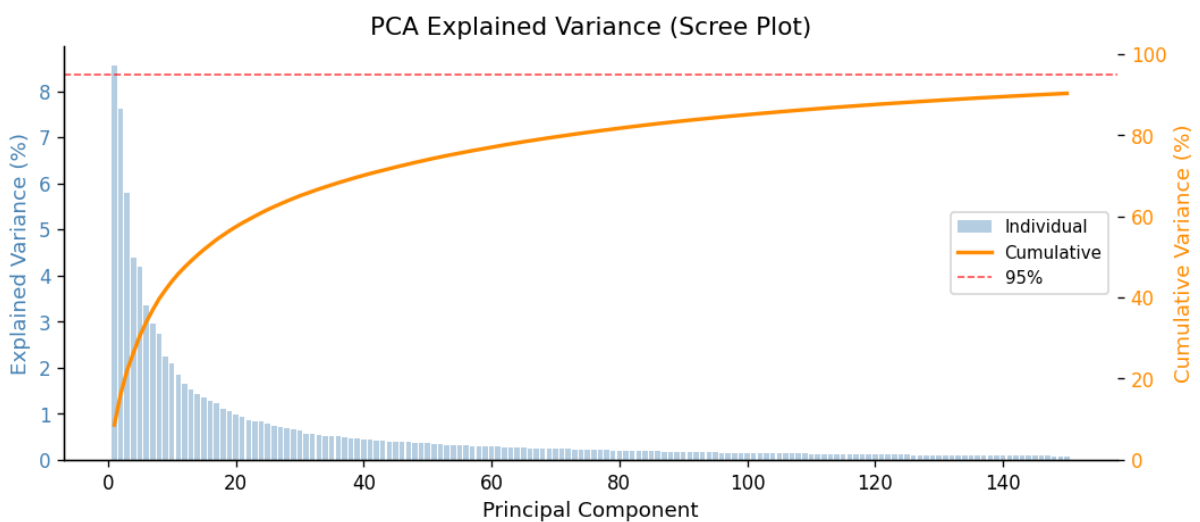


Figura 5.4: Varianța explicată cumulativă în funcție de numărul de componente PCA



Figura 5.5: Primele 16 eigenfaces ale setului de date LFW

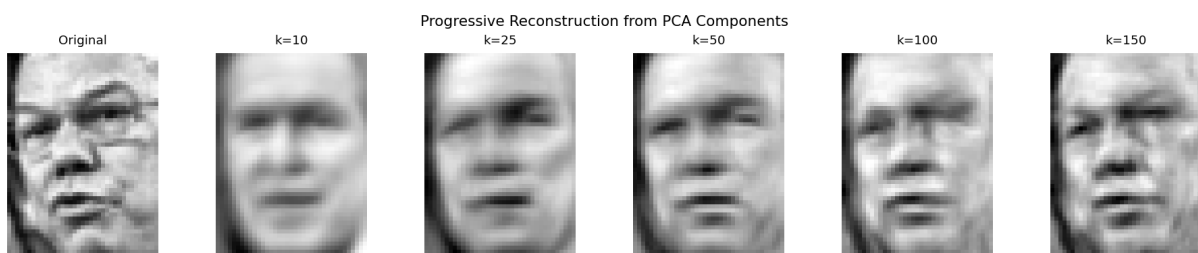


Figura 5.6: Reconstrucția progresivă a unei fețe din  $k$  componente PCA

## 5.6 Comparație cu metoda clasică Eigenfaces

Pentru a izola contribuția SVM-ului față de proiecția PCA, am comparat patru strategii de clasificare în spațiul PCA, toate folosind aceeași împărțire antrenare/test (random\_state = 42,  $k = 150$  componente):

Metodă	Whitening	Acuratețe	F1 macro
PCA + SVM RBF	da	87,3%	0,835
PCA + nearest centroid (Turk-Pentland)	nu	64,3%	0,612
PCA + nearest centroid	da	<b>88,5%</b>	<b>0,853</b>
PCA + 1-NN euclidian	nu	74,2%	0,674
PCA + 1-NN euclidian	da	73,8%	0,673

Tabelul 5.3: Comparație între SVM și clasificatori bazați pe distanța euclidiană în spațiul PCA, pe aceleași date

Rezultatul invită la nuanță. Față de metoda clasică *Eigenfaces* [15], în care Turk și Pentland folosesc *nearest centroid* pe componente PCA fără whitening, SVM aduce un câștig major (87,3% vs. 64,3%). Acest rezultat reproduce, pe LFW filtrat, observația din literatură că un clasificator discriminativ depășește substanțial clasificarea prin distanță euclidiană pe componente brute.

În schimb, când se aplică *whitening* înainte de calculul distanței euclidiene, *nearest centroid* ajunge la 88,5%, depășind ușor SVM. Whitening-ul reechilibrează scala componentelor PCA, transformând distanța euclidiană într-o aproximare a distanței Mahalanobis față de centrozii claselor, ceea ce neutralizează principalul dezavantaj al metodei originale. Pentru cele 6 clase compacte și aproximativ liniar separabile din LFW filtrat, ipoteza implicită a metodei nearest centroid (clase aproximativ sferice în spațiul whitened) este satisfăcută, iar avantajul SVM dispare.

## 5.7 Analiza rezultatelor

**Acuratețe globală.** Modelul atinge 87,3% acuratețe pe setul de test, cu un scor F1 ponderat de 0,87. Performanța variază între clase: George W. Bush, având cel mai mare număr de exemple de antrenare, obține  $F1 = 0,92$ , în timp ce Ariel Sharon (doar 16 exemple de test) prezintă un recall mai scăzut (0,62), explicabil prin variabilitatea ridicată a pozelor.

**Numărul de componente PCA.** Cu puține componente ( $k < 20$ ), clasificatorul nu dispune de suficientă informație pentru a distinge între persoane. Pe măsură ce  $k$  crește, acuratețea se îmbunătățește până la un platou în jurul a  $k = 100-150$ , după care componentele suplimentare aduc preponderent zgomot.

**Rolul *whitening*-ului.** Comparația din Tabelul 5.3 arată că whitening-ul este factorul decisiv pentru ambele clasificatoare (SVM și nearest centroid). Fără whitening, primele componente PCA au varianță cu ordine de mărime mai mare decât ultimele; nucleul RBF, și distanța euclidiană în general, sunt dominate de aceste componente, iar informația din restul este efectiv ignorată [13].

**SVM vs. nearest centroid.** Pe acest dataset, SVM nu aduce un câștig semnificativ peste clasificarea prin distanță euclidiană whitened. Avantajul SVM ar deveni vizibil pe seturi cu clase non-sferice sau cu suprapuneri marcate, unde granițele liniare prin centroizi nu mai sunt suficiente.

## 6 Concluzii

Lucrarea a arătat cum tehnici fundamentale de algebră liniară (descompunerea spectrală, SVD, diagonalizarea matricelor simetrice) stau la baza unor algoritmi practici de recunoaștere facială.

**PCA** realizează o reducere eficientă a dimensionalității: din 1850 de pixeli, 150 de componente principale rețin informația vizuală relevantă pentru identificarea unei persoane. Reducerea este optimă în sensul varianței explicate și decurge direct din descompunerea spectrală a matricii de covarianță.

**SVM** adaugă un nivel suplimentar de clasificare: găsirea hiperplanului cu marginea maximă printr-o problemă de optimizare convexă, rezolvată prin dualitate lagrangiană. Nucleul RBF extinde separarea la cazuri nelineare fără a calcula explicit transformarea în spațiul de trăsături.

Modelul antrenat pe LFW filtrat obține **87,3% acuratețe** pe setul de test, cu un scor F1 ponderat de 0,87. Comparația cu metoda clasică *Eigenfaces* (Secțiunea 5.6) arată că SVM aduce un câștig substanțial față de varianta originală a lui Turk și Pentland (nearest centroid pe componente PCA fără whitening, 64,3%); însă, adăugând whitening, un simplu clasificator prin distanță euclidiană la centroidul clasei atinge 88,5%, ușor peste SVM. Pe acest dataset, ipoteza implicită a metodei nearest centroid (clase aproximativ sferice în spațiul whitened) este satisfăcută, iar avantajul SVM dispare. Concluzia este că, pe LFW filtrat, componenta cu adevărat critică este whitening-ul, nu alegerea clasificatorului.

# Declarație privind utilizarea instrumentelor de inteligență artificială

În conformitate cu art. 6 din Regulamentul UNSTPB privind utilizarea tehnologiilor de inteligență artificială, declarăm că am folosit asistentul de tip IA Claude (Anthropic, <https://claude.ai>; versiunea disponibilă în perioada de redactare, 2026) exclusiv ca instrument de sprijin: reformularea și corectarea limbajului, structurarea textului și sinteza unor surse deja citate în bibliografie. Întrebarea de cercetare, derivările matematice, analiza, rezultatele și concluziile aparțin în întregime autorilor. Pasajele redactate cu sprijinul IA au fost verificate de autori față de sursele primare din bibliografie.

## Bibliografie

- [1] Richard E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.
- [2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92)*, pages 144–152, 1992.
- [3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- [4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [5] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [6] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6–7):417–441, 498–520, 1933.
- [7] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2007. University of Massachusetts, Amherst, Technical Report 07-49.
- [8] Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, New York, 2nd edition, 2002.
- [9] James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.
- [10] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.
- [11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

- [12] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.
- [13] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.
- [14] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.
- [15] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [16] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
- [17] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518, 2001.
- [18] Karel Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics Gems IV*, pages 474–485. Academic Press, 1994.