Universitatea Transilvania din Brașov

ȘCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea de Inginerie Electrică și Știința Calculatoarelor

Ing. Alina-Luminița FECHETE (căs. MACHIDON)

Algoritmi și metode statistice pentru analiza datelor multidimensionale

Algorithms and statistical methods for multidimensional data analysis

REZUMAT / ABSTRACT

Conducător ştiințific

Prof.dr.ing. Petre Lucian OGRUȚAN

BRAȘOV, 2020

Universitatea Transilvania din Brașov

D-lui (D-nei) .....

COMPONENȚA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Braşov
Nr. ..... din .....

PREŞEDINTE:	Conf.dr.ing. Carmen GERIGAN
CONDUCĂTOR ŞTIINȚIFIC:	Prof.dr.ing. Petre Lucian OGRUȚAN
REFERENȚI:	Conf.dr.ing. Fabio DEL FRATE
	Prof.dr.ing. Romulus Mircea TEREBEȘ
	Prof.dr.ing. Cristian GRAVA
	Conf.dr. Daniel Tudor COTFAS

Data, ora şi locul susținerii publice a tezei de doctorat: .....,

.....ora ....., sala .....

Eventualele aprecieri sau observații asupra conținutului lucrării vor fi transmise electronic, în timp util, pe adresa alina.machidon@unitbv.ro

Totodată, vă invităm să luați parte la şedința publică de susținere a tezei de doctorat.

Vă mulțumim.

Universitatea Transilvania din Brașov

CUPRINS

CONTENTS

Introducere

Noutatea și importanța temei de cercetare

Lumea de astăzi este copleșită de o cantitate fără precedent de date (Figura 1), adesea având o dimensionalitate ridicată, ceea ce face ca manipularea unor astfel de seturi de date să fie o sarcină complexă și dificilă. Soluția pentru o procesare și analiză eficientă în acest caz este aplicarea unei etape de pre-procesare constând în reducerea dimensionalității, care "atenuează dezavantajele dimensionalității și ale altor proprietăți nedorite ale spațiilor multidimensionale" [22].

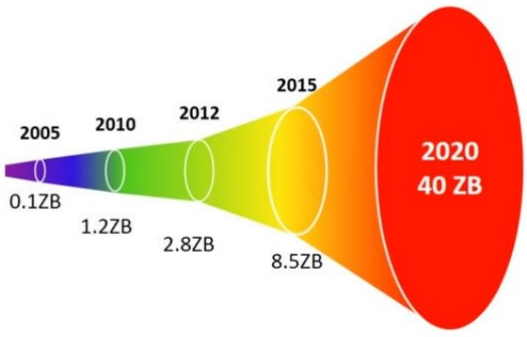


Figure 1: Creșterea explozivă a volumului de date [6].

O categorie de date multidimensionale particulare care a cunoscut o creștere explozivă în ultima decadă sunt datele de Remote Sensing. Acest domeniu este caracterizat de achiziția de informații de la distanță despre suprafața Pământului prin măsurarea radiației electromagnetice [17].

Evoluțiile recente în domeniul Remote Sensing au deschis noi posibilități și, în același timp, au adus noi provocări în ceea ce privește analiza și procesarea cantităților mari de date multidimensionale [20]. În consecință, o sarcină cheie în domeniul analizei hiperspectrale a imaginilor devine gestionarea acestui volum mare de date; această sarcină este realizată în prezent fie prin aplicarea tehnicilor de reducere a dimensionalității, fie prin selectarea unui subset al benzilor

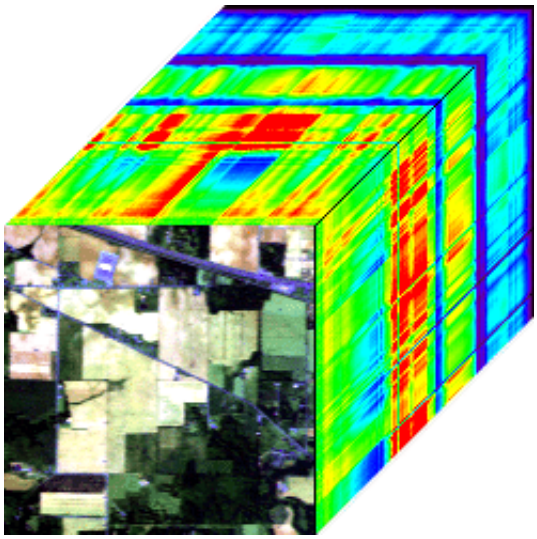


Figure 2: Cubul hiperspectral dintr-o bază de date AVIRIS.

spectrale disponibile [19].

Tehnicile de reducere a dimensionalității (DR) [63] sunt proceduri matematice care transformă "datele cu o dimensionalitate ridicată într-o reprezentare semnificativă de dimensionalitate redusă" [64]. Tehnicile de reducere a dimensionalității sunt denumite și metode de proiecție, și sunt instrumente utilizate pe scară largă pentru aplicațiile de Remote Sensing datorită numeroaselor beneficii aduse [15].

Obiectivele cercetării

Principalul obiectiv al acestei teze este conceperea, proiectarea, implementarea, testarea, validarea și optimizarea unei noi metode de reducere a dimensionalității, legată de tehnica analizei componentelor principale (PCA - Principal Component Analysis).

Principala motivație pentru această cercetare este dorința de a contribui la analiza datelor multidimensionale prin îmbunătățirea metodelor și tehnicilor pentru reducerea dimensionalității, prelucrarea și analiza datelor, vizualizarea și extragerea de informații utile.

Din obiectivul principal derivă următoarele cinci *obiective specifice*:

1. *Identificarea stării actuale a tehnicilor de reducere a dimensionalității*, urmând studiul literaturii de specialitate, evidențiind astfel punctele cheie care pot beneficia de soluții inovatoare.

2. *Proiectarea unei noi metode de reducere a dimensionalității* care sa fie capabilă să adreseze și să reducă deficiențele tehnicilor de reducere a dimensionalității identificate anterior.

3. *Implementarea noii metode.*

4. *Testarea și validarea noii metode* folosind date sintetice și din lumea reală pentru a demonstra eficiența acesteia, prin compararea rezultatelor sale cu alte metode de referință.

5. *Optimizarea noii metode* atât din punct de vedere al calității rezultatelor, cât și al timpului de calcul.

Structura și organizarea tezei de doctorat

În ceea ce privește structura tezei, aceasta este organizată în șase capitole și include 62 de figuri, 47 de tabele și 195 de referințe bibliografice.

- **Introducerea** prezintă oportunitatea și motivația subiectului abordat, obiectivul principal și obiectivele specifice propuse pentru rezolvarea acestuia în cadrul cercetării doctorale, precum și descrierea structurii tezei.
- **Capitolul 1**, "Principal Component Analysis (PCA)" prezintă stadiul actual al metodelor utilizate pentru reducerea dimensionalității, cu accent pe metoda analizei componentelor principale, una din cele mai cunoscute tehnici de reducere a dimensionalității. Avantajele și dezavantajele sale, gama de aplicații în diverse domenii și diferite adaptări sunt rezumate în acest capitol.
- **Capitolul 2**, "Geometrical Approximated Principal Component Analysis" introduce o metodă nouă, gaPCA, bazată pe o construcție geometrică, ca alternativă la metoda standard PCA.
- **Capitolul 3**, "Analiza și clasificarea imaginilor hiperspectrale" prezintă rezultatele obținute după validarea noii metode gaPCA în domeniul imaginilor hiperspectrale, în scopul analizei și clasificării imaginilor. Performanța metodei gaPCA a fost evaluată folosind mai multe metrici, pentru evaluarea calității imaginii, calitatea reconstrucției, redundanța informațiilor și acuratețea clasificării, iar rezultatele au fost comparate cu cele ale metodei PCA.
- **Capitolul 4**, "Recunoaștere facială" prezintă rezultatele noii metode gaPCA în domeniul recunoașterii faciale, având ca reper metoda PCA.
- **Capitolul 5**, "Paralelizare" prezintă implementarea noii metode gaPCA folosind arhitecturi de calcul paralel pentru accelerarea timpilor de calcul și pentru a obține o eficiență energetică îmbunătățită.
- **Capitolul 6**, "Concluzii finale și contribuții originale", prezintă într-o abordare sistematică concluziile capitolelor anterioare și analizează contribuțiile originale realizate în această cercetare doctorală. De asemenea, o serie de direcții de cercetare viitoare, care vor fi urmărite pentru a continua și dezvolta contribuțiile existente, este prezentată.

CAPITOLUL 1

Principal Component Analysis (PCA)

Potrivit dicționarului Merriam-Webster, termenul "multidimensional" înseamnă "având dimensiuni sau aspecte multiple" [8]. Deși, în sensul strict, seturile de date care au mai mult de două dimensiuni sunt considerate multidimensionale, cuvântul "multidimensional" este utilizat în general numai pentru seturi de date cu mai mult de trei dimensiuni [21].

Reducerea dimensionalității se referă la "transformarea datelor cu dimensionalitate ridicată într-o reprezentare semnificativă de dimensionalitate redusă" [37]. Scopul este de a găsi o reprezentare nouă care să poată reprezenta datele originale într-un spațiu dimensional inferior, astfel încât relațiile cheie intrinseci ale datelor să fie păstrate.

Analiza componentelor principale aparține familiei de metode Projection Pursuit (PP) [27], o clasă de metode non-parametrice statistice pentru analiza datelor propusă inițial de Friedman și Tukey în 1974 [27], care presupune căutarea de direcții în spațiul multidimensional pe care datele pot fi proiectate.

1.1 Descrierea metodei

Algoritmul PCA produce un set nou de variabile, în care fiecare reprezintă o combinație liniară a caracteristicilor inițiale [7]. Aceste coordonate, care sunt, de asemenea, numite componente principale (PC-uri), sunt perpendiculare una pe alta și prezintă o ordonare specifică (adică fiecare PC indică direcția dată de varianța maximă și este ortogonală pe fiecare dintre PC-urile anterioare). Prima componentă este un vector în spațiul inițial, definit de distanța euclidiană medie pătrată minimă între aceasta și datele inițiale. A doua componentă este o altă linie în spațiu, ortogonală cu prima și care satisface aceeași metrică a distanței euclidiene pătrate minime și așa mai departe [60] (Figura 1.1).

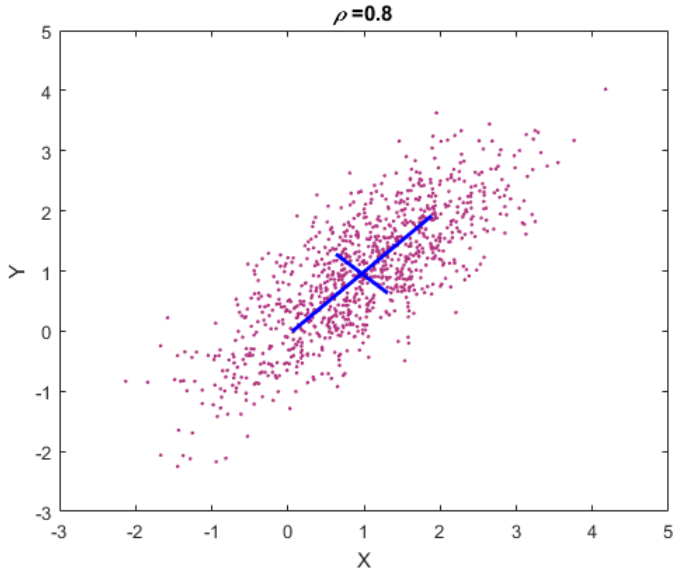


Figure 1.1: Axele componentelor principale pe un nor de puncte bidirecțional, corelat, cu o distribuție normală.

1.2 Aplicații ale PCA

PCA poate fi aplicat în aproape toate disciplinele științifice pentru sarcini precum reducerea dimensionalității, extragerea de informații utile, reducerea zgomotului, vizualizare, compresie. De exemplu, în domeniul îmbunătățirii imaginii, metoda PCA a fost utilizată pentru a îmbunătăți contrastul și a elimina zgomotul pentru imaginile nocturne [56]. În biologie, această metodă a fost utilizată pentru predicția interacțiunilor proteină-proteină [65]. În silvicultură, metoda PCA a fost utilizată pentru analiza relațiilor sol-vegetație [24]. În sociologie, tehnica PCA a fost utilizată pentru analiza factorilor socioeconomici și asocierea lor cu malaria și riscul de arbovirus în Tanzania [29]. Avantajele utilizării PCA se extind și la alte domenii precum finanțe [13], criminalistică și restaurare de artă, de exemplu.

1.3 Avantaje și limitări ale PCA

Printre principalele avantaje ale PCA ([62], [28], [34]), se pot enumera:

- PCA oferă o reprezentare de o dimensionalitate redusă a datelor, deoarece doar câteva componente principale sunt necesare pentru a reține majoritatea informației din setul de date original.
- PCA permite reconstrucția (parțială) a datelor de intrare. Toate metodele de reducere a dimensionalității pierd unele informații, dar PCA este recunoscută pentru capacitatea sa de a minimiza această pierdere.
- PCA este generică. Deoarece această metodă nu este ajustată pentru o sarcină specifică, are o flexibilitate ridicată și poate fi folosită într-o gamă variată de aplicații.

- PCA reduce eficient corelația și redundanța datelor. Variabilele rezultate sunt necorelate între ele.

Cu toate acestea, există și unele probleme legate de PCA ([62], [28], [34], [60]) cum ar fi:

- PCA se bazează pe presupunerea că variațele mari sunt cele mai importante. Deci, componentele principale care corespund variațelor mai mari sunt mai interesante decât cele cu variațe mai mici. Având în vedere că unul dintre obiectivele PCA este de a găsi proiecțiile pe fiecare componentă dată de direcția de variață maximă, înseamnă că în anumite cazuri, datele cu o variață mai mică pot fi ignorate și greșit etichetate drept "zgomot", deși pot fi de fapt date de interes;

- Flexibilitatea PCA vine de asemenea, cu un preț, constând în cerințe de calcul relativ mai mari în comparație cu, de exemplu, Transformata Fourier Rapidă și, de asemenea, dificultăți inerente de paralelizare a algoritmului pentru a beneficia de arhitecturile de tip High Performance Computing.

1.4 Adaptări și aproximări ale PCA

Mai multe cercetări au explorat cele mai bune soluții de a depăși aceste limitări prin extinderea, adaptarea sau chiar reinventarea PCA. În consecință, în ultimele decenii au fost dezvoltate mai multe adaptări ale metodei PCA standard, fieare concentrându-se pe tipuri de date, structuri și aplicații specifice [32], rezultând numeroase extensii sau variante PCA.

Diverse încercări de "îmbunătățire" a metodei PCA canonice au fost propuse de literatura științifică, care vizează "imunizarea" metodei la erorile din seturile de date [35] [18]. Independent Component Analysis produce "o reprezentare a noilor variabile care sunt independente una de cealaltă, nu numai necorelate" [38]. Metoda PCA neliniară [61] abordează problema liniarității; transformă neliniar datele ordinale în date cantitative [51]. Această metodă se bazează pe backpropagation pentru instruirea unui perceptron multistrat (MLP), actualizând atât ponderile cât și intrările [61].

1.5 Concluzii

Acest capitol a prezentat o imagine de ansamblu a celor mai cunoscute tehnici de reducere a dimensionalității, cu accent pe analiza componentelor principale (descrierea metodei, avantajele și limitările și aplicarea acesteia în diverse domenii).

Acest capitol subliniază importanța tehnicilor de reducere a dimensionalității, în special a PCA, în diferite domenii pentru prelucrarea seturilor de date multidimensionale mari. În contextul Big Data și furnizarea continuă a unor cantități mari de date de la senzori, dispozitive de achiziție etc., există un interes din ce în ce mai mare pentru o astfel de metodă, în special adaptările bazate pe PCA, care permit îmbunătățirea capacității de extragere a informației. În lumina celor de mai sus, această cercetare doctorală s-a concentrat pe dezvoltarea de algoritmi

îmbunătățiți pentru reducerea dimensionalității, capabili să ofere noi informații și să extragă informații semnificative din seturi de date multidimensionale.

CAPITOLUL 2

Geometrical Approximated Principal Component Analysis (gaPCA)

2.1 Introducere în metoda inovativă gaPCA

GaPCA este o metodă inovatoare și originală care calculează componentele principale ale unui set de date multidimensional pe baza direcției date de extremitățile distribuției (punctele din setul de date separate de distanța maximă), oferind, prin urmare, o estimare a direcției componentelor principale ale PCA standard. Metoda PCA standard calculează componentele principale bazate pe direcția varianței maxime a datelor, prin calcularea vectorilor proprii ai matricei de covarianță a setului de date. Dezavantajul este, însă, că acești vectori proprii au tendința de a trece cu vederea informațiile date de elementele (aparent) minore din setul de date, care sunt considerate ca având o contribuție mai mică la varianța totală.

O caracteristică unică care diferențiază gaPCA de PCA-ul canonic este faptul că ordonarea componentelor gaPCA (care sunt fiecare ortogonale reciproc) este cea rezultată din iterațiile algoritmului (componentele gaPCA nefiind ordonate în niciun fel), în timp ce în cazul PCA, acestea sunt ordonate în funcție de varianță. În consecință, spre deosebire de metoda standard (unde informațiile utile se regăsesc în mod preponderent în primele componente), în cazul gaPCA, aceste informații sunt mai dispersate între componente [33].

2.2 Descrierea metodei

Etapa inițială a gaPCA constă în normalizarea setului de date de intrare, scăzând media. Dat fiind un set de n–dimensional, $\mathbf{P_0} = \{\mathbf{p_{01}}, \mathbf{p_{02}}, \ldots\} \subset \mathbb{R}^n$, media μ este calculată și scăzută.

$$\mathbf{P_1} = \mathbf{P_0} - \mu; \tag{2.1}$$

Prima componentă principală gaPCA este calculată ca fiind vectorul $\mathbf{v_1}$ care conectează cele

două puncte: $\mathbf{v_1} = \mathbf{e_{11}} - \mathbf{e_{12}}$, separate de distanța euclidiană maximă:

$$\{\mathbf{e_{11}}, \mathbf{e_{12}}\} = \arg \max_{\mathbf{p_{1i}}, \mathbf{p_{1j}} \in P1} d(\mathbf{p_{1i}}, \mathbf{p_{1j}}) \tag{2.2}$$

unde $d(\cdot, \cdot)$ reprezintă distanța euclidiană.

Al doilea vector componentă principală este calculat ca diferența dintre cele două proiecții ale elementelor originale din $\mathbf{P_1}$ pe hiperplanul $\mathbf{H_1}$, determinat de vectorul normal $\mathbf{v_1}$ și care conține $\mathbf{o}$, originea:

$$H_1 = \{\mathbf{x} \in \mathbb{R}^n | < \mathbf{v_1}, \mathbf{x} >=< \mathbf{v_1}, \mathbf{o} >\} \tag{2.3}$$

unde $< \cdot, \cdot >$ reprezintă produsul scalar. $P_2 = \{\mathbf{p_{21}}, \mathbf{p_{22}}, \ldots\}$ reprezintă punctele inițiale proiectate, calculate după următoarea formulă:

$$\mathbf{p_{2i}} = \mathbf{p_{1i}} + (< \mathbf{v_1}, \mathbf{o} > - < \mathbf{v_1}, \mathbf{p_{1i}} >) \cdot \mathbf{v_1} / ||\mathbf{v_1}||^2 \tag{2.4}$$

Ca atare, vectorul de bază cu indicele i este calculat proiectând $\mathbf{P_{i-1}}$ pe hiperplanul $\mathbf{H_{i-1}}$, găsind proiecțiile separate de distanță maximă și calculând diferența lor, $\mathbf{v_i}$.

Algoritmul gaPCA este format din 2 faze iterative principale, fiecare fază fiind executată de mai multe ori, în funcție de numărul prevăzut de componente principale care urmează să fie calculate:

1. Determinarea vectorului de proiecție definit de cele două extremități ale setului de date (separate de distanța maximă);
2. Realizarea reducerii dimensionalității pe setul de date proiectându-l pe subspațiul ortogonal la proiecția anterioară.

Pentru a reconstrui datele originale, scorurile componentelor $\mathbf{S}$ (calculate prin proiectarea tuturor punctelor pe componentele principale) sunt calculate (într-o manieră similară algoritmului PCA), prin înmultirea datelor centrate în jurul mediei cu matricea conținând vectorii componente principale.

$$\mathbf{S} = \mathbf{P_1} \cdot \mathbf{v} \tag{2.5}$$

Datele originale pot fi reconstruite multiplicând scorurile S cu matricea componentelor principale transpuse și adăugând media.

$$\mathbf{P_0} = \mathbf{S} \cdot \mathbf{v^T} + \mu \tag{2.6}$$

Rezultatul final este baza vectorială $\mathbf{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_n}\}$. Reprezentarea datelor n–dimensionale folosind $\mathbf{V}$ se face folosind produsul scalar; astfel, în noua reprezentare, componenta $\mathbf{c_i}$ a unui vector $\mathbf{x} = [x_1, x_2, ..., x_n]$ este calculată ca: $\mathbf{c_i} =< \mathbf{x}, \mathbf{v_i} >$.

2.3 Algoritmul gaPCA

Algoritmul 1 conține pseudocodul pentru metoda gaPCA.

Algorithm 1: gaPCA

Input: $P_1 = [p_{11}, p_{12}, \ldots, p_{1m}]$, k where:
m = number of pixels of the image ;
p_{1i} = the pixel vector (n bands) of the i-th pixel ;
k = the number of principal components to be computed ;
Output: $v = [v_1, v_2, \ldots v_k]$;
e_{11}, e_{12} = computeMaximumDistance(P_1);
v_1 = (e_{11}-e_{12}) / $||v_1||$;
o = mean(P_1);
for $i \leftarrow 2$ **to** k **do**
    P_i = computeProjectionsHyperplane(P_{i-1}, o, v_{i-1});
    e_{i1}, e_{i2} = computeMaximumDistance(P_i);
    v_i = (e_{i1}-e_{i2})/ $||v_i||$;
end
return v

Algoritmul 2 ilustrează pseudocodul pentru metoda care calculează toate distanțele euclidiene între toate punctele unei matrici P.

Algorithm 2: computeMaximumDistance

Input:$P = [p_1, p_2, \ldots, p_m]$;
Output: e_1, e_2 ;
$e_1 = 0$;
$e_2 = 0$;
$distMax = 0$;
for $i \leftarrow 1$ **to** $m - 1$ **do**
    for $j \leftarrow i + 1$ **to** m **do**
        $dist$ = EuclideanDistance($P[i]$, $P[j]$) ;
        if *($dist > distMax$)* **then**
            $distMax = dist$;
            $e_1 = P[i]$;
            $e_2 = P[j]$;
        end
    end
end
return e_1, e_2

Algoritmul 3 prezintă pseudocodul pentru rutina care calculează proiecțiile euclidiene ale fiecărei intrări a matricei P, pe hiperplanul dat de vectorul ortogonal v și incluzând media

datelor md.

Algorithm 3: computeProjectionsHyperplane

Input: $P = [p_1, p_2, \ldots, p_m]$, o, v ;
Output: $R = [r_1, r_2, \ldots, r_m]$;
for $i \leftarrow 1$ **to** m **do**
  $R[i] = P[i] + (< v, o > - < v, P[i] >) \cdot v/||v||^2$;
end
return R

Algoritmul 4 conține pseudocodul pentru metoda care calculează distanța euclidiană între vectorul P și vectorul R.

Algorithm 4: EuclideanDistance

Input: $P = [p_1, p_2, \ldots, p_m]$, $R = [r_1, r_2, \ldots, r_m]$;
Output: d ;
$d = 0$;
for $i \leftarrow 1$ **to** m **do**
  $d = d + (P[i] - R[i])^2$;
end
return sqrt(d)

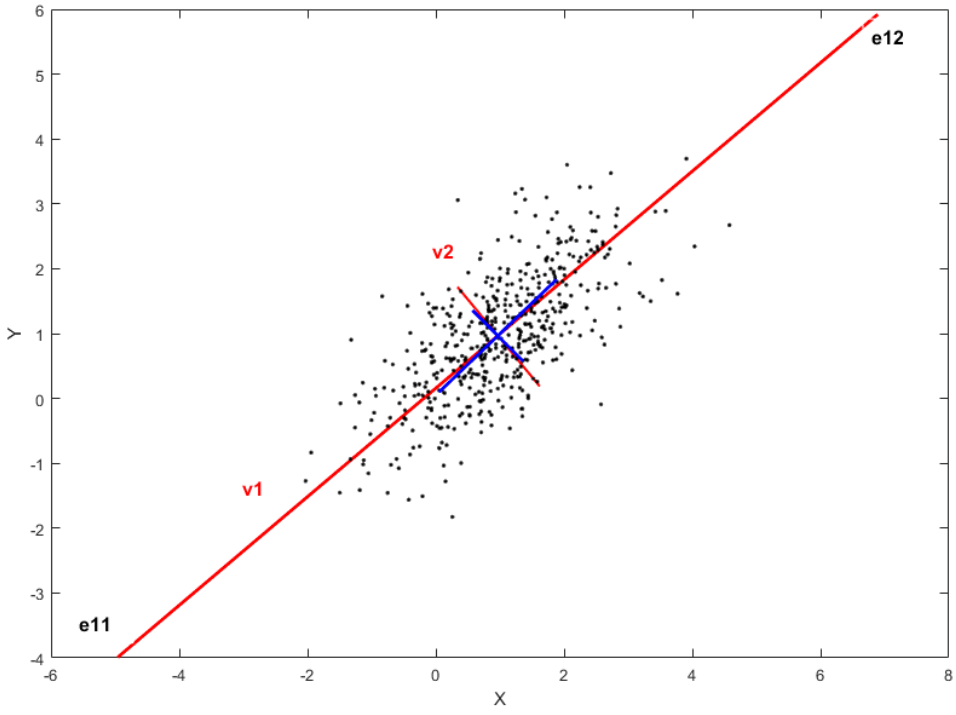


Figure 2.1: Axele PCA (albastru) vs. gaPCA (roșu) într-un nor de puncte 2D.

Figura 2.1 și Figura 2.2 ilustrează o reprezentare grafică a modului în care gaPCA și PCA calculează componentele principale pentru un set generat aleatoriu de puncte 2D având o

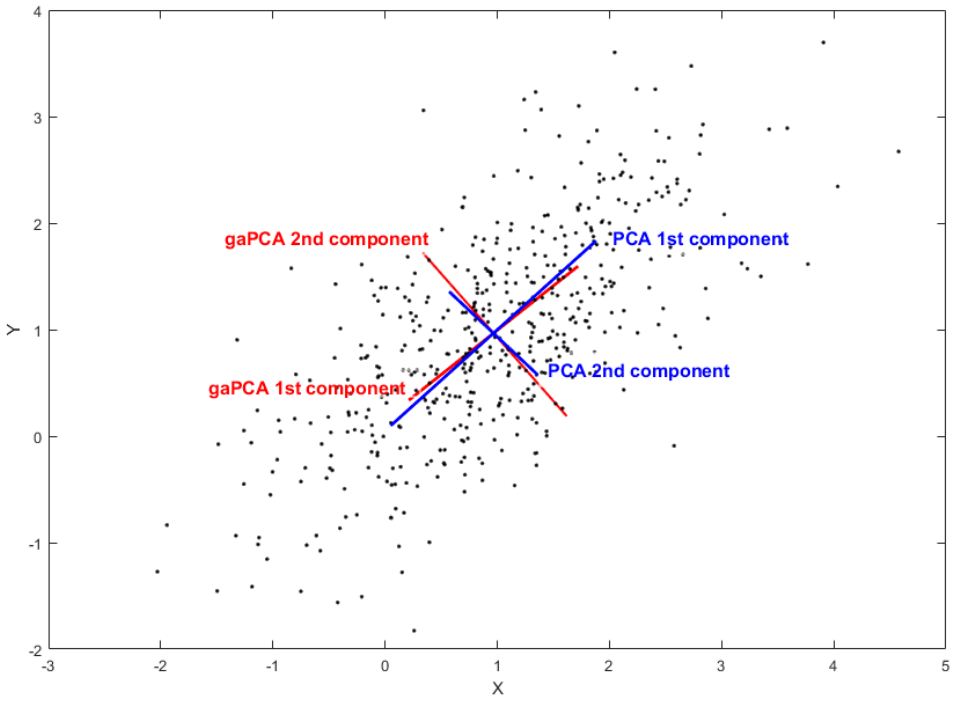


Figure 2.2: Axele normate ale PCA (albastru) vs. gaPCA (roșu) într-un nor de puncte 2D.

distribuție normală. Figura 2.3 prezintă calculul vectorilor de bază pentru un set tridimensional de puncte generat.

2.4 Validarea pe date sintetice

Pentru validarea metodei gaPCA pe date sintetice, au fost generate 100 de seturi aleatorii de 1000 de puncte bidimensionale fiecare, cu o distribuție gaussiană (această distribuție a fost aleasă deoarece este acceptată pe scară largă ca model statistic pentru majoritatea tipurilor de date[50]). Datele generate au fost corelate cu un coeficient de corelație incremental de la 0,5 până la 1. Pentru aceste experimente, datele nu au fost centrate in jurul mediei înainte de calcularea metodei gaPCA.

În fiecare caz au fost folosite două metrici pentru a evalua apropierea geometrică între gaPCA și metoda PCA standard:

(i) unghiul format de prima componentă PCA și respectiv de cea gaPCA, adică unghiul între linia dată de punctele separate de distanța maximă și de prima componentă PCA;

(ii) distanța dintre media setului de date și punctul mediu al segmentului dat de punctele cele mai îndepărtate din setul de date.

Figura 2.4 ilustrează trei nori bidimensionali generați aleatoriu de puncte (negru) având valori diferite ale coeficientului de corelație. Pentru fiecare set de date, au fost calculate primele 2

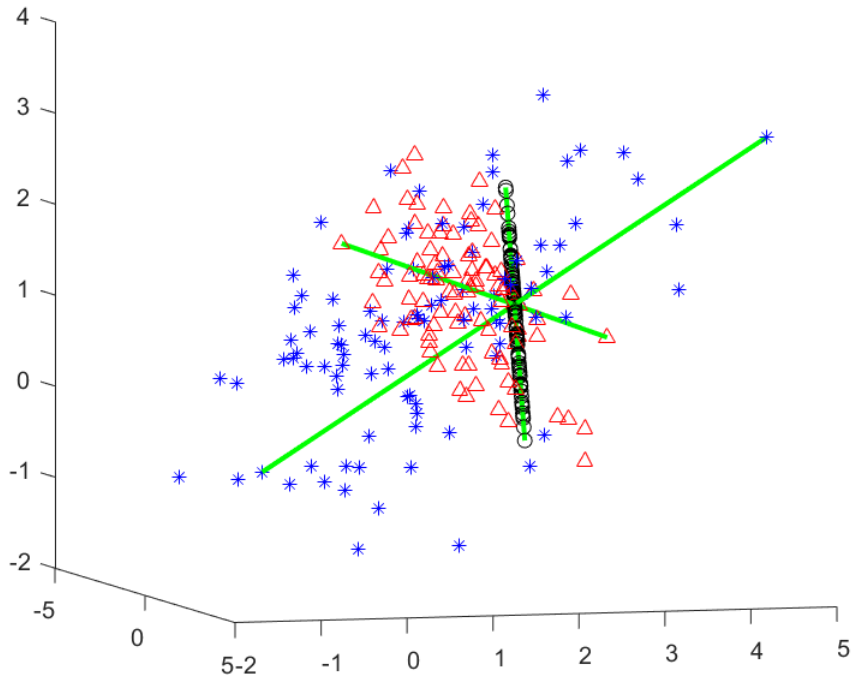


Figure 2.3: Axele gaPCA (verde) într-un nor de puncte tridimensionale, corelate.

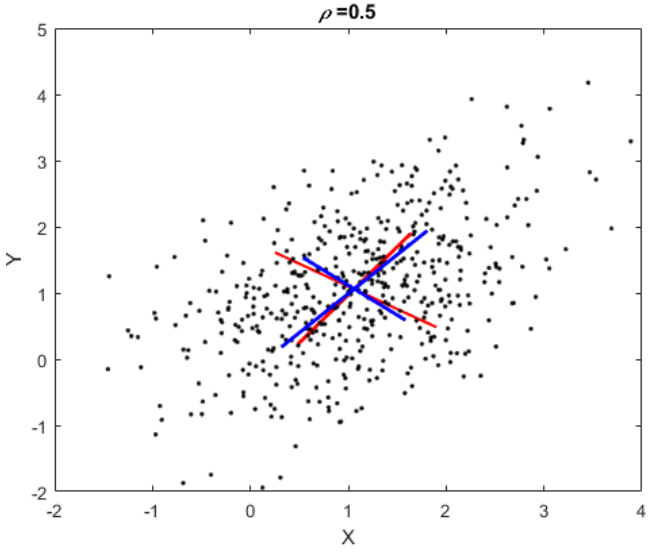
componente principale, atât pentru gaPCA (roșu), cât și pentru standard PCA (albastru). Se observă că tendința unghiului este să scadă pentru valori mai mari ale ρ. Acest fenomen dovedește că precizia aproximării PCA dată de gaPCA crește pe măsură ce variabilele sunt mai corelate.

Figura 2.5 prezintă unghiul și distanța ca reprezentări de tip boxplot, variind cu coeficientul de corelație ρ al setului de date. Aceste rezultate arată că, în medie, ambele metrici prezintă o tendință descrescătoare odată cu creșterea ρ.

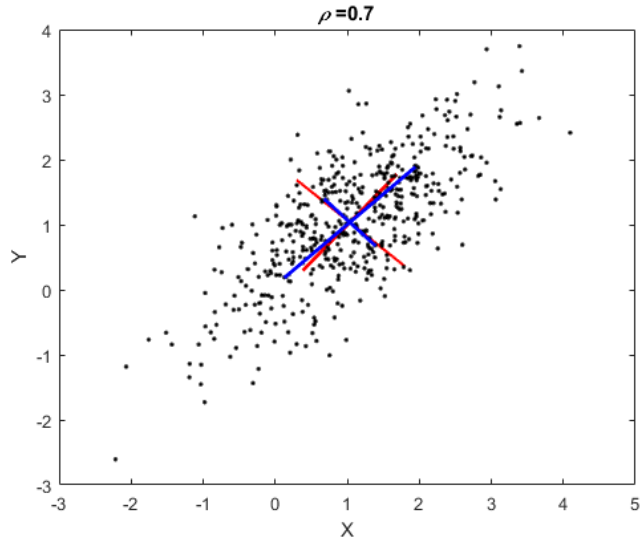
2.5 Concluzii

În acest capitol a fost introdus un algoritm alternativ PCA bazat pe o construcție geometrică, și anume metoda gaPCA, care se bazează pe ipoteza că într-un set de date multidimensional segmentul care leagă punctele cele mai îndepărtate dă o direcție relativ apropiată de cea dată de prima componentă principală a PCA. A fost prezentată validarea preliminară a metodei gaPCA, baza de comparație fiind algoritmul PCA standard.

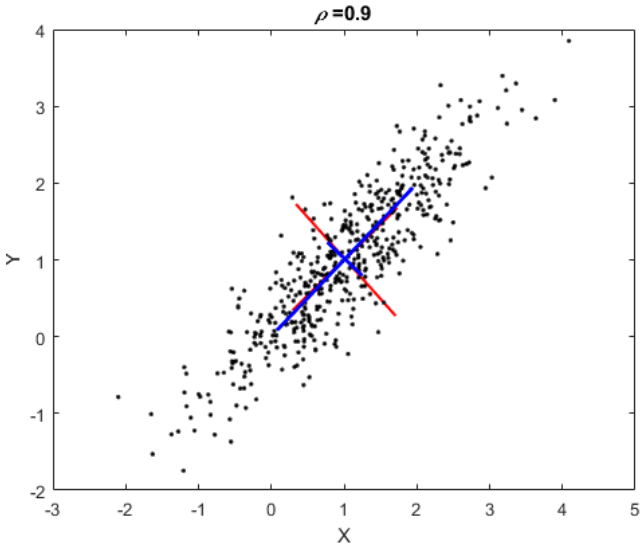
Astfel, gaPCA a fost validat pe seturi de date sintetice bidimensionale cu o distribuție gaussiană, asigurând condiția de normalitate. Pentru o evaluare comparativă a performanței gaPCA asupra datelor sintetice, rezultatele acesteia au fost analizate comparativ cu rezultatele PCA standard, prin calcularea unghiului și a distanței între principalele componente ale celor două metode. Rezultatele obținute arată că ambele metrici pentru majoritatea seturilor de date sunt sub 10%, ceea ce confirmă o fidelitate ridicată a metodei gaPCA în comparație cu PCA standard, pe datele sintetice.



(a) ρ=0.5

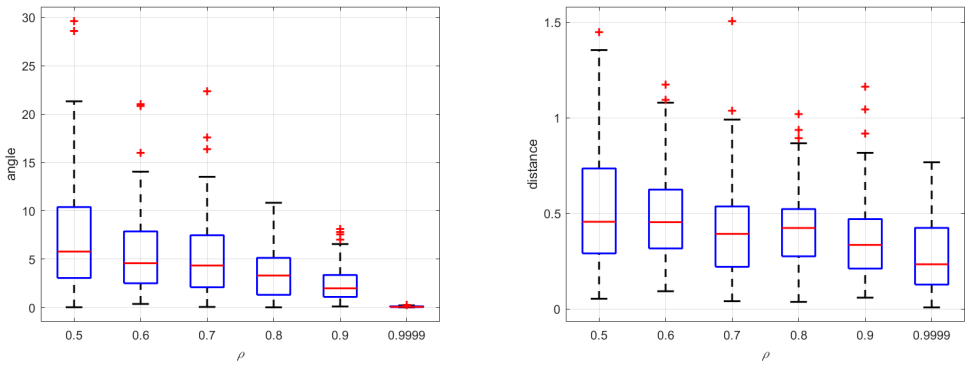


(b) ρ=0.7



(c) ρ=0.9

Figure 2.4: Axele gaPCA vs. PCA pentru seturi de date bidimensionale cu diverși coeficienți de corelație (ρ).



(a) (b)

Figure 2.5: Unghiul (a) și distanța (b) între axele gaPCA și PCA - reprezentare de tip boxplot.

Cercetările și experimentele prezentate în acest capitol au fost validate și diseminate în următoarele publicații științifice:

- [45] A. L. Machidon, F. Del Frate, M. Picchiani, O. M. Machidon, and P. L. Ogrutan. Geometrical Approximated Principal Component Analysis for Hyperspectral Image Analysis. *Remote Sensing*, 12(11), 2020
- [26] L. Fasano, D. Latini, A. L. Machidon, C. Clementini, G. Schiavon, and F. Del Frate. SAR Data Fusion Using Nonlinear Principal Component Analysis. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019 also indexed in IEEE Xplore Digital Library
- [44] A. L. Machidon, R. Coliban, O. Machidon, and M. Ivanovici. Maximum Distance-based PCA Approximation for Hyperspectral Image Analysis and Visualization. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018 also indexed in IEEE Xplore Digital Library
- [25] L. Fasano, F. Del Frate, D. Latini, A. L. Machidon, and C. Clementini. Classification of Urban areas by Means of Multiband SAR Data Fusion. In *European Space Agency 2nd Mapping Urban Areas from Space 2018 - MUAS 2018*. ESA, 2018

CAPITOLUL 3

Analiza și clasificarea imaginilor hiperspectrale

Progresele recente în domeniul Remote Sensing au deschis noi perspective și oportunități dar au adus și provocări inerente legate de procesarea și analiza volumelor mari de date. Prin urmare, creșterea cantității și calității imaginilor hiperspectrale disponibile a determinat lărgirea spectrului informațional, dar și creșterea considerabilă a complexității computaționale asociată seturilor mari de date. Una dintre principalele provocări este gestionarea unui număr mare de benzi hiperspectrale, care are ca efect creșterea substanțială a timpul de procesare și a complexității computaționale. În acest context, reducerea eficientă a dimensionalității datelor din domeniul Remote Sensing reprezintă o sarcină critică în analiza imaginilor hiperspectrale [59].

PCA, ca tehnică de referință pentru reducerea dimensionalității, este utilizată frecvent în aplicații de imagistică hiperspectrală ca metodă de pre-procesare a datelor. Cele mai utilizate aplicații pentru PCA în domeniul Remote Sensing, includ clasificarea imaginilor [59], [52], recunoașterea caracteristicilor [14] și detectarea schimbărilor (identificarea modificărilor în anumite zone din cadrul mai multor serii de imagini temporale) [57], vizualizare [39] sau compresie [23].

În acest context, acest capitol prezintă aplicațiile noii metode gaPCA în domeniul analizei și vizualizării datelor hiperspectrale de Remote Sensing [44], [25], [46].

3.1 Metrici de evaluare

Performanțele metodei gaPCA au fost evaluate calitativ, dar și cantitativ. Primul criteriu a fost aplicat pentru evaluarea imaginilor obținute în termeni de metrici de anliză texturală (Gray level co-occurrence matrix -GLCM : contrast, entropie, energie), calitatea reconstrucției (Signal to Noise Ratio -SNR, Peak Signal to Noise Ratio -PSNR, Root Mean Square Error -RMSE, Spectral Angle Mapper -SAM și Correlation Coefficient -CC) și redundanța componentelor

principale (prin calcularea indicelui de Mutual Information - MI), în timp ce ultimul a implicat evaluarea acurateții clasificării obținute pe componentele principale gaPCA.

Având în vedere că metoda PCA, alături de alți algoritmi similari, este utilizată în mod eficient pentru reducerea dimensionalității datelor de Remote Sensing, extragerea informațiilor privind acoperirea terestră sau pentru extragerea caracteristicilor [42], această secțiune a avut în vedere o evaluare a eficienței metodei gaPCA în domeniul clasificării suprafețelor terestre, având algoritmul PCA standard ca bază de referință pentru evaluarea comparativă.

Pentru fiecare set de date, a fost calculat un număr specific de componente principale, determinat în fiecare caz, pe baza criteriului de a obține maximum de varianță explicată (de exemplu 98-99 %) printr-un număr minim de componente principale. Pe baza acestei reguli, pentru fiecare set de date a fost calculat următorul număr de componente: Indian Pines - 10, Pavia University - 4, DC Mall - 3 și AHS - 3. Primele componente principale calculate atât de metoda gaPCA, cât și de PCA reprezintă benzile imaginilor pe care s-a efectuat clasificarea, folosind software-ul ENVI [4]. În ceea ce privește algoritmii de clasificare, pentru toate seturile de date și pentru fiecare din cele două metode, s-au utilizat algoritmii Maximum Likelihood (ML) și Support Vector Machine (SVM). Pentru evaluarea acurateții clasificării în fiecare caz, un set de pixeli a fost generat aleator și s-a realizat inspecția vizuală având ca reper imaginea de referință.

Acuratețea clasificării a fost calculată folosind două metrici: acuratețea generală (OA constând în numărul de eșantioane clasificate corect per numărul de probe) și coeficientul kappa de acord (k care reprezintă procentul de acord corectat cu valoarea acordului care ar putea fi de așteptat din cauza întâmplării). Pentru a evalua semnificația statistică a rezultatelor clasificării date de cele două tehnici, a fost efectuat testul McNemar [49] pentru fiecare clasificator în parte.

3.2 Rezultate experimentale și discuție

3.2.1 Metrici de analiză texturală folosind GLCM

Metricile de analiză texturală GLCM au fost folosite pentru evaluarea calității imaginilor componente principale calculate cu ambele metode, deoarece sarcina de clasificare a suprafețelor terestre este efectuată pe acestea. Scopul acestei analize este de a evalua atât calitatea, cât și cantitatea de informație a imaginilor reprezentate de componentele principale corespunzătoare ambelor metode, deoarece acești factori influențează acuratețea rezultatelor clasificării. Cele trei metrici GLCM (contrast, energie, entropie) au fost calculate atât pe imaginile componente gaPCA, cât și pe cele PCA, iar numărul de componente calculate a fost același pentru ambele metode și a fost în concordanță cu numărul utilizat în toate experimentele (Indian Pines - 10, Pavia University - 4, DC Mall - 3, AHS - 3).

Tabelul 3.1 prezintă valorile metricii contrast calculate pentru fiecare imagine principală a componentelor și mediată, în ambele cazuri (gaPCA și standard PCA). În cazul seturilor de date Indian Pines și Pavia University, componentele principale gaPCA au prezentat valori de contrast medii mai mari în comparație cu PCA canonic, în timp ce pentru celelalte seturi de date tendința este inversă. Valorile medii ale energiei calculate pe imaginile componente principale produse atât de gaPCA cât și de PCA pentru toate cele 4 seturi de date sunt afișate

în Tabelul 3.2. Cifrele arată că imaginile componente principale gaPCA au un scor mai bun în ceea ce privește calitatea imaginii (deci valori ale energiei mai mici) decât PCA pentru toate seturile de date, cu excepția Indian Pines. În ceea ce privește entropia, rezultatele prezentate în Tabelul 3.3 sunt perfect corelate cu cele referitoare la contrast, gaPCA având valori ale entropiei mai mari (și deci scoruri mai bune) decât PCA pentru DC Mall și AHS, în timp ce în celelalte două cazuri, situația este inversă.

Analiza GLCM a gaPCA și PCA a arătat că, în ceea ce privește contrastul și entropia, cele două metode se comportă similar, în timp ce în cazul energiei gaPCA a dovedit o calitate spațială superioară a imaginilor componente principale în comparație cu cele PCA, un fapt care poate duce la rezultate mai bune ale clasificării.

Table 3.1: Metrica contrast GLCM pentru ambele metode pe toate seturile de date.

	Indian Pines	**Pavia University**	**DC Mall**	**AHS**
PCA	0.96	0.14	0.25	0.17
gaPCA	0.34	0.12	0.32	0.18

Table 3.2: Metrica energie GLCM pentru ambele metode pe toate seturile de date.

	Indian Pines	**Pavia University**	**DC Mall**	**AHS**
PCA	0.14	0.58	0.29	0.23
gaPCA	0.21	0.53	0.20	0.21

Table 3.3: Metrica entropie GLCM pentru ambele metode pe toate seturile de date.

	Indian Pines	**Pavia University**	**DC Mall**	**AHS**
PCA	6.95	5.28	6.07	6.72
gaPCA	6.61	5.17	6.38	6.75

3.2.2 Metrici privind calitatea reconstrucției

În această subsecțiune, obiectivul a fost de a evalua relația dintre imaginea reconstruită și setul de date inițial. Deoarece una dintre principalele avantaje ale metodei PCA este capacitatea sa de a păstra principalele caracteristici ale datelor, în doar câteva componente, câteva metrici bine stabilite au fost utilizate, pentru a evalua calitatea imaginilor reconstruite folosind atât PCA cât și gaPCA. Experimente au fost realizate pentru a urmări conexiunea dintre numărul de componente principale utilizate pentru reconstrucție și calitatea informațiilor păstrate. Setul de date Indian Pines a fost folosit pentru efectuarea analizei comparative. Numărul componentelor principale care au fost utilizate pentru reconstrucție a variat între 1 și 200 pentru ambele metode, pentru a evidenția cum numărul componentelor principale utilizate pentru reconstrucție afectează calitatea imaginii reconstruite. Prima metrică evaluată a fost

Table 3.4: RMSE pentru setul de daate Indian Pines.

	1PC	**2PC**	**10PC**	**100PC**	**200PC**
gaPCA	0.06	0.05	0.02	0.00	0.00
PCA	0.22	0.13	0.08	0.05	0.00

RMSE, calculată între imaginea inițială și cea reconstruită cu PCA canonic și apoi cu gaPCA. Rezultatele sunt afișate în Tabelul 3.4 și reprezentate în Figura 3.1(a).

Rezultatele arată că RMSE folosind metoda gaPCA sunt semnificativ mai mici decât cele ale PCA. Se poate observa că valorile RMSE tind la zero, pentru ambele metode, pentru gaPCA rezultatele sunt mai bune, mai ales în spațiile dimensionale mai mici (până la 10 componente principale calculate, caz adesea întâlnit în aplicațiile din lumea reală).

A doua metrică folosită a fost SNR. Valorile calculate între imaginea inițială și cea reconstruită cu PCA și respectiv gaPCA sunt afișate în Tabelul 3.5 și Figura 3.1(b).

Table 3.5: SNR pentru setul de date Indian Pines.

	1PC	**2PC**	**10PC**	**100PC**	**200PC**
gaPCA	13.47	15.39	24.84	42.19	275.66
PCA	10.97	24.33	26.46	35.86	303.67

Se poate observa că gaPCA depășește PCA în ceea ce privește SNR atunci când este utilizată o componentă principală, sau în cazul a 100 de componente principale, de exemplu, în timp ce PCA are rezultate mai bune pentru 2 componente și atunci când toate componentele principale sunt utilizate.

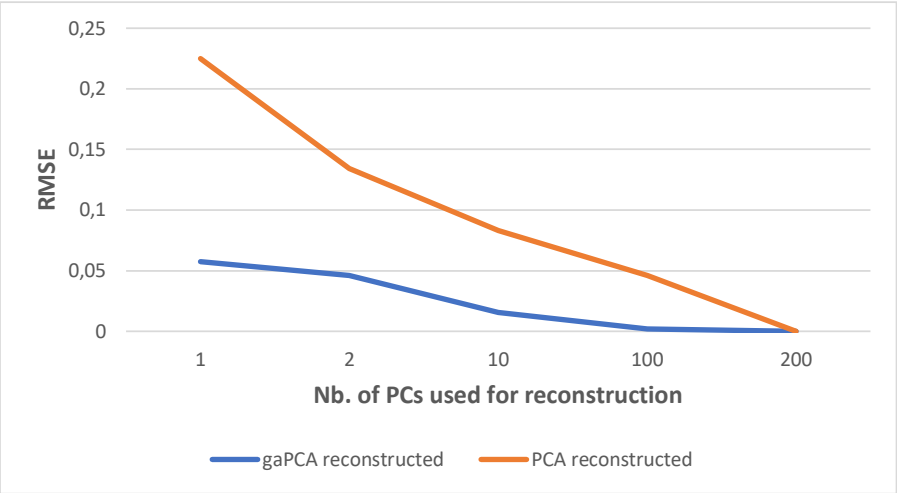
Tabelul 3.6 și Figura 3.1(c) arată valorile PSNR între imaginea inițială și cea reconstruită folosind metoda PCA și respectiv, gaPCA.

Table 3.6: PSNR pentru setul de date Indian Pines.

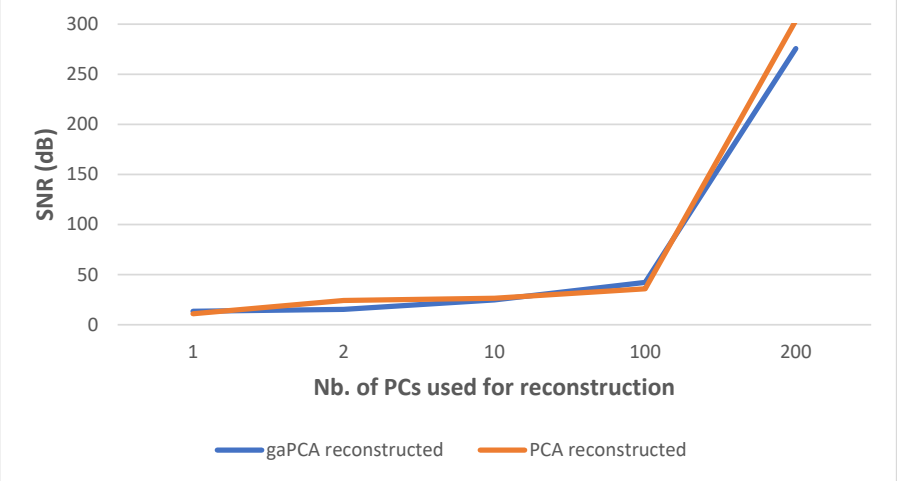
	1PC	**2PC**	**10PC**	**100PC**	**200PC**
gaPCA	24.87	26.79	36.24	53.60	287.06
PCA	22.37	35.73	37.86	47.26	315.03

Rezultatele pentru PSNR sunt similare cu cele obținute cu metrica SNR. Din nou, gaPCA arată rezultate mai bune atunci când este folosită o componentă principală, sau în cazul a 100 de componente principale, în timp ce PCA are un scor mai mare pentru 2 și 200 de componente.

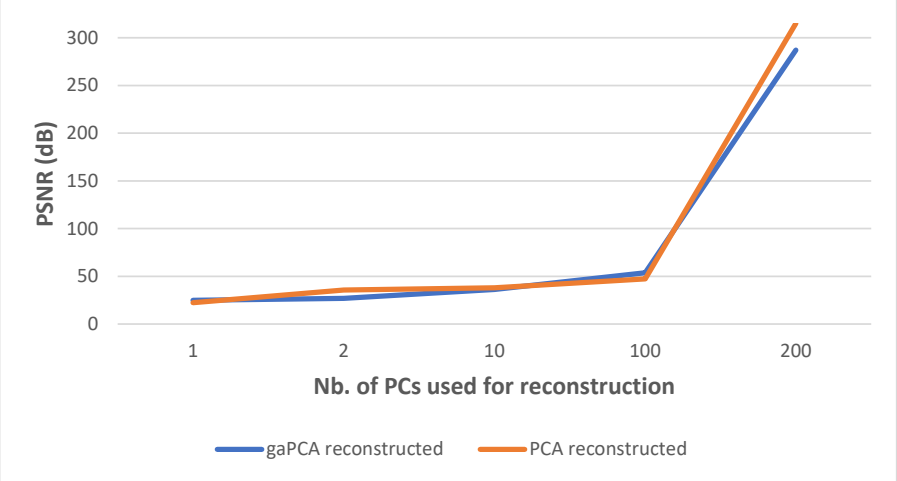
Mai mult, calitatea reconstrucției pentru toate seturile de date a fost evaluată, folosind un număr fix de componente principale, același număr care a fost utilizat pentru clasificare. Toate seturile de date menționate anterior au fost utilizate pentru efectuarea analizei comparative. Numărul de componente principale utilizate pentru reconstrucție a fost același număr de componente principale care au fost utilizate pentru clasificare: 10 pentru Indian Pines, 4 pentru Pavia University, 3 pentru DC Mall și 3 pentru AHS.



(a) RMSE



(b) SNR



(c) PSNR

Figure 3.1: RMSE (a), SNR (b) și PSNR (c) între imaginea originală și imaginea reconstruită cu PCA, respectiv gaPCA.

Valorile SNR calculate între imaginea originală și imaginea reconstruită din componentele principale PCA sau gaPCA pentru toate cele patru seturi de date sunt redate în Tabelul 3.7 și în Figura 3.2.

Table 3.7: SNR pentru toate seturile de date.

	PaviaU	**Indian Pines**	**DC Mall**	**AHS**
gaPCA	24.84	12.75	14.66	26.32
PCA	26.46	15.13	9.38	32.03

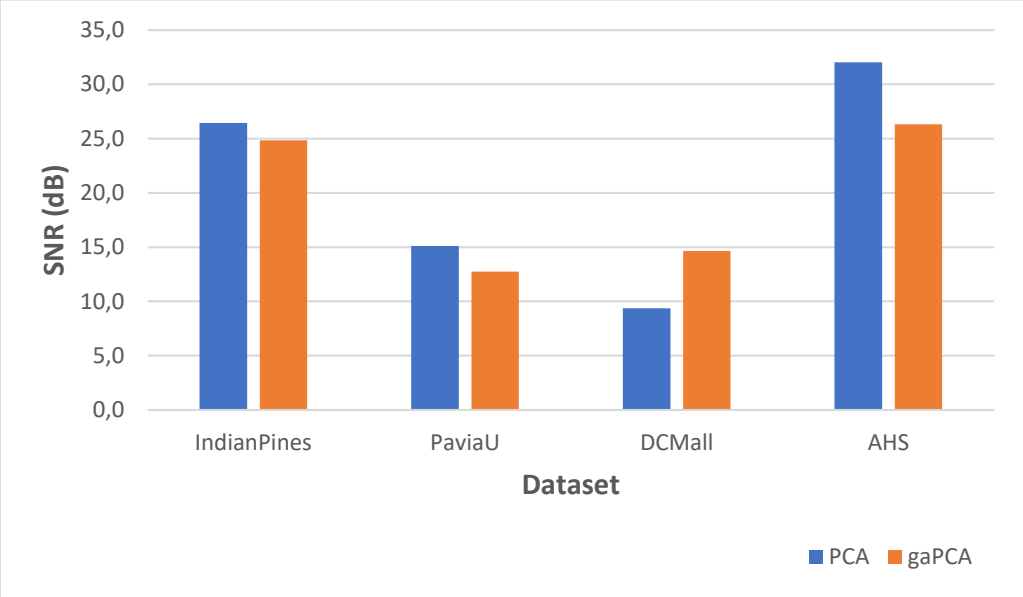


Figure 3.2: SNR între imaginea originală și imaginea reconstruită folosind PCA respectiv gaPCA pentru toate seturile de date.

Rezultatele arată valori similare pentru PCA și gaPCA în termeni de SNR pentru toate seturile de date. Deși diferențele nu sunt semnificative, pentru trei seturi de date, PCA are rezultate mai bune, în timp ce pentru setul de date rămas, gaPCA înregistrează rezultate mai bune.

Valorile SAM (în radiani) calculate între spectrele imaginii originale și cele ale imaginii reconstruite din componentele principale PCA sau gaPCA sunt redate în Tabelul 3.8 și în Figura 3.3. Numărul componentelor principale utilizate pentru reconstrucție a fost același ca cel utilizat pentru clasificare.

Table 3.8: SAM pentru toate seturile de date.

	PaviaU	**Indian Pines**	**DC Mall**	**AHS**
gaPCA	0.03	0.18	0.08	0.04
PCA	0.02	0.14	0.03	0.02

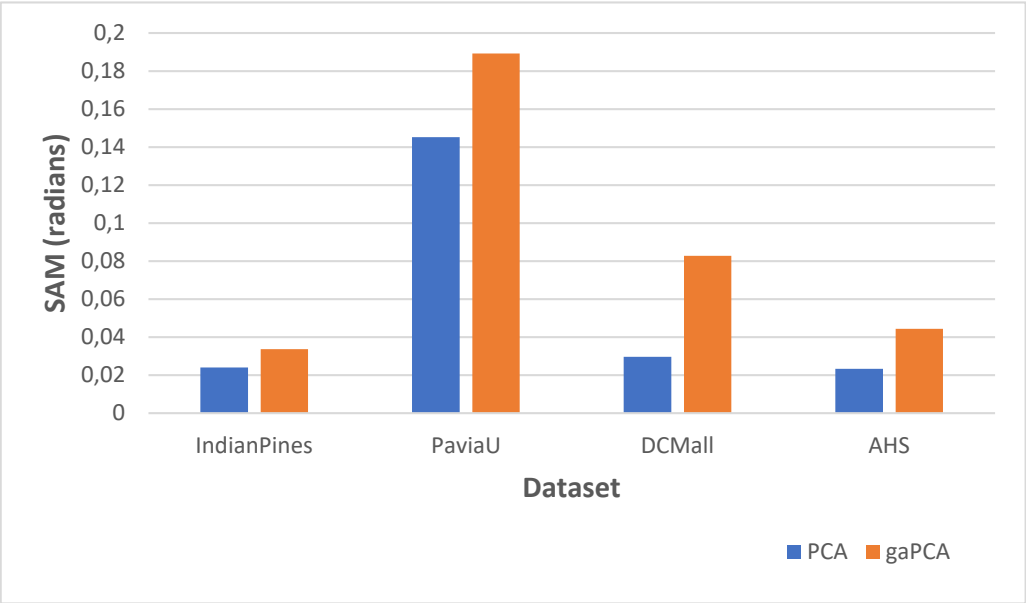


Figure 3.3: SAM între imaginea originală și cea reconstruită folosind PCA respectiv gaPCA pentru toate seturile de date.

Valorile SAM între spectrele reconstruite și cele originale sunt similare pentru ambele metode. Deși PCA are un scor ușor mai bun în ceea ce privește SAM, diferențele sunt foarte mici.

CC între imaginea originală și imaginea reconstruită folosind PCA sau gaPCA este prezentat în Tabelul 3.9 și în Figura 3.4.

Table 3.9: CC pentru toate seturile de date.

	PaviaU	**Indian Pines**	**DC Mall**	**AHS**
gaPCA	0.998	0.951	0.994	0.944
PCA	0.999	0.997	0.999	0.923

După cum arată rezultatele, pentru majoritatea seturilor de date, CC-ul gaPCA și PCA se află în aceeași gamă de valori.

Aceste numere confirmă că atât gaPCA, cât și PCA au obținut rezultate similare în ceea ce privește valorile de reconstrucție SNR, PSNR, RMSE, SAM și CC, fapt confirmat și de forma aproape identică a pantei în ambele cazuri și de comportamentul dictat de numărul de componente principale implicate în reconstrucție.

Mai mult decât atât, rezultatele evidențiază faptul că gaPCA depășește PCA atunci când se efectuează reconstrucția doar cu primele componente principale, în timp ce PCA produce rezultate mai bune atunci când sunt utilizate toate componentele principale.

3.2.3 Metrici privind redundanța componentelor principale

Metrica Mutual Information atât pentru componentele principale PCA, cât și pentru cele gaPCA este ilustrată în Figura 3.5 ca o matrice care ilustrează MI pentru fiecare pereche de

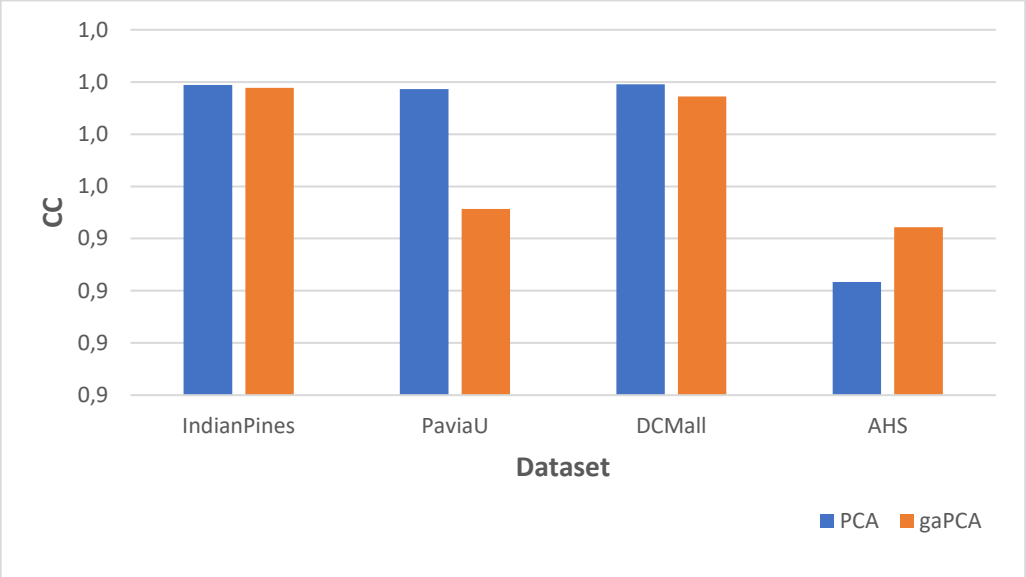
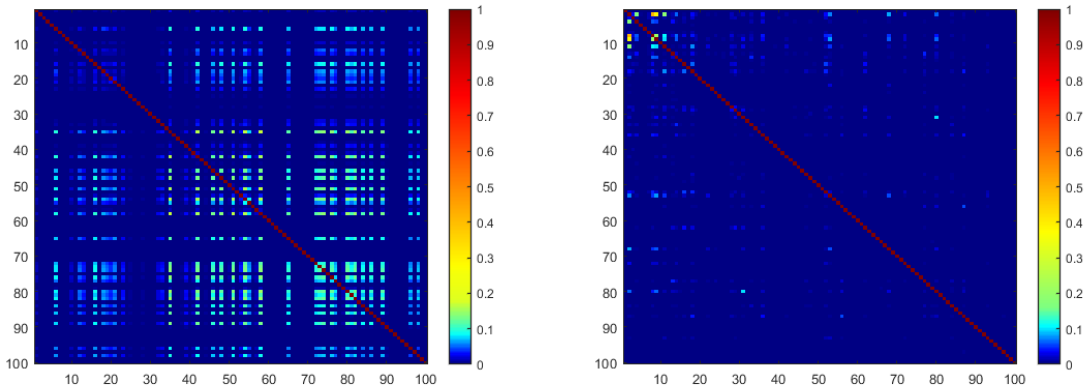


Figure 3.4: CC între imaginea originală și cea reconstruită folosind PCA sau gaPCA pentru toate seturile de date.

componente calculate cu cele două metode din setul de date Indian Pines. Figura evidențiază valori crescute ale MI între componentele PCA (patch-uri galbene și portocalii) în comparație cu cele gaPCA, indicând faptul că informație redundantă este partajată de componentele PCA și, în consecință, sunt mai puține informații noi sunt încorporate în acestea, ceea ce are și un impact asupra performanței clasificării, subiect care va fi abordat în detaliu în secțiunile următoare.

Deoarece componentele gaPCA nu sunt ordonate după nicio criteriu, ca în cazul PCA standard, se poate observa un anumit grad de redundanță pentru primele componente, cu toate acestea, reprezentarea vizuală a matricei MI evidențiază că mai multă informație nouă este conținută de componentele gaPCA, comparativ cu cele PCA.



(a) PCA MI

(b) gaPCA MI

Figure 3.5: MI pentru (a) PCA respectiv (b) gaPCA pentru setul de date Indian Pines.

3.2.4 Clasificarea imaginilor hiperspectrale

Ca metodă bine stabilită de reducere a dimensionalității, PCA este utilizată pe scară largă ca o rutină de pre-procesare în diferite aplicații de Remote Sensing. Prin urmare, majoritatea cercetărilor în acest domeniu au studiat utilizarea PCA pentru obținerea unei clasificări eficiente a imaginilor [19] [52], recunoașterea caracteristicilor [14] și identificarea zonelor de schimbare cu imagini multitemporale (detectarea modificărilor) [57], dar și vizualizarea imaginilor [40] și compresie [16].

Având în vedere gama largă de aplicații a PCA (și a metodelor bazate pe PCA) în domeniul Remote Sensing pentru reducerea dimensionalității, eliminarea datelor redundante, extragerea informațiilor privind acoperirea terestră sau extragerea caracteristicilor [42], în această secțiune, metoda PCA canonică a fost selectată ca referință pentru compararea și evaluarea comparativă a metodei gaPCA în domeniul clasificării terestre.

Pentru a asigura coerența metodologiei de cercetare, a fost ales numărul de componente principale calculate pentru fiecare set de date pentru a asigura cantitatea maximă de varianță explicată (98-99%) cu cel mai mic număr posibil de componente principale. Prin aplicarea acestui principiu, au fost calculate următorul număr de componente în fiecare caz: 10 - Indian Pines, 4 - Pavia University, 3 - DC Mall și 3 - AHS.

După calcularea componentelor principale folosind atât metoda gaPCA cât și PCA, clasificarea a fost realizată pe imaginile componentelor principale pentru toate cele 4 seturi de date, iar performanța în ceea ce privește acuratețea clasificării metodei gaPCA a fost evaluată comparativ cu valorile obținute de PCA.

Pentru această cercetare, s-au utilizat mai multe metode de clasificare: Maximum Likelihood, Support Vector Machine și Neural Network.

Rezultatele clasificării au fost validate cantitativ folosind două valori: Overall Accuracy (OA) și coeficientul Kappa (k). Ipoteza testată este aceea că gaPCA produce rezultate superioare de clasificare în comparație cu PCA, deoarece păstrează un grad mai mare de informație spectrală decât PCA, nefiind axată pe maximizarea variației datelor.

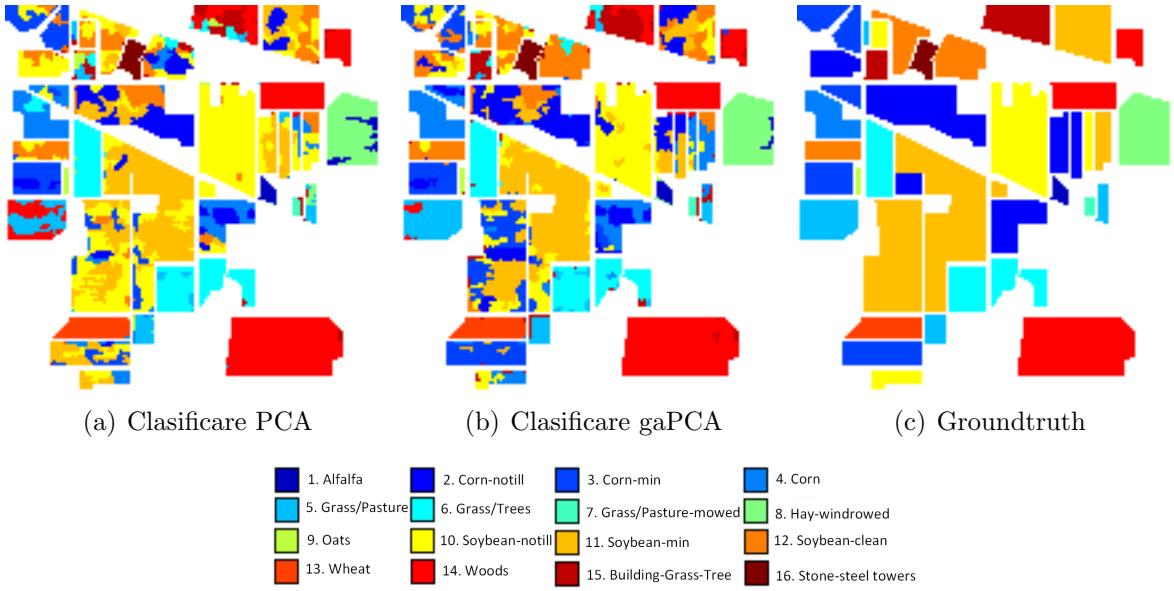
Fiecare dintre cele două metode au fost folosite pentru a produce un număr predefinit de componente principale care vor reprezenta benzile noii imagini multidimensionale folosite pentru clasificare. Aceiași algoritmi de clasificare, Maximum Likelihood (ML) și Support Vector Machine (SVM) au fost folosiți pentru toate seturile de date și pentru imaginile obținute cu ambele metode. Acuratețea clasificării în fiecare caz a fost evaluată prin generarea aleatorie a unui set de pixeli și efectuarea unei inspecții vizuale a rezultatului clasificării compartiv cu imaginea de referință a setului de date.

Pentru a evalua semnificația statistică a rezultatelor clasificării date de cele două metode, testul McNemar a fost efectuat pentru fiecare clasificator (ML și SVM).

3.2.4.1 Indian Pines

Efectuarea clasificării suprafețelor terestre pe setul de date Indian Pines este o sarcină dificilă, având în vedere numărul considerabil de clase existente în scenă, rezoluția spațială de 20 m (care

este una moderată), dar mai ales datorită asemănării spectrale ridicate dintre clasele existente. Această similaritate ridicată este cauzată de configurația scenei în momentul achiziției, soia și porumbul (cele două culturi principale din scenă) fiind într-o etapă de creștere destul de timpurie în acel moment. Figurile 3.6 (a) și (b) ilustrează rezultatele clasificării obținute cu metodele PCA și gaPCA, împreună cu imaginea de referință la momentul achiziției (c). În timp ce figura arată că există o abundență de pixeli cu conținut mixt care se reflectă în imagini clasificate destul de neomogene (pentru ambele metode), harta de clasificare a gaPCA este vizibil mai bună decât cea obținută prin metoda PCA clasică.



(a) Clasificare PCA (b) Clasificare gaPCA (c) Groundtruth

Figure 3.6: Imginile clasificate (cu Maximum Likelihood) folosind PCA (a) și gaPCA (b) vs. imaginea de referință (c) pentru Indian Pines.

Rezultatele acurateții clasificării sunt prezentate analitic în Tabelul 3.10, unde acuratețea ambelor metode este prezentată pentru fiecare clasă din scenă dar și per ansamblu, pentru ambele metode cu ambii algoritmi de clasificare (ML și SVM). Pentru testare, un set de 2000 de pixeli a fost generat aleator. Rezultatele arată că acuratețea generală este mai mare pentru gaPCA decât pentru PCA, ceea ce este reflectat și de valori gaPCA mai mari pentru majoritatea claselor.

Unul din motivele ce stă în spatele rezultatelor mai bune ale clasificării gaPCA este acela că, spre deosebire de metoda PCA, gaPCA nu se bazează pe ipoteza că acele caracteristici cu varianță ridicată din setul de date sunt responsabile de asigurarea unei discriminări eficiente între diferite clase, în timp ce caracteristicile cu varianță scăzută sunt desconsiderate ca fiind redundante. Această ipoteză se dovedește de multe ori greșită, mai ales în aplicațiile de clasificare a terenurilor unde scena este alcătuită din clase spectrale similare, cum este cazul Indian Pines. Aici se pot observa diferențe considerabile între scorurile obținute de cele două metode pentru seturi de clase similare; mai precis, gaPCA s-a dovedit a discrimina mai bine între subseturile similare (Corn, Corn notill, Corn mintill) și (Grass-pasture, Grass-pasture mowed) decât PCA, confirmând astfel capacitatea crescută de a distinge între clase cu semnături spectrale similare.

Table 3.10: Rezultatele clasificării pentru Indian Pines.

Clasa	Pixeli de training	PCA ML	gaPCA ML	PCA SVM	gaPCA SVM
Alfalfa	32	98.7	80.5	18.2	18.2
Corn notill	1145	30.6	47.6	65.2	69.3
Corn mintill	595	51.6	69.2	34.9	46.1
Corn	167	84.9	100	31.4	37.7
Grass pasture	328	55.7	80.5	64.6	71.9
Grass trees	463	96.1	90.6	91.2	92.5
Grass pasture mowed	19	68.3	71.7	60	60
Hay windrowed	528	88.5	96.7	99.5	99.6
Oats	20	100	96.9	15.6	6.3
Soybean notill	681	83.7	77.1	40.9	56.1
Soybean mintill	1831	46.6	47.7	79.4	78.3
Soybean clean	457	36.9	77.7	11.8	36.1
Wheat	150	97.2	97	91.1	93.1
Woods	884	98.7	96.9	97.3	97.3
Buildings Drives	263	33.7	61.4	45.5	52.1
Stone Steel Towers	103	100	100	95.5	97.2
z_{ML}=25.1 (signif=yes)	**OA(%)**	**62.1**	**70.2**	**67.2**	**72.1**
z_{SVM}=24.8 (signif=yes)	**Kappa**	**0.57**	**0.67**	**0.62**	**0.68**

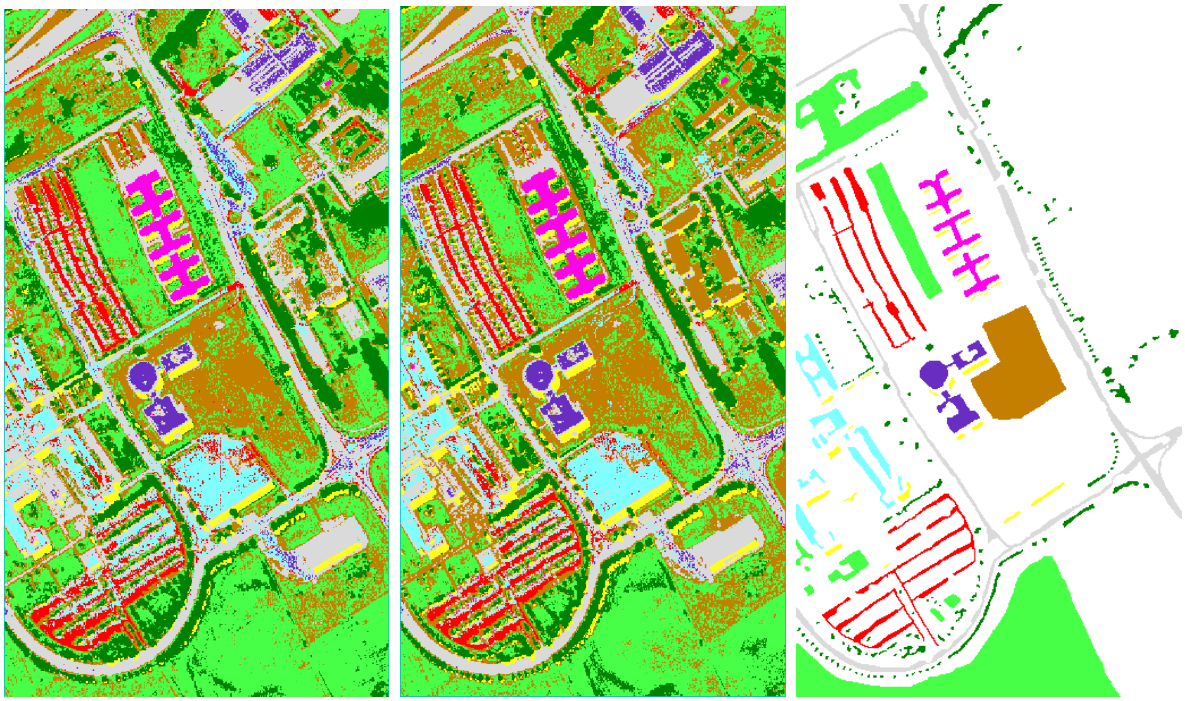
3.2.4.2 Pavia University

Pentru setul de date Pavia University, rezultatele clasificării sunt prezentate în Figura 3.7 pe baza imaginilor componente principale calculate cu PCA (a) și respectiv gaPCA (b) folosind algoritmul ML, în comparație cu imaginea de referință (c).

Rezultatele complete de clasificare (inclusiv acuratețea clasificării pentru fiecare clasă și per ansamblu), obținute pe baza unui set de 1000 de pixeli generați aleator, folosind ambii algoritmi de clasificare (ML și SVM) sunt prezentate în Tabelul 3.11. Aceste numere arată că, la fel ca în cazul setului de date Indian Pines, gaPCA a depășit PCA-ul canonic în acuratețea generală și în acuratețea de clasificare pentru majoritatea claselor.

Se poate observa din nou că gaPCA are o performanță superioară (o acuratețe mai mare a clasificării) decât PCA în cazul claselor compuse din structuri mici și forme complexe, cum ar fi Asphalt sau Bricks. Acest lucru se explică prin atenția mai mare acordată de metoda gaPCA obiectelor și claselor spectrale mai mici, având astfel un numărul de predicții false mai mic pentru astfel de cazuri, comparativ cu PCA. De exemplu, matricea de confuzie evidențiază o interpretare greșită în cazul metodei PCA, pentru clasele Bricks(cărămizi) confundate cu Gravel (pietriș), Asphalt (asfalt) confundat cu Bitumen (bitum).

Astfel de confuzii sunt cauzate de similitudinea spectrală crescută între clasele respective și nu datorită apropierii spațiale, așa cum se evidențiază în Tabelul 3.12. Cifrele susțin presupunerea că gaPCA are o capacitate sporită de a discrimina între clase spectrale similare, deoarece, spre deosebire de PCA, nu este axat preponderent spre clase având o contribuție dominantă la varianța totală.



(a) Clasificare PCA (b) Clasificare gaPCA (c) Groundtruth

Figure 3.7: Imaginile clasificate (cu Maximum Likelihood) PCA (a) și gaPCA (b) vs. imaginea de referință (c) pentru setul de date Pavia University.

Table 3.11: Rezultatele clasificării pentru setul de date Pavia University.

Clasa	Pixeli de training	PCA ML	gaPCA ML	PCA SVM	gaPCA SVM
Asphalt (grey)	1766	60.5	61.5	67.2	78.3
Meadows (light green)	2535	68.3	80	65	86.9
Gravel (cyan)	923	100	100	33.3	40
Trees (dark green)	599	88.2	89.7	100	67.7
Metal sheets (magenta)	872	100	100	100	100
Bare soil (brown)	1579	77.8	79.4	53.2	68.3
Bitumen (purple)	565	89.7	89.7	89.7	55.2
Bricks (red)	1474	68.3	72	81.7	86.6
Shadows (yellow)	876	100	100	100	100
z_{ML}=4.87 (signif=yes)	**OA(%)**	**72.2**	**78**	**69**	**78**
z_{SVM}=5.97 (signif=yes)	**Kappa**	**0.65**	**0.72**	**0.61**	**0.72**

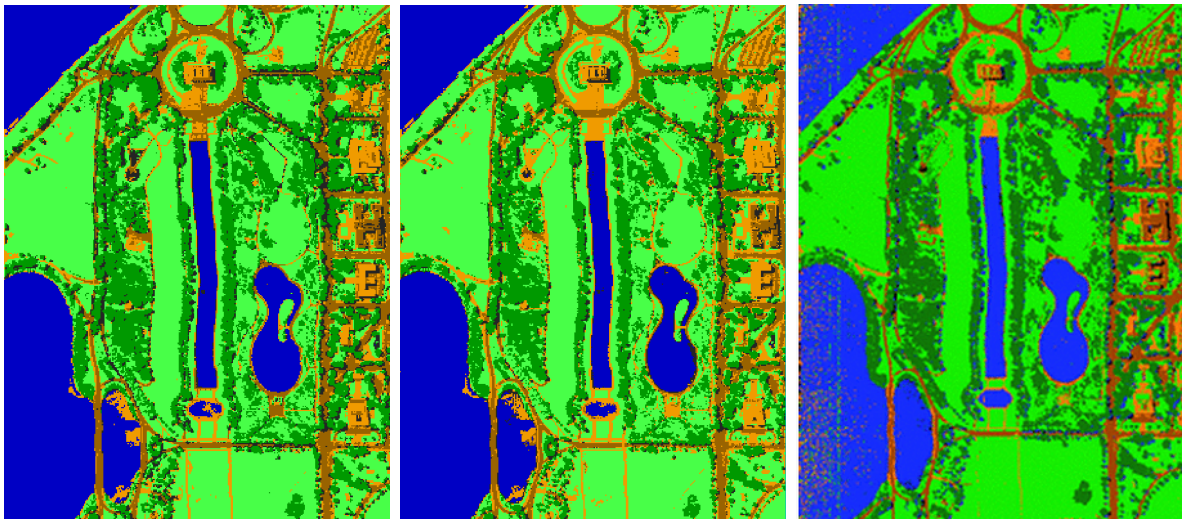
Table 3.12: Matricea de confuzie pentru setul de date Pavia University.

Class	True	False
Asphalt (PCA)	60.5 Asphalt	29.5 Bitumen
Asphalt (gaPCA)	61.5 Asphalt	21.8 Bitumen
Meadows (PCA)	68.3 Meadows	25.8 Bare soil
Meadows (gaPCA)	80 Meadows	17.6 Bare soil
Bricks (PCA)	68.3 Bricks	25.6 Gravel
Bricks (gaPCA)	72 Bricks	24.3 Gravel

Având în vedere aceste rezultate, gaPCA dovedește a avea rezulate mai bune în clasificarea obiectelor sau a claselor spectrale mai mici, consolidând presupunerea că are o capacitate sporită în păstrarea informațiilor legate de varianța semnalelor mai mici.

3.2.4.3 DC Mall

Rezultatele clasificării pentru setul de date DC Mall sunt prezentate în Figura 3.8 pe baza imaginilor componente principale calculate cu PCA (a) și gaPCA (b) folosind algoritmul ML, comparativ cu imaginea de referință a scenei DC Mall (c).



(a) Clasificare PCA (b) Clasificare gaPCA (c) Groundtruth

Figure 3.8: Imaginile clasificate (folosind Maximum Likelihood) PCA (a) și gaPCA (b) vs. imaginea de referință pentru setul de date DC Mall.

O imagine completă a rezultatelor clasificării rezultă în termeni de acuratețe generală și acuratețe pentru fiecare clasă, pentru algoritmii de clasificare ML și SVM, atât pentru metoda gaPCA cât și pentru PCA este ilustrată în Tabelul 3.13. Pentru evaluarea clasificării, a fost utilizat un set de 140 de pixeli generați aleator, iar rezultatele arată că gaPCA are o acuratețe mai mare decât PCA (atât acuratețe generală, cât și coeficientul kappa). Metoda gaPCA are

Table 3.13: Rezulatatele clasificării pentru setul de date DC Mall.

Clasa	Pixeli de training	PCA ML	gaPCA ML	PCA SVM	gaPCA SVM
Road (dark brown)	862	90	100	100	100
Trees (dark green)	413	75.9	82.7	75.9	75.9
Water (blue)	466	86.7	83.3	86.7	86.7
Grass (light green)	992	86.9	91.3	67.4	71.7
Shadows (black)	121	87.5	75	37.5	50
Roofs&paths(brown)	358	64.7	94.1	52.9	52.9
z_{ML}=2 (signif=yes)	**OA(%)**	**82**	**88**	**72**	**74**
z_{SVM}=1.13 (signif=no)	**Kappa**	**0.77**	**0.85**	**0.65**	**0.67**

performanță la fel ca în cazul celorlalte seturi de date, obținând valori superioare în comparație cu PCA în cazul structurilor mici cu forme complexe, cum ar fi clasa Roofs&paths (caz în care depășește PCA cu peste 30%). O altă clasă preponderent spectrală pentru care gaPCA obține o acuratețe de clasificare mai bună decât PCA este Trees, confirmând capacitatea sa mai mare de a păstra informații asociate acestei clase specifice. În final, gaPCA depășește PCA în ceea ce privește acuratețea generală cu peste 5%. Rezultatele clasificării cu metoda PCA (a), respectiv gaPCA (b) folosind algoritmul ML, în comparație cu imaginea de referință (c) pentru setul de date DC Mall sunt prezentate în Figura 3.8.

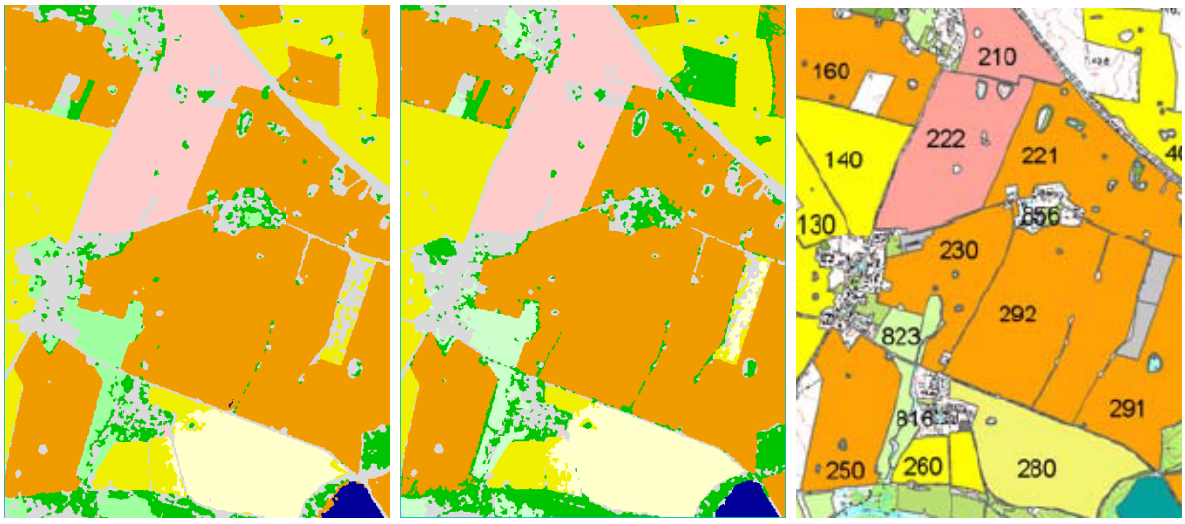
3.2.4.4 AHS

În cazul setului de date AHS, rezultatele clasificării obținute pe baza metodelor PCA și gaPCA au rezultat regiuni relativ omogene, așa cum arată imaginile de clasificare prezentate în Figura 3.9.

Rezultatele complete ale clasificării, atât în ceea ce privește acuratețea fiecărei clase, cât și acuratețea generală, pentru cele două metode, utilizând algoritmii de clasificare ML și SVM sunt prezentate în Tabelul 3.14. Aceste numere, obținute după evaluarea rezultatelor față de imaginea de referință, pe baza unui set de 100 de pixeli generați aleator, arată o acuratețe de clasificare mai mare pentru metoda gaPCA pentru majoritatea claselor din scenă.

Aceste rezultate evidențiază, de asemenea, acuratețea de clasificare pentru ambele metode în ceea ce privește fiecare clasă. Ca atare, se poate observa că, în cazul claselor celor mai extensiv reprezentate (de exemplu, oil seed rape, corn, set aside: oil seed rape), atât gaPCA, cât și PCA standard raportează rezultate similare. O acuratețe puțin mai bună este obținută de PCA în cazul clasei de winter wheat, cu toate acestea, atunci când se trece la clase preponderent spectrale, cum ar fi grassland sau cutting pasture, gaPCA depășește clar PCA. Clasa Urban este cea mai dificil de clasificat datorită naturii specifice pentru această scenă particulară (formată din structuri mixte, cum ar fi clădiri, drumuri locale și vegetația adiacentă din mediul rural). În general, rezultatele confirmă capacitatea gaPCA de a clasifica corect clase spectrale mai mici sau clase cu pixeli similari sau având o compoziție mixtă.

Testul McNemar (scorul z) a fost calculat pe rezultatele clasificării pentru toate seturile de date și confirmă (cu o excepție izolată) că precizia de clasificare gaPCA mai mare în comparație



(a) Clasificare PCA (b) Clasificare gaPCA (c) Groundtruth

Figure 3.9: Imaginile clasificate (folosind Maximum Likelihood) PCA (a) și gaPCA (b) vs. imaginea de referință pentru setul de date AHS.

Table 3.14: Rezultatele clasificării pentru setul de date AHS.

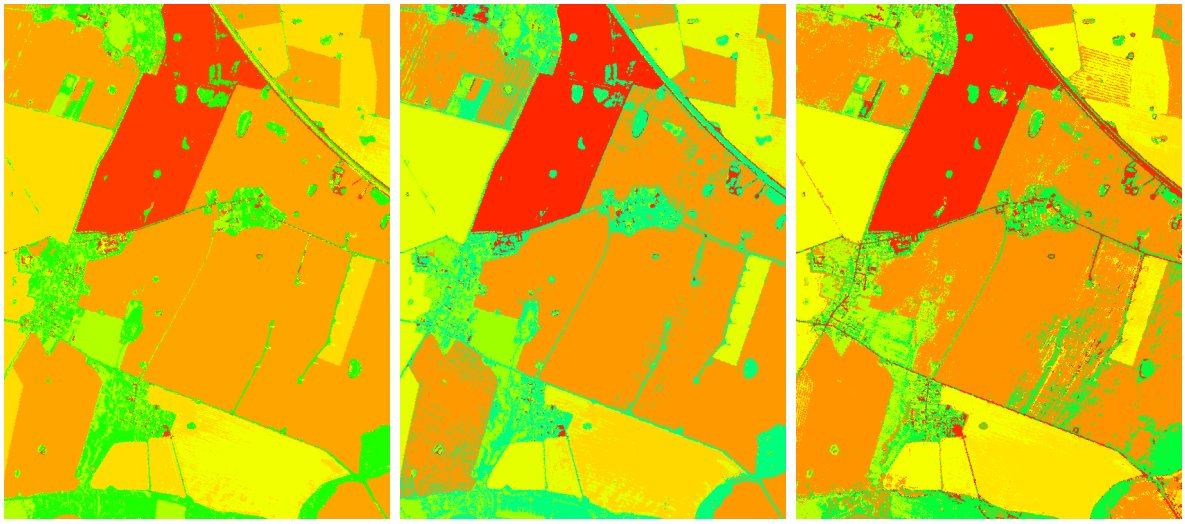
Clasa	Pixeli de training	PCA ML	gaPCA ML	PCA SVM	gaPCA SVM
Rape (dark yellow)	2786	93.3	93.3	93.2	97.7
Rape (light yellow)	1013	80	80	90	95
Maize (pink)	969	100	100	100	100
Winter wheat (orange)	4429	100	98.1	97.3	97.3
Pasture (light green)	1788	66.7	66.7	84.6	92.3
Grassland (dark green)	1242	60	80	52.4	95.2
Urban (grey)	1079	60	90	64	92
z_{ML}=1.97 (signif=yes)	**OA**	**90.6**	**93.8**	**90**	**96.6**
z_{SVM}=3.92 (signif=yes)	**Kappa**	**0.86**	**0.91**	**0.86**	**0.95**

cu PCA canonic este semnificativă statistic.

Pentru obținerea acestor rezultate, toate experimentele au fost efectuate în Matlab R2018b și ENVI 5.5, rulând pe un procesor Intel (R) Xeon (R) X3440 cu 2,53 GHz și 8 GB memorie instalată.

3.2.4.5 Compararea gaPCA cu alte metode

Un experiment suplimentar a fost efectuat pe setul de date AHS, urmărind compararea rezultatelor clasificării prin intermediul rețelelor neuronale (folosind Neumapper) a metodelor PCA standard, gaPCA și nonlinear PCA.



(a) PCA (b) gaPCA (c) NLPCA

Figure 3.10: Imaginile clasificate (folosind rețele neurale) PCA (a) gaPCA (b) și NLPCA pentru setul de date AHS.

Este interesant de remarcat asemănările ridicate între cele trei metode pentru majoriatatea claselor scenei (oil seed rape, maize, etc.). Diferențe mici apar la clasele winter wheat, pasture, cutting pasture, în timp ce clasa urban pare a fi cea mai dificil de clasificat și datorită specificului acestei clase care cuprinde un amestec de clădiri, drumuri de țară și vegetație într-o zonă rurală. Scorurile gaPCA în termeni de clasificare sunt comparabile cu cele obținute de PCA standard și PCA neliniar.

3.3 Concluzii

În acest capitol, a fost prezentată o metodă originală bazată pe PCA, intitulată "Geometric Approximated Principal Component Analysis" (sau gaPCA), iar performanța acesteia a fost ilustrată în aplicațiile de Remote Sensing, care implică clasificarea terestră realizată pe imagini hiperspectrale multidimensionale. Aplicațiile descrise au implicat aplicarea gaPCA pe patru seturi de date hiperspectrale pentru reducerea dimensionalității și evaluarea performanței și

a rezultatelor sale atât calitativ (prin calcularea metricilor de evaluare a calității imaginilor componente principale obținute folosind gaPCA), cât și cantitativ (evaluarea exactității sarcinii de clasificare a terenurilor) pentru fiecare set de date. Evaluarea metodei gaPCA a fost efectuată prin calcularea mai multor valori obiective și folosind PCA ca referință.

În ceea ce privește performanțele obținute de gaPCA în activitatea de de clasificare pentru cele 4 seturi de date, această nouă metodă a depășit PCA în fiecare caz în ceea ce privește acuratețea generală calculată pentru fiecare set de date și a obținut, de asemenea, un nivel mai mare decât PCA pentru majoritatea claselor. O analiză detaliată a rezultatelor în ceea ce privește acuratețea fiecărei clase a confirmat ipoteza inițială potrivit căreia gaPCA, spre deosebire de metoda PCA standard, ia în considerare informația cu o contribuție mai mică la varianța totală a semnalului, care este considerată redundantă sau lipsită de importanță de către PCA standard. Prin urmare, gaPCA are o capacitate superioară de a discrimina obiecte mici sau clase similare, caracteristică confirmată de performanța sa în experimentele pentru clasele preponderent spectrale în fiecare set de date, unde a depășit PCA standard.

În concluzie, rezultatele experimentale și analiza descrisă în acest capitol au arătat că noua metodă gaPCA este mai potrivită (decât PCA-ul canonic) în sarcinile de clasificare din domeniul Remote Sensing, care implică imagini hiperspectrale cu structuri mici sau obiecte reduse ca dimensiune, sau unde sunt preponderente clase spectrale sau clase similare spectral.

Cercetările și experimentele prezentate în acest capitol au fost validate și diseminate în următoarele publicații:

- [45] A. L. Machidon, F. Del Frate, M. Picchiani, O. M. Machidon, and P. L. Ogrutan. Geometrical Approximated Principal Component Analysis for Hyperspectral Image Analysis. *Remote Sensing*, 12(11), 2020
- [44] A. L. Machidon, R. Coliban, O. Machidon, and M. Ivanovici. Maximum Distance-based PCA Approximation for Hyperspectral Image Analysis and Visualization. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018 also indexed in IEEE Xplore Digital Library
- [46] A. L. Machidon, M. Ivanovici, R. Coliban, and F. Del Frate. A Geometrical Approximation of PCA for Hyperspectral Data Dimensionality Reduction. In *The ESA Earth Observation Phi-week EO Open Science and FutureEO*. ESA, 2018

CAPITOLUL 4

Recunoaștere facială

4.1 Reducerea dimensionalității pentru recunoașterea facială

Un alt domeniu de aplicații pentru metoda PCA este cel al Eigenfaces [67]. Utilizarea Analizei Componentelor Principale în aplicațiile de recunoaștere facială a fost susținută mai întâi de Kirby și Sirovich [36], care au ilustrat modul în care PCA poate fi aplicat pentru a forma un set de caracteristici de bază dintr-o colecție de imagini faciale. Eigenfaces sau PCA și-au demonstrat succesul în recunoașterea, detectarea și urmărirea fețelor [66]. În consecință, în acest capitol s-a aplicat gaPCA pentru recunoașterea facială și s-a comparat acuratețea recunoașterii cu cea obținută prin utilizarea PCA standard, pe patru baze de date diferite cu imagini faciale.

Pentru analiza eficienței algoritmilor gaPCA și standard PCA în domeniul recunoașterii fetei, au fost utilizate patru bine-cunoscute seturi de date open source cu imagini faciale: FEI [5], Yale [12], Cambridge [3] și Labeled Faces in the Wild (LFW) [30]. Pentru toate cele patru seturi de date s-au calculat fețele proprii (eigenfaces) folosind ambii algoritmi.

4.2 Metodologie

4.2.1 Seturile de date FEI, Yale and Cambridge

În această lucrare, procesul de recunoaștere facială a fost aplicat pe primele 3 seturi de date folosind atât metoda PCA cât și gaPCA. Fețele proprii rezultate în urma aplicării fiecărei metode au fost utilizate pentru a căuta, pentru toate fețele din fiecare set de test, cea mai asemănătoare față din setul de antrenament aferent.

Pentru calcularea scorului de similaritate, a fost utilizată o metrică bazată pe distanța euclidiană inversă.

Fețele proprii calculate cu cele două metode (gaPCA și PCA standard) au fost utilizate pentru a identifica, pentru fiecare față din setul de test, cea mai potrivită față din setul de antrenament, pe baza scorului de similaritate calculat. În cele din urmă, pentru ambele metode a fost calculat scorul general de acuratețe pentru fiecare set de date.

4.2.2 Setul de date LFW

Al patrulea experiment de recunoaștere facială folosind metoda gaPCA a fost efectuat pe setul de date Labeled Faces in the Wild (LFW) [30]. În acest experiment, a fost utilizat un sub-set al setului de date LFW, care cuprinde subiecții pentru care au fost disponibile cel puțin 100 de fețe, ceea ce duce la un total de 1140 fețe a 5 persoane. Acest subset a fost împărțit într-un set de antrenament care conține 70% din numărul total de imagini și un set de test cuprinzând restul de 30%. Atât algoritmul gaPCA cât și omologul canonic PCA au fost utilizate (separat) pentru a efectua reducerea dimensionalității pe setul de antrenament și pentru a calcula fețele proprii corespunzătoare. Setul de date de antrenament cu dimensiuni reduse rezultat (care cuprinde fețele proprii calculate cu fiecare metodă) a fost utilizat pentru a antrena un clasificator pe bază de de rețea neurală, un perceptron multistrat (MLP) cu un strat ascuns.

Acuratețea clasificării a fost evaluată pentru trei scenarii: instruirea MLP cu fețele proprii calculate de PCA-ul canonic, calculate prin metoda gaPCA, sau instruirea lui pe datele originale (date brute, fără aplicarea vreunei reduceri de dimensionalitate). Pe lângă acuratețea generală a clasificării, evaluarea a avut în vedere și numărul de iterații necesare în fiecare caz pentru ca clasificatorul MLP să fie instruit cu succes.

4.3 Rezultate și discuție

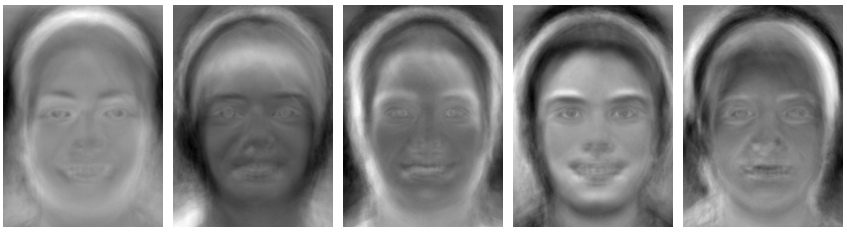
4.3.1 FEI

S-a efectuat o comparație între abordarea Eigenfaces standard, bazată pe metoda PCA și cea bazată pe metoda gaPCA, pe baza de date FEI [5] și s-a utilizat metrica bazată pe distanța euclidiană pentru calcularea acurateții recunoașterii în ambele cazuri.

Pentru realizarea recunoașterii faciale, baza de date a fost împărțită în 2 subseturi: un subset de antrenament care cuprinde aproximativ 85% din imagini și un subset de test cu restul de 15%. Pe subsetul de instruire, a fost aplicat atât algoritmul PCA standard, cât și algoritmul gaPCA. Figura 4.1 prezintă primele 10 eigenfaces obținute cu metoda PCA standard, în timp ce Figura 4.2 afișează eigenfaces corespunzătoare după implementarea gaPCA pe subsetul de instruire de 350 de imagini din baza de date FEI.

Graficul din Figura 4.3 prezintă variațele cumulate pentru primele 200 de componente principale. După cum era de așteptat, primele 100-150 componente principale cuprind 95-98% din varianța imaginilor din setul de date.

(a) 1st PCA eigenface (b) 2nd PCA eigenface (c) 3rd PCA eigenface (d) 4th PCA eigenface (e) 5th PCA eigenface



(f) 6th PCA eigenface (g) 7th PCA eigenface (h) 8th PCA eigenface (i) 9th PCA eigenface (j) 10th PCA eigenface

Figure 4.1: Primele zece eigenfaces din baza de date FEI obținute prin metoda PCA standard.

(a) 1st gaPCA eigenface (b) 2nd gaPCA eigenface (c) 3rd gaPCA eigenface (d) 4th gaPCA eigenface (e) 5th gaPCA eigenface

(f) 6th gaPCA eigenface (g) 7th gaPCA eigenface (h) 8th gaPCA eigenface (i) 9th gaPCA eigenface (j) 10th gaPCA eigenface

Figure 4.2: Primele zece eigenfaces din baza de date FEI obținute prin metoda gaPCA.

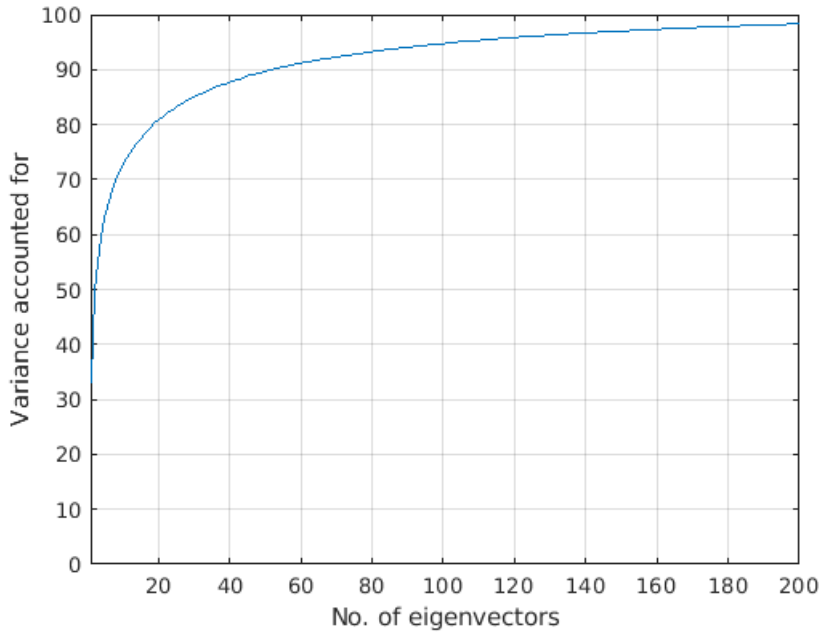


Figure 4.3: Procentul varianței explicate de vectorii proprii ai setului de date FEI.

În Figura 4.4 sunt ilustrate fețele medii obținute cu metoda PCA standard (a) și gaPCA (b) pe subsetul de instruire de 350 de imagini din baza de date FEI.

(a) Average face PCA (b) Average face gaPCA

Figure 4.4: Fața medie din baza de date FEI folosind metoda PCA (a) și metoda gaPCA (b).

Rezultatele recunoașterii faciale arată că pentru un procent relativ ridicat din imaginile testate, (98% pentru PCA standard și 92% pentru gaPCA în cazul a 150 de vectori proprii reținuți), imaginea feței oferită pe baza comparației scorului de similaritate a fost corectă (Tabelul 4.1).

O ușoară scădere a acestor valori este observată pentru ambele metode (PCA standard și gaPCA) dacă numărul componentelor reținute este redus Figura 4.5. Cu toate acestea, este interesant de observat că ambele metode prezintă aceeași creștere a acurateții recunoașterii de la 10 eigenvectori la 150 eigenvectori, respectiv o creștere de 14 procente, în ambele cazuri.

Analiza rezultatelor arată că se poate obține o performanță foarte bună (92% în acuratețe) cu modelul gaPCA. În cazul particular al bazei de date FEI, performanța gaPCA a fost evaluată în principal cu privire la factori precum expresia facială, coafura, podoabele (ochelari, cercei

Table 4.1: Acuratețea recunoașterii faciale pentru baza de date FEI folosind standard PCA și gaPCA.

Metoda	**Acuratețea recunoașterii faciale**
PCA	98%
gaPCA	92%

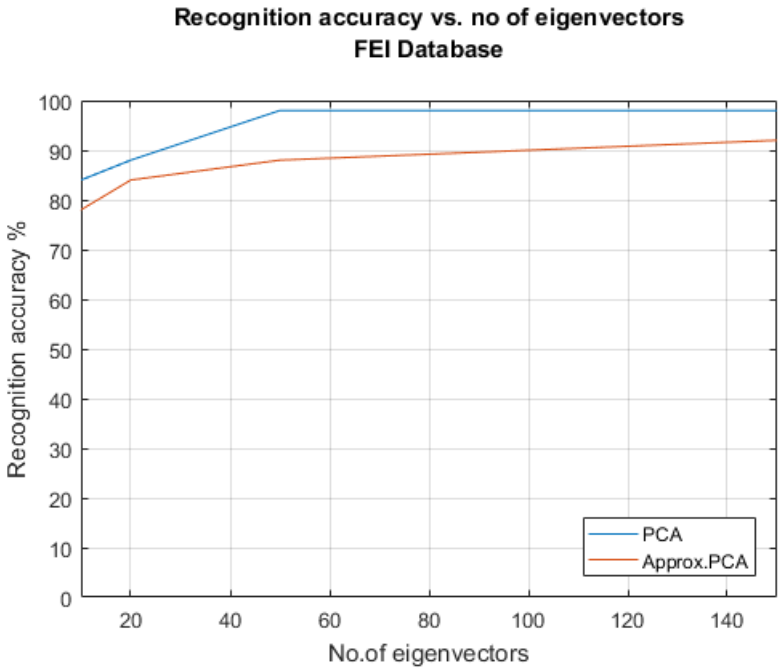
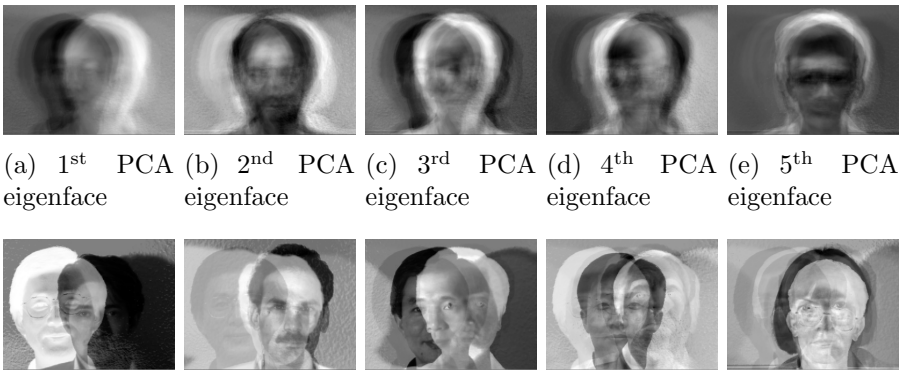


Figure 4.5: Acuratețea recunoașterii vs. nr. de eigenevectori pentru baza de date FEI.

etc.). Alți factori, cum ar fi iluminarea și variația pozelor, tind să fie mai puțin proeminenți în medii relativ controlate.

4.3.2 Yale

Pentru setul de date Yale, a fost utilizată aceeași metodologie: metoda gaPCA a fost validată comparativ în ceea ce privește performanțele sale de recunoaștere a feței, comparând rezultatele acurateții sale cu cele obținute de PCA. Ca atare, fețele proprii ale setului de date Yale au fost calculate folosind ambele metode. Pentru efectuarea evaluării acurateții de recunoaștere facială, setul de date a fost împărțit într-un set de antrenament (care conține 135 de imagini) și un set de test (format din restul de 30 de imagini). Având în vedere faptul că setul de date Yale este destul de mic, au fost selectați 20 de vectori proprii, atât în cazul gaPCA, cât și în cazul PCA. Primele 5 componente principale ale acestui set de date sunt ilustrate în Figura 4.6 pentru ambele metode (gaPCA – jos, PCA standard – sus), în timp ce rezultatele acurateții recunoașterii faciale sunt prezentate în Tabelul 4.2 .



(a) 1st PCA eigenface (b) 2nd PCA eigenface (c) 3rd PCA eigenface (d) 4th PCA eigenface (e) 5th PCA eigenface

(f) 1st gaPCA eigenface (g) 2nd gaPCA eigenface (h) 3rd gaPCA eigenface (i) 4th gaPCA eigenface (j) 5th gaPCA eigenface

Figure 4.6: Primele cinci fețe proprii (eigenfaces) din baza de date Yale obținute cu metoda PCA standard (sus) și metoda gaPCA (jos).

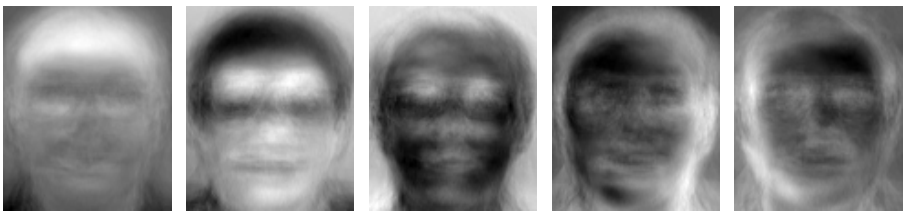
Table 4.2: Acuratețea recunoașterii faciale pentru baza de date Yale folosind standard PCA și gaPCA.

Metoda	**Acuratețea recunoașterii faciale**
PCA	76.66%
gaPCA	73.33%

Procentele sunt mai mici pentru ambele metode în cazul bazei de date Yale decât în cazul setului de date FEI, în principal datorită variației mari în poziționarea și iluminarea fețelor. Cu toate acestea, este interesant de subliniat faptul că acuratețea gaPCA este aproape aceeași cu cea obținută de PCA standard, în cazul a 20 de vectori proprii reținuți.

4.3.3 Cambridge

Figura 4.7 ilustrează primele 5 componente principale calculate pe setul de date Cambridge, folosind atât gaPCA (jos) cât și canonical PCA (sus).



(a) 1st PCA eigenface (b) 2nd PCA eigenface (c) 3rd PCA eigenface (d) 4th PCA eigenface (e) 5th PCA eigenface

(f) 1st gaPCA eigenface (g) 2nd gaPCA eigenface (h) 3rd gaPCA eigenface (i) 4th gaPCA eigenface (j) 5th gaPCA eigenface

Figure 4.7: Primele cinci fețe proprii (eigenfaces) din baza de date Cambridge obținute cu metoda PCA standard (sus) și metoda gaPCA (jos).

Table 4.3: Acuratețea recunoașterii faciale pentru baza de date Cambridge folosind PCA și gaPCA.

Metoda	**Acuratețea recunoașterii faciale**
PCA	94.14%
gaPCA	93.33%

Acuratețea de recunoaștere facială pe setul de date Cambridge atât pentru gaPCA, cât și pentru metoda PCA este prezentată în Tabelul 4.3. Setul de date Cambridge original a fost împărțit într-un set de antrenament, format din 280 de imagini și un set de test, care conține restul de 120 de imagini. Pe setul de antrenament, fețele proprii au fost calculate cu fiecare dintre cele două metode. Variația acurateții de recunoașterii în funcție de numărul de vectori proprii reținuți este ilustrată în Figura 4.8, care evidențiază o precizie relativ constantă pentru ambele metode, indiferent de numărul de vectori proprii utilizați.

4.3.4 LFW

Pentru setul de date LFW, Figura 4.9 ilustrează primele 5 eigenfaces calculate cu metoda PCA standard, în timp ce fețele proprii calculate cu algoritmul gaPCA sunt afișate în Figura

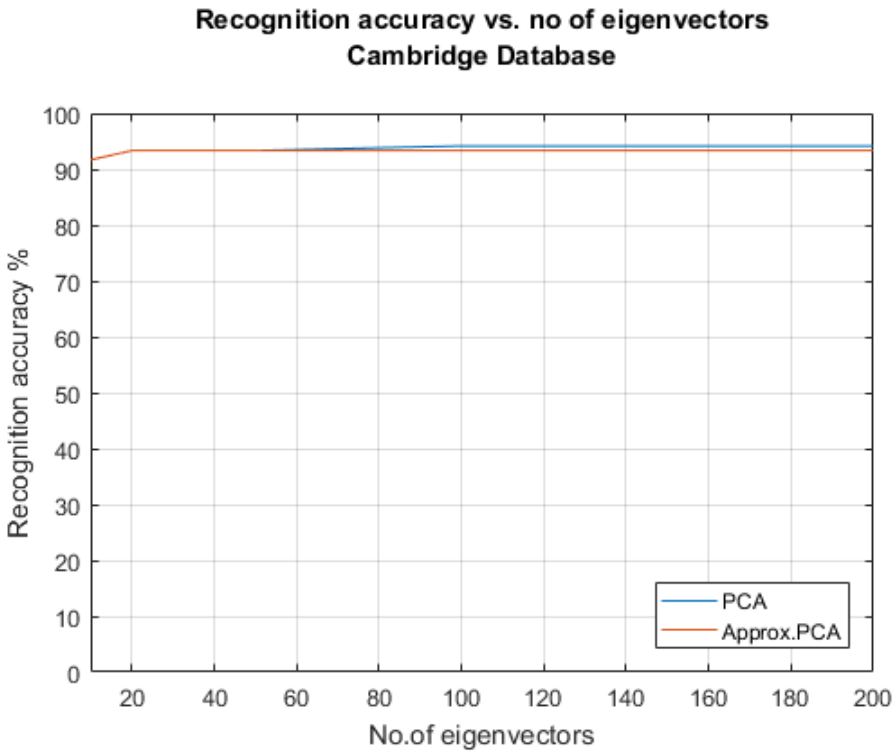


Figure 4.8: Acuratețea recunoașterii vs. nr. de eigenevectori pentru baza de date Cambridge.

4.10. Acuratețea clasificării pentru fiecare dintre cei 5 subiecți implicați în acest experiment este prezentată în Tabelul 4.4, pentru cele trei scenarii: instruirea clasificatorului MLP cu datele brute (originale) (NO PCA), cu datele obținute prin aplicarea PCA sau cu datele rezultate după utilizarea metodei gaPCA.

Table 4.4: Acuratețea medie pentru 10 clasificări cu NO PCA, PCA standard și gaPCA.

Personality	**NO PCA**	**PCA**	**gaPCA**
C. Powell	78%	83%	81%
D. Rumsfeld	56%	70%	72%
G.W. Bush	86%	88%	85%
G. Schroeder	53%	74%	68%
T. Blair	67%	71%	68%

Rezultatele evidențiază faptul că metoda gaPCA a obținut o acuratețe medie de 75%, foarte aproape de 77%, cazul PCA-ului canonic și ambele metode au depășit clar cazul în care nu a fost utilizată nicio metodă de reducere a dimensionalității asupra setului de date, caz în care acuratețea este de numai 68%. Mai mult decât atât, gaPCA și-a depășit omologul pentru clasa specifică "D.Rumsfeld ", pentru care a obținut un scor superior comparativ cu PCA. În ceea ce privește modul în care reducerea dimensionalității influențează procesul de instruire a clasificatorului, rezultatele din Tabelul 4.5 prezintă numărul de iterații necesare pentru instruire în fiecare dintre cele trei scenarii detaliate mai sus. Se poate observa cu ușurință că fie ca se utilizează PCA standard, fie gaPCA pentru a reduce dimensionalitatea setului de date de instruire, clasificatorul este instruit aproape de două ori mai rapid.

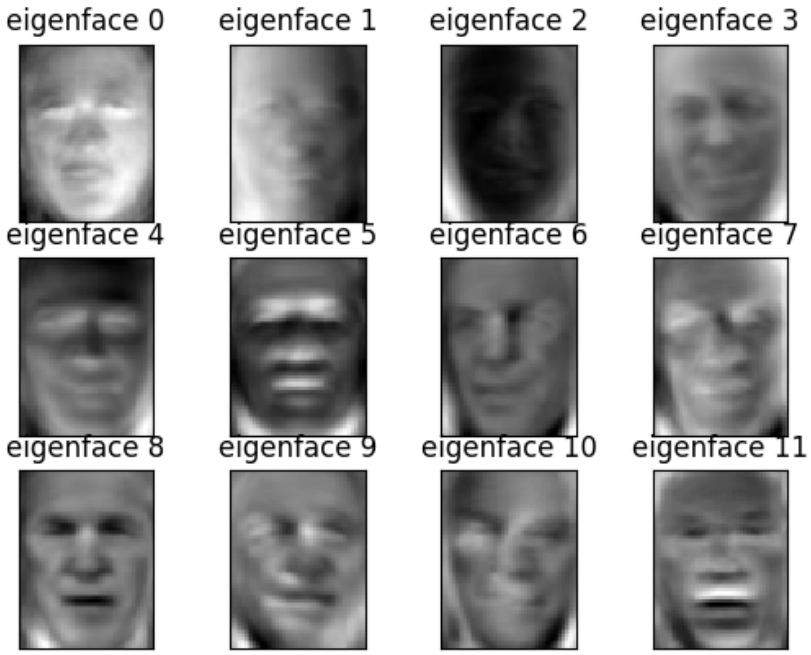


Figure 4.9: Fețele proprii ale bazei de date LFW folosind PCA standard.

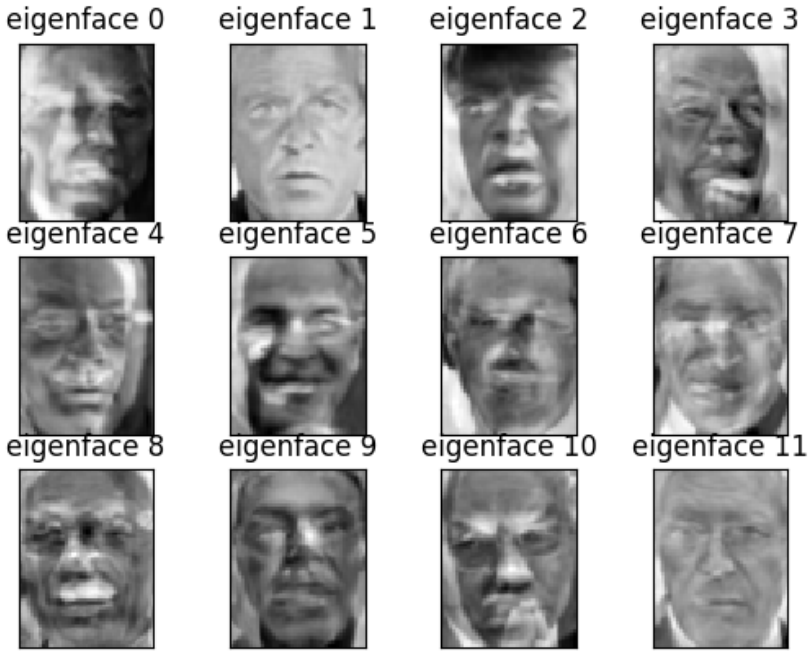


Figure 4.10: Fețele proprii ale bazei de date LFW folosind gaPCA.

Table 4.5: Numărul de iterații necesare clasificatorului folosind NO PCA, PCA standard și gaPCA

Reducerea dimensionalității	**Nr. iterații**
No PCA	48.4
PCA	19.2
gaPCA	25

Antrenarea clasificatorului cu datele originale, brute, necesită 48 iterații, comparativ cu doar 25 în cazul aplicării metodei gaPCA ca tehnică de reducere a dimensionalității și doar 19 în cazul PCA standard.

4.4 Concluzii

Acest capitol a prezentat evaluarea și validarea metodei gaPCA în domeniul recunoașterii faciale. Pentru experimentele prezentate în acest capitol, au fost utilizate patru seturi de date cu imagini faciale: FEI, Yale, Cambridge și LFW. Procesul de recunoaștere facială a fost realizat pentru primele trei seturi de date pe baza unui scor de similaritate bazat pe inversul distanței euclidiane. Acuratețea recunoașterii a fost calculată pentru ambele metode, iar evaluarea comparativă a ilustrat că gaPCA s-a comportat foarte similar cu omologul său mai consacrat, notând în unele cazuri rezultate mai bune sau egale, iar în alte cazuri doar ușor inferioare (sub 10%) comparativ cu PCA standard.

În cazul setului de date LFW, recunoașterea facială a fost efectuată folosind un clasificator cu o rețea neurală, instruit în trei scenarii: pe datele originale, brute, pe datele obținute aplicând PCA standard și pe datele rezultate din aplicarea gaPCA. Rezultatele acurateții au plasat gaPCA foarte aproape de PCA-ul canonic, cu doar 2% sub nivelul acurateții medii, pentru unele clase obținând rezultate superioare. În plus, eficacitatea gaPCA ca tehnică de reducere a dimensionalității a fost confirmată și cantitativ, arătând că aplicarea gaPCA pe setul de antrenament scade substanțial numărul de iterații de antrenament necesare pentru clasificatorul bazat pe rețela neurală.

Cercetările și experimentele prezentate în acest capitol au fost validate și diseminate în următoarea publicație:

- [48] A. L. Machidon, O. M. Machidon, and P. L. Ogrutan. Face Recognition Using Eigenfaces, Geometrical PCA Approximation and Neural Networks. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pages 80–83, 2019, indexat în Web of Science (Proceedings paper) și în IEEE Xplore Digital Library.

CAPITOLUL 5

Paralelizare

Progresele tehnologice recente atât în ceea ce privește senzorii hiperspectrali cât și în domeniul achiziției și procesării de date au crescut exponențial volumul datelor de Remote Sensing. Faptul că o cantitate semnificativă din aceste date este definită printr-un grad ridicat de redundanță, permite reducerea la un număr mult mai mic de variabile, fără pierderi substanțiale de informație. Acest lucru poate fi realizat folosind tehnicile de reducere a dimensionalității [63], metode și algoritmi matematici care transformă datele de dimensionalitate ridicată într-o "reprezentare semnificativă cu o dimensionalitate redusă" [64]. Totuși, aceste tehnici prezintă o complexitate computațională ridicată și necesită resurse de calcul considerabile, o problemă care poate afecta aplicațiile cu restricții de timp stricte; prin urmare, arhitecturile de tip High Performance Computing (HPC) au potențialul de a facilita implementarea acestor tehnici, eficiente din punctul de vedere al timpului de execuție și al resurselor de calcul.

5.1 Metode PCA paralelizate

Datorită faptului că timpul de execuție al metodelor de Projection Pursuit crește exponențial cu dimensionalitatea datelor, acești algoritmi tind să prezinte timpi ridicați de calcul. Acest lucru a motivat cercetările menite a găsi abordări alternative pentru metodele de tip Projection Pursuit, aducând optimizări nu numai în ceea ce privește structurile de date, ci și în ceea ce privește resursele de calcul [31], [55]. Multe eforturi depuse de comunitatea științifică în ultimul deceniu s-au concentrat pe dezvoltarea implementărilor paralele ale unui caz particular al metodelor Projection Pursuit, care este PCA, pentru a obține performanțe sporite utilizând arhitecturi paralele de calcul, cum ar fi procesoare multi-core sau GPU.

5.2 Contribuții originale

5.2.1 Paralelizarea algoritmului gaPCA

Această secțiune prezintă eforturile de cercetare îndreptate spre dezvoltarea implementărilor paralele ale algoritmului gaPCA pe diverse arhitecturi de calcul și folosirea mai multor limbaje de programare și scripting: C++, Matlab și Python pentru versiunile CPU multi-core și PyCUDA și CUDA care rulează pe GPU-uri NVIDIA. A fost efectuată o analiză comparativă a timpului de execuție în fiecare caz, care arată superioritatea netă a implementărilor paralele în ceea ce privește reducerea timpului de execuție.

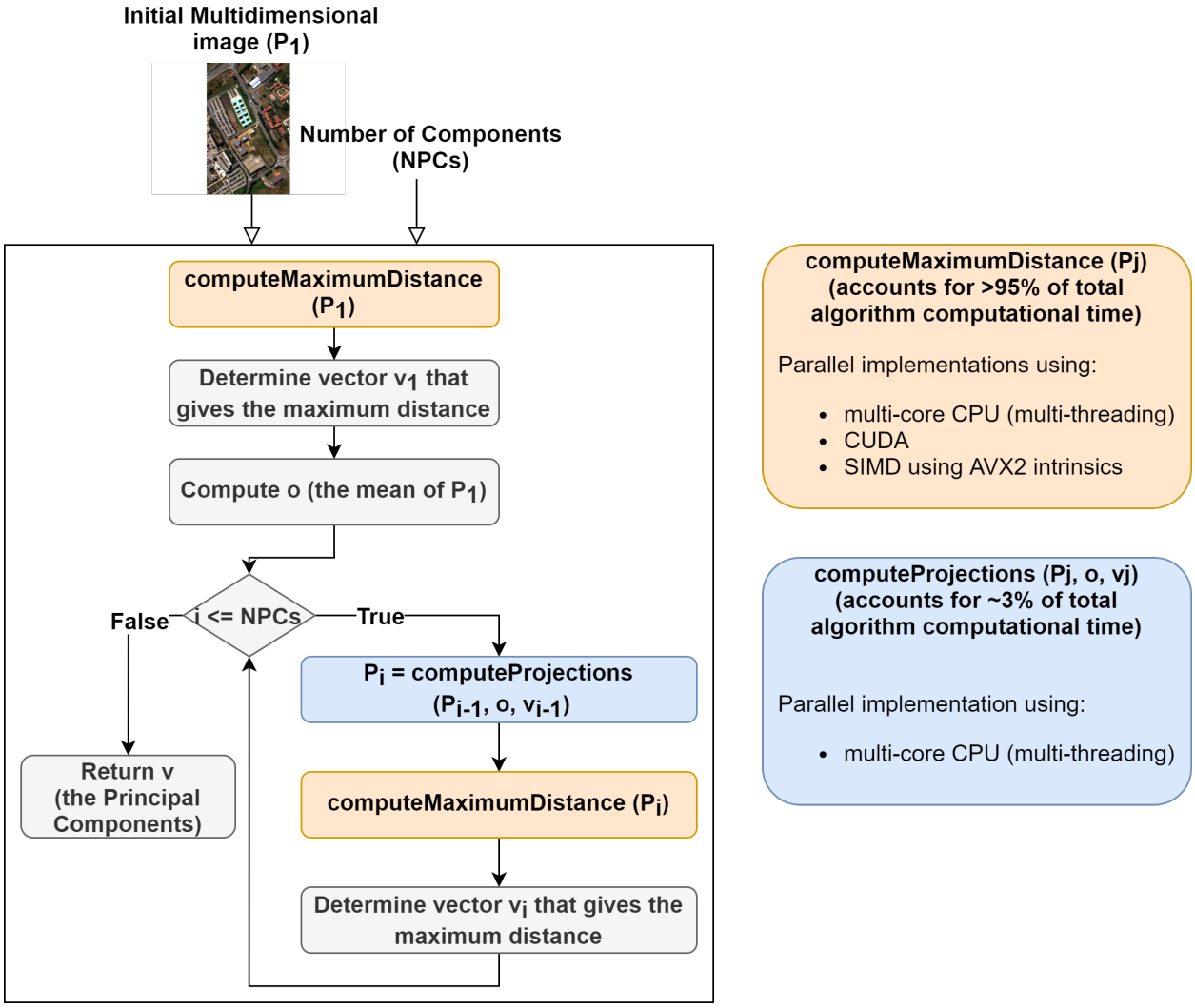


Figure 5.1: Diagrama care prezintă designul algoritmului gaPCA și cele două sub-rutine cu implementări paralele.

Primul pas în elaborarea implementărilor paralele ale algoritmului gaPCA a fost profilarea codului algoritmului (implementarea secvențială, inițială) pentru determinarea timpilor de execuție intermediari pentru fiecare dintre sub-rutinele metodei și, prin urmare, stabilirea părților din algoritm adecvate pentru implementări paralele. Rezultatele acestei etape de

profilare a codului sunt ilustrate într-o schemă din figura 5.1, unde este evidențiată contribuția fiecărei sub-rutine a metodei gaPCA la timpul total de execuție. Considerând cazul în care doar calculul distanțelor euclidiene (care are o pondere de 94,39% din timpul total al algoritmului) face obiectul unei implementări paralele, ne putem aștepta la o accelerare maximă de până la 20×, care variază în funcție de numărul de procesoare utilizate.

Eforturile de cercetare s-au concentrat pe două direcții: pe de o parte, elaborarea eficientă a implementărilor paralele (SIMD, multi-threading și bazate pe GPU) ale metodei gaPCA, pe de altă parte furnizarea unei analize relevante asupra performanțelor acestor implementări în ceea ce privește timpul de execuție și consumul de energie în urma rulării lor pe mai multe platforme hardware: placa de dezvoltare NVIDIA Jetson Nano, CPU Intel Xeon W3670, CPU NVIDIA GeForce GTX 1050 Ti, CPU AMD Ryzen 5 3600 și NVIDIA GeForce GTX 1650.

5.2.2 Implementările Matlab, Python și PyCUDA

Implementarea inițială în Matlab (Listing 5.1) [43] a fost elaborată folosind instrucțiunile specifice din Matlab Parallel Processing Toolbox. Experimentele au fost realizate în Matlab R2019a, pe sistemul de operare Linux Ubuntu 18.04.

```
function [i_extreme, j_extreme, dist_max] euclidDist(X)
{
[m, n] = size(X);
parfor i=1: m-1
[d(i),j(i)] = max(pdist2(X(i,:),X(i+1:m,:)));
end
[dist_max,ind] = max(d);
i_extreme = ind;
j_extreme = j(ind)+ind;
return
}
```

Listing 5.1: Matlab implementation for the Euclidean Distances function

Următoarele două implementări Python realizează o execuție paralelă a metodei gaPCA pe un procesor multi-core și, respectiv, un GPU NVIDIA. Ambele sunt elaborate folosind bibliotecile Python Numba și Numpy; versiunea paralelă multi-core a procesorului utilizează compilatorul Numba JIT, în timp ce versiunea GPU CUDA a fost elaborată folosind biblioteca PyCUDA Python [11].

```
@jit(nopython=True, parallel=True, nogil=True)
def euclidDist(A):
dist = numpy.zeros(len(A))
index = numpy.zeros(len(A))
for i in prange(len(A)-1):
  temp_dist = numpy.zeros(len(A))
for j in prange(i+1,len(A)):
  temp_dist[j] = numpy.linalg.norm(A[i]-A[j])
dist[i] = numpy.amax(temp_dist)
index[i] = numpy.argmax(temp_dist)
return (numpy.amax(dist), numpy.argmax(dist), int(index[numpy.argmax(dist)]))
```

Listing 5.2: Python implementation for the Euclidean Distances function

5.2.3 Implementarea CUDA

În această lucrare, nucleul CUDA a fost elaborat astfel încât un bloc de thread-uri să calculeze un element al matricei de distanțe, fiecare thread calculând distanța dintre elementele corespunzătoare ale celor două rânduri din matricea de intrare. Pentru a evidenția mai bine funcționalitatea nucleului, aceasta este ilustrată grafic și în figura 5.2. În această reprezentare grafică, setul de date a este matricea de intrare pentru care fiecare rând conține valorile pixelilor din fiecare bandă spectrală. Un bloc de thread-uri calculează distanța dintre rândurile i și j, cu fiecare thread tid_q calculând $(a_{i,q}-a_{j,q})^2$. Odată ce toate thread-urile dintr-un bloc și-au încheiat sarcinile (trecând printr-un pas de sincronizare), un mecanism paralel de "tree reduction" este folosit pentru a calcula distanțele totale între rândurile i și j din setul de date de intrare a însumând toate valorile calculate de thread-uri.

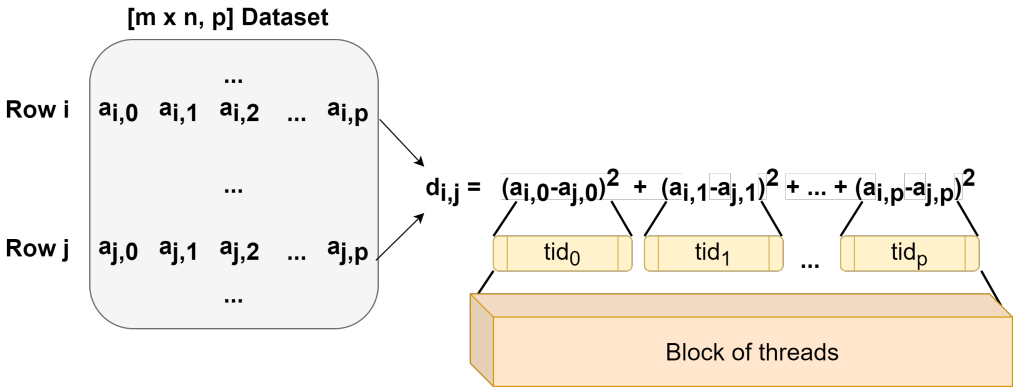


Figure 5.2: Diagramă cu implementările paralele CUDA ale funcției de calcul a distanțelor euclidiene.

Listing-ul 5.3 arată pseudocodul pentru nucleul CUDA (funcția *euclidean*). Această funcție are ca matrice de intrare X și returnează C, un tablou de trei elemente care conține distanța maximă (C[0]) și indecșii celor două puncte cele mai îndepărtate din matricea X (C[1] și C[2]).

```
euclidean kernel(input X, output C):
elem1 = X[blockIdx.x, threadIdx.y];
elem2 = X[blockIdx.y, threadIdx.y];
result = (elem1-elem2)*(elem1-elem2);
accum[threadIdx.y] = result;
Synchronize threads
Perform parallel tree-reduction and compute dist
if (dist>C[0])
C[0] = dist;
update indexes C[1] and C[2];
endif
```

Listing 5.3: Pseudocode for the CUDA kernel computing pairwise Euclidean distance between the rows of the input matrix X

Un vector rezident în memoria partajată intitulat *accumulResult* a fost adăugat la nucleul CUDA; fiecare thread va calcula diferența pătrată corespunzătoare și o va stoca în acest vector în locația determinată de indexul thread-ului. Finalizarea cu succes a acestei etape de către

toate thread-urile este marcată de un punct de control de sincronizare (asigurat prin apelarea metodei *__syncthreads*();); în continuare se realizează "parallel tree reduction" (așa cum se arată în Listing-ul 5.4) care oferă în final valoarea distanței finale. După calcularea acestei valori finale, un thread din fiecare bloc (a fost ales thread-ul cu $threadIdx.y = 0$, pentru uniformitate) realizează comparația acestei valori cu distanța maximă calculată anterior și, dacă este cazul, actualizează distanța maximă stocată în consecință, împreună cu indecșii respectivi.

```
__syncthreads();
// Parallel tree-reduction
for (int stride = SIZE/2 ; stride > 0 ; stride >>= 1) {
if (ty < stride)
accumResult[tx*SIZE+ty] += accumResult[stride + tx*SIZE+ty];
__syncthreads();
}
```

Listing 5.4: CUDA kernel code for parallel tree-reduction

5.2.4 Implementările C++

Primele două versiuni C++ ale algoritmului gaPCA sunt o implementare single-core și o implementare multi-threading bazată pe OpenMP, o "extensie a compilatorului C/C++/Fortran, care permite adăugarea de paralelism multithreading în codul sursă existent" [9]. Codul multi-threading este prezentat în Listing-ul 5.5; directiva compilatorului "#pragma omp parallel" definește secțiunea de cod desemnată pentru execuție paralelă, această directivă determină alocarea sarcinilor în thread-uri anterior execuției codului; mai precis, această directivă alocă sarcinile de calcul thread-urilor active existente, deci nu "creează" așa-numitul "thread pool". Fiecare dintre thread-urile lansate simultan în execuție calculează o distanță între două rânduri ale matricei de intrare (doi pixeli cu valorile corespunzătoare ale acestora în toate benzile spectrale).

```
void parallelDist(short **X, int n, int m, int& index1, int& index2, long long
    d)
{
long long dist[n] = { 0 };
int index[n] = { 0 };
#pragma omp parallel num_threads(12)
{
#pragma omp for
for (int i =0; i<n-1; i++)
{
long long temp_dist[n] = {0};
for (int j =i+1; j<n; j++)
{
temp_dist[j] = squarediff(m,X[i],X[j]);
}
dist[i] = *max_element(temp_dist,temp_dist+n);
index[i] = distance(temp_dist, max_element(temp_dist,temp_dist+n));
}
}
d = *max_element(dist, dist+n-1);
index1 = distance(dist, max_element(dist,dist+n-1));
```

```
index2 = index[index1];
}
```

Listing 5.5: Source code for the C++ multi-core function computing pairwise Euclidean distances between the rows of the input matrix X

A treia implementare C++ a fost elaborată prin extinderea versiunii OpenMP multi-threading cu ajutorul setului de instrucțiuni SIMD. Scopul a fost de a beneficia de paralelismul la nivel de date (DLP) folosind instrucțiunile SIMD care pot lucra pe regiștrii vectori extinși (variind între 64 și 512 biți, în funcție de extensia setului de instrucțiuni SIMD folosit: MultiMedia eXtensions (MMX) [54], Streaming SIMD Extensions (SSE) [58], Advanced Vector eXtensions (AVX) [41]). Folosind această abordare, cea mai mare eficiență (spre 100%) poate fi obținută asigurând că matricea de intrare are o dimensiune multiplu a lățimii registrului vectorial (4, 8, 16 sau 32 de elemente de 16 biți). Experimentele SIMD descrise în această lucrare au fost efectuate pe un procesor AMD Ryzen 5 3600 [1] care acceptă extensia setului de instrucțiuni SIMD AVX2 [2].

```
alignas(__m256i) short **arr = (short **)malloc(ROWS * sizeof(short *));
std::size_t sz = COLS;
for (i=0; i<ROWS; i++)
arr[i] = static_cast<short*>(aligned_alloc(32,sz*2));
```

Listing 5.6: Source code for the C++ alignment of a two-dimensional matrix of short

```

long long squarediff_avx(int size, short *p1, short *p2)
{
std::size_t sz = size;
long long s = 0;
int i = 0;
for (; i + 16 <= size; i+=16 )
{
// load 256-bit chunks of each array
__m256i first_values = _mm256_load_si256((__m256i*)  &p1[i]);

__m256i second_values = _mm256_load_si256((__m256i*) &p2[i]);

//substract each pair of 16-bit integers in the 256-bit chunks
__m256i substracted_values = _mm256_sub_epi16(first_values, second_values);

// multiply each pair of 16-bit integers in the 256-bit chunks
__m256i multiplied_values_lo = _mm256_mullo_epi16(substracted_values,
    substracted_values);
__m256i multiplied_values_hi = _mm256_mulhi_epi16(substracted_values,
    substracted_values);

s += sum_avx(multiplied_values_lo, multiplied_values_hi);
}

for (; i < size; i++)
{
s+= pow(p1[i] - p2[i],2);
}
```

```
return s;
}
```

Listing 5.7: Source code for the C++ SIMD function computing pairwise Euclidean distances

Suma diferențelor pătrate a fost calculată cu o altă funcție C++ pe baza instrucțiunilor SIMD, afișate în Listing-ul 5.8.

```
long long sum_avx(__m256i a_part1, __m256i a_part2)
{
short extracted_partial_sums1[16] = {0};
short extracted_partial_sums2[16] = {0};
_mm256_storeu_si256((__m256i*) &extracted_partial_sums1, a_part1);
_mm256_storeu_si256((__m256i*) &extracted_partial_sums2, a_part2);
long long sssum=0;
for(int i=0;i<16;i++) {
int temp = ((extracted_partial_sums2[i]<<16) | ((extracted_partial_sums1[i]) &
    0xffff));
sssum+=temp;
}
return sssum;
}
```

Listing 5.8: Source code for the C++ multi-core function computing the sum of two 256-bit registers

5.3 Rezultate și discuție

Pentru a evalua accelerarea algoritmului gaPCA, s-a măsurat timpul de execuție al algoritmului pe fiecare selecție din cele două seturi de date menționate mai sus, pentru 1, 3 și 5 componente principale. Algoritmul a fost executat de mai multe ori pentru fiecare test (pentru a minimiza timpul asociat fenomenului de "cache warming" cite 1364731) și s-au mediat rezultatele obținute.

5.3.1 Python vs. PyCUDA

Jetson Nano Prima evaluare comparativă între multi-core CPU în Python și implementările CUDA GPU a fost efectuată pe platforma Jetson Nano.

Testele experimentale au fost realizate pe primele cinci selecții din imaginea Pavia University, de la 20×20 la 125×125 și toate benzile spectrale (103) și un număr fix de 6 componente principale calculate. Algoritmul a rulat de 10 ori pentru fiecare dimensiune a imaginii. Rezultatele sunt prezentate în Tabelul 5.1.

Rezultatele arată că, așa cum era de așteptat, implementarea CUDA este de până la 5,72 × mai rapidă decât versiunea CPU pentru cele mai mari dimensiuni (80×80, 100×100 și 125×125).

Table 5.1: Timing analysis of gaPCA for varying image crop size of the Pavia University dataset on the Jetson Nano platform.

Image dimensions	CPU (seconds)	GPU (seconds)	Speedup
20x20	1.15	1.98	0.58×
40x40	15.40	5.26	2.93×
80x80	229.40	45.88	5×
100x100	556.94	104.45	5.33×
125x125	1347.41	235.59	5.72×

Rezultatele de accelerare obținute sunt de asemenea prezentate grafic în Figura 5.3. Se poate observa că pentru testele de dimensiuni mai mici accelerarea este mai redusă (pentru 40×40 chiar sub 3×) sau versiunea CUDA este chiar mai lentă decât CPU pentru cea mai mică dimensiune 20×20, 0,58× accelerare). Acest lucru se datorează "overhead"-ului CUDA (adică timpii asociați cu transferul datelor din memoria CPU în memoria GPU și invers). Impactul acestui fenomen se reduce, pe măsură ce dimensiunea setului de date crește.

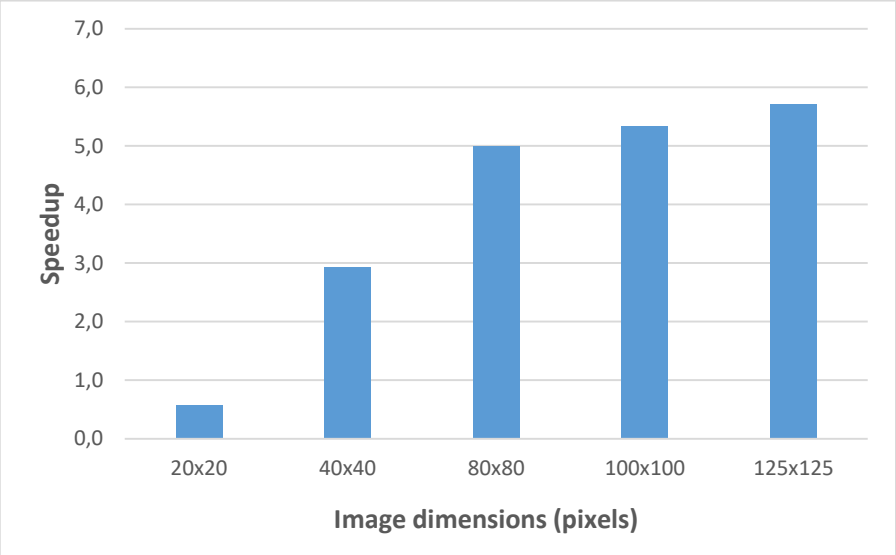


Figure 5.3: Accelerarea între implementările gaPCA GPU și CPU pe Jetson Nano.

Intel Xeon W3670 - GTX 1050Ti Cea de-a doua evaluare comparativă a fost realizată între implementarea multi-core CPU în Python și implementarea CUDA GPU pe platforma Intel Xeon W3670 - GTX 1050Ti.

Experimentele au fost realizate pe toate dimensiunile imaginilor, atât pentru Indian Pines (Tabelul 5.2) cât și pentru Pavia University (Tabelul 5.4), fiind calculate 1, 3 și 5 componente principale.

Rezultatele arată că din nou implementarea CUDA este mai rapidă decât cea CPU Python de până la 7,55×. Cele mai mari accelerări sunt întâlnite pentru imaginile de dimensiunile cele mai mari ale setului de date Indian Pines.

Table 5.2: Timpii de execuție ai algoritmului gaPCA GPU vs. CPU (secunde) pe Indian Pines pentru diferite dimensiuni ale imaginii și diverse numere de componente principale pe platforma Intel Xeon W3670 - GTX 1050Ti.

Image dimensions	GPU 1 PC	CPU 1 PC	GPU 3 PC	CPU 3 PC	GPU 5 PC	CPU 5 PC
20×20	0.45	0.68	0.45	1.18	0.72	1.28
40×40	0.57	1.45	1.01	3.18	1.68	4.55
80×80	2.68	14.61	7.40	33.97	12.02	52.68
100×100	5.46	33.84	15.94	80.12	25.66	126.00
145×145	20.45	154.37	61.25	353.37	102.20	561.91

Table 5.3: Rezultatele accelerării GPU vs. CPU pentru calcularea 1, 3 și 5 componente principale pentru Indian Pines pe platforma Intel Xeon W3670 - GTX 1050Ti.

Image dimensions	1 PC	3 PC	5 PC
20×20	1.51×	1.65×	1.80×
40×40	2.56×	3.15×	2.71×
80×80	5.46×	4.59×	4.38×
100×100	6.20×	5.03×	4.91×
145×145	7.55×	5.77×	5.50×

Rezultatele accelerării obținute sunt prezentate și grafic în Figura 5.4.

Tabelul 5.4 și Tabelul 5.5 arată timpii de execuție pentru implementările CPU și GPU pentru setul de date Pavia University, pe platforma Intel Xeon W3670 - GTX 1050Ti. Accelerarea este de până la 11,78× pentru versiunea GPU a algoritmului în comparație cu cea CPU. Câștigul de viteză este cel mai semnificativ pentru dimensiunile mai mari ale imaginilor din setul de date Pavia University. Pe măsură ce dimensiunile imaginii cresc, paralelizarea pe GPU devine din ce în ce mai eficientă din punct de vedere al timpilor de execuție.

Rezultatele accelerării obținute pentru setul de date Pavia University pe platforma Intel Xeon W3670 - GTX 1050Ti sunt reprezentate grafic în Figura 5.5.

5.3.2 Matlab vs. Python și PyCUDA

Tabelul 5.6 arată rezultatele de timp pentru cele trei implementări, pentru fiecare dimensiune a imaginii setului de date Indian Pines. În mod similar, Tabelul 5.7 arată rezultatele de timp pentru setul de date Pavia University. Figurile 5.6 și 5.7 prezintă accelerarea în cazul celor trei implementări (cu implementarea Matlab luată ca referință pentru comparație).

Pentru primul experiment (Indian Pines), implementarea Matlab este mai rapidă decât cea Python de până la 4,58× pentru imaginea de 40×40, în timp ce pentru celelalte dimensiuni ale imaginii (80×80, 100×100 și 145×145), implementarea Python este mai rapidă. Pentru cel de-al doilea experiment (Pavia University), se observă faptul că implementarea Matlab este mai

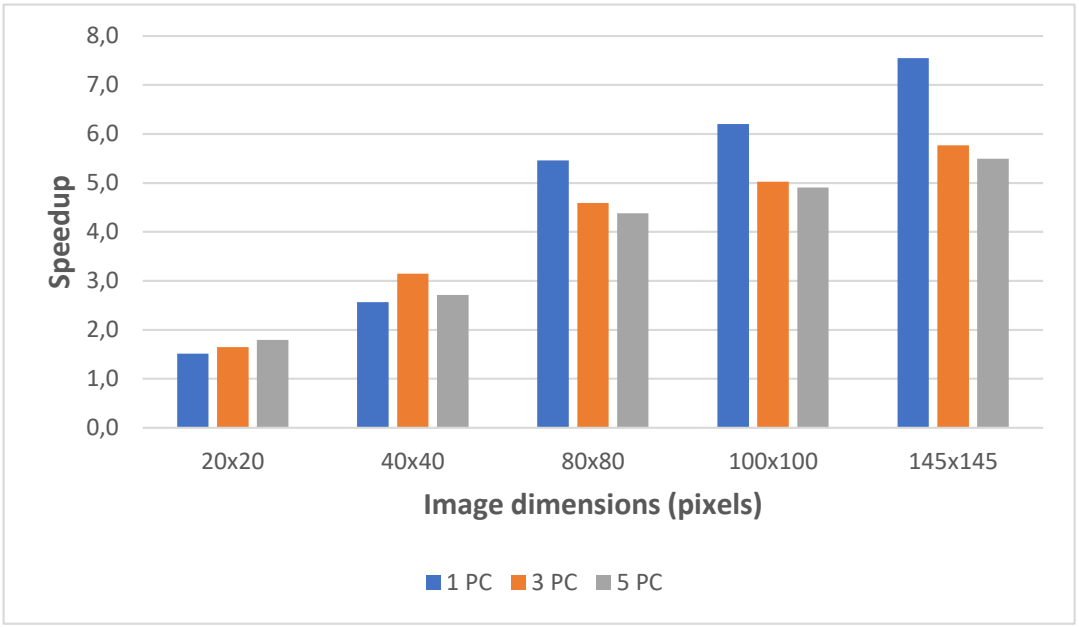


Figure 5.4: Accelerarea CPU vs. GPU pentru calculul a 1, 3 și 5 componente principale pentru setul de date Indian Pines pe platforma Intel Xeon W3670 - GTX 1050Ti.

Table 5.4: Timpii de execuție gaPCA GPU vs. CPU în cazul setului de date Pavia University pentru diferite dimensiuni ale imaginilor și diverse numere de componente principale pe platforma Intel Xeon W3670 - GTX 1050Ti.

Image dimensions	GPU 1 PC	CPU 1 PC	GPU 3 PC	CPU 3 PC	GPU 5 PC	CPU 5 PC
20x20	0.23	0.67	0.40	1.14	0.64	1.25
40x40	0.25	1.18	0.68	2.72	1.12	3.80
80x80	1.06	8.70	3.20	26.15	5.35	42.04
100x100	2.16	21.41	6.72	62.27	11.23	102.05
200x200	29.44	343.98	90.17	984.66	150.96	1611.97
250x250	71.72	812.26	217.11	2373.87	362.32	3921.91
300x300	155.01	1719.58	444.90	4936.99	742,93	8179.16
400x340	335.10	3947.68	1007.32	11338.15	1679.32	18633.12
610x340	774.35	9035.44	2336.16	26188.72	3890.46	43145.17

Table 5.5: Accelerarea CPU vs. GPU pentru calcularea a 1, 3 sau 5 componente principale pentru Pavia University pe platforma Intel Xeon W3670 - GTX 1050Ti.

Image dimensions	1 PC	3 PC	5 PC
20x20	2.93×	1.80×	1.96×
40x40	4.64×	3.99×	3.38×
80x80	8.19×	8.17×	7.86×
100x100	9.90×	9.27×	9.09×
200x200	11.68×	10.92×	10.68×
250x250	11.33×	10.93×	10.82×
300x300	11.09×	11.10×	11.01×
400x340	11.78×	11.26×	11.10×
610x340	11.67×	11.21×	11.09×

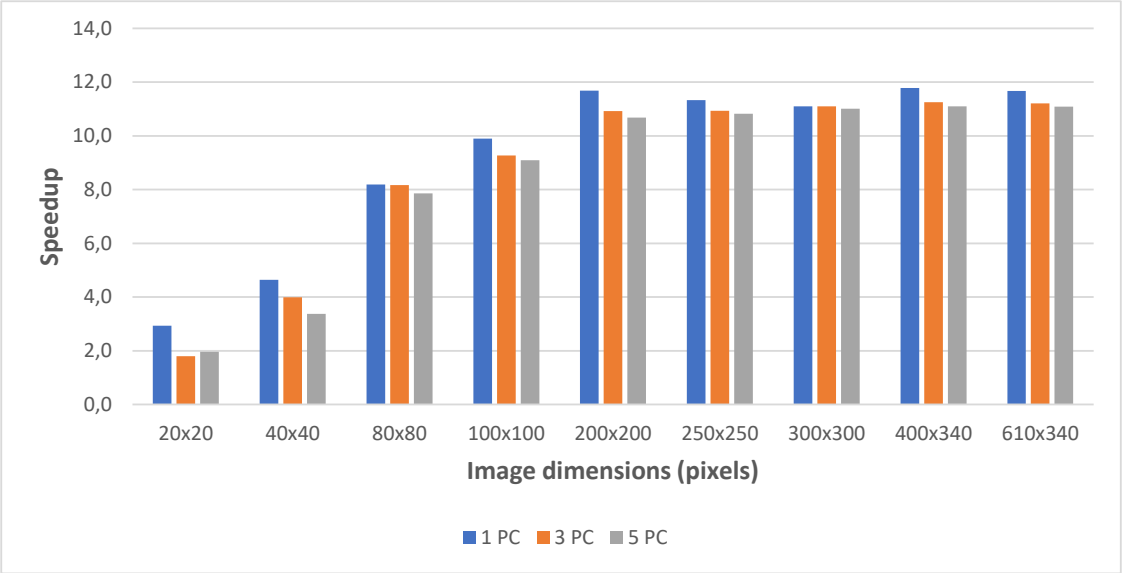


Figure 5.5: Accelerarea CPU vs. GPU pentru calculul a 1, 3 și 5 componente principale în cazul setului de date Pavia University pe platforma Intel Xeon W3670 - GTX 1050Ti.

rapidă decât cea Python, cu aproximativ 20% în medie (variind de la 4% pentru cazul 200×200 și 5 componente principale până la 42% pentru imaginea 100×100 și o singură componentă principală).

Crop size	No. of PCs	Matlab	Python	PyCUDA
40x40	1	0.275	1.260	0.153
40x40	3	0.832	2.802	0.449
40x40	5	1.448	3.519	0.756
80x80	1	8.326	6.797	0.769
80x80	3	24.678	18.265	2.319
80x80	5	40.990	29.333	3.884
100x100	1	22.090	14.531	1.592
100x100	3	66.377	41.929	4.843
100x100	5	110.449	68.647	8.004
145x145	1	104.134	64.498	5.843
145x145	3	313.070	181.152	18.036
145x145	5	521.057	298.212	30.137

Table 5.6: Timpii de execuție Matlab vs.Python vs. PyCUDA (s) pentru setul de date Indian Pines.

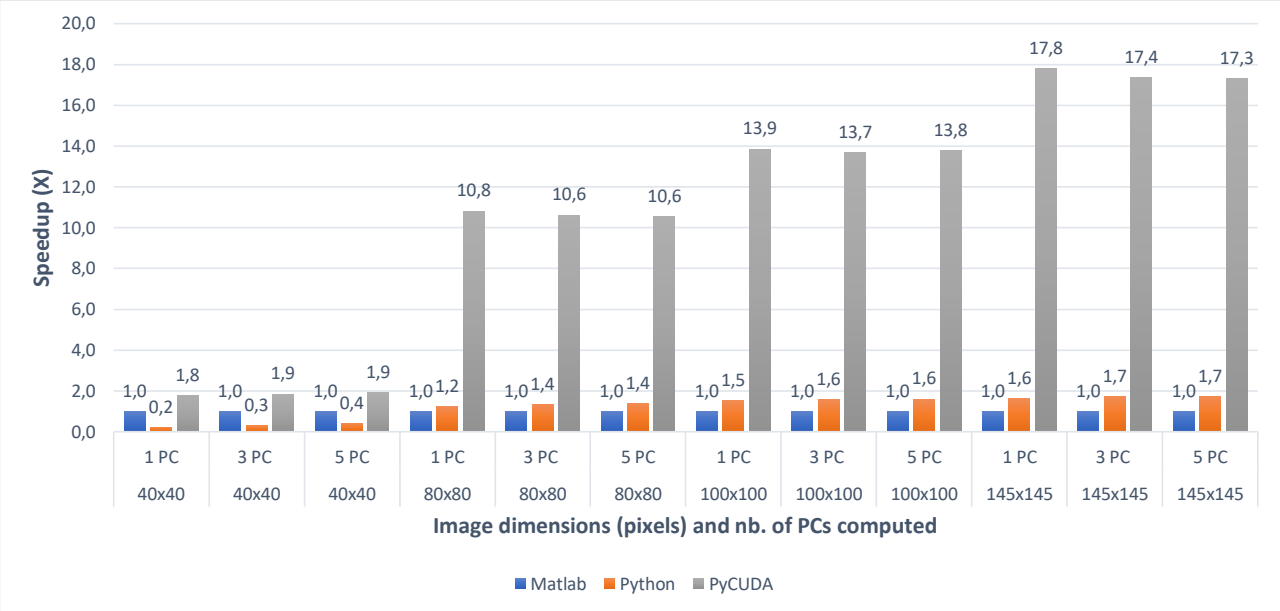


Figure 5.6: Accelerarea Matlab vs.Python vs. PyCUDA pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) pentru diferite dimensiuni ale imaginii în cazul setului de date Indian Pines.

Crop size	No. of PCs	Matlab	Python	PyCUDA
100x100	1	9.507	13.476	0.866
100x100	3	28.439	39.126	2.745
100x100	5	47.580	64.131	4.497
200x200	1	195.801	204.855	10.391
200x200	3	575.083	601.477	31.884
200x200	5	957.494	992.193	53.496
300x300	1	883.342	1027.397	50.905
300x300	3	2653.649	3036.512	155.068
300x300	5	4432.107	5030.831	260.203
610x340	1	4501.181	5453.752	267.402
610x340	3	13588.632	16035.242	806.866
610x340	5	22675.191	26702.160	1347.312

Table 5.7: Timpii de execuție Matlab vs.Python vs. PyCUDA (s) pentru setul de date Pavia University.

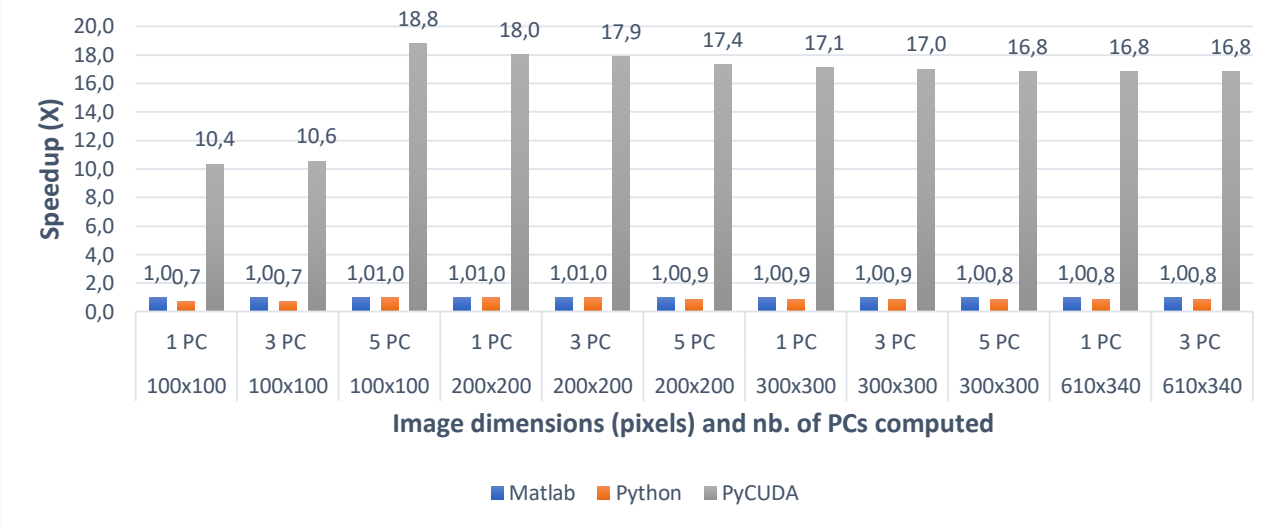


Figure 5.7: Accelearea Matlab vs.Python vs. PyCUDA pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) pentru diferite dimensiuni ale imaginii în cazul setului de date Pavia University.

5.3.3 C++ single core vs. multicore

A doua evaluare comparativă a fost realizată între timpii de execuție ai mai multor implementări C++: single core (SC), single core cu AVX2 (SC AVX2), multi-core (MC) și multi-core folosind AVX2 (MC AVX2). Tabelul 5.8 arată rezultatele pentru cele patru implementări, pentru fiecare dimensiune de imagine a setului de date Indian Pines. În mod similar, Tabelul 5.9 arată rezultatele pentru setul de date Pavia University.

Accelerările pentru cele patru implementări (cu implementarea C++ SC luată ca referință

pentru comparație) sunt prezentate în Figurile 5.8 și 5.9, pentru toate dimensiunile de imagine ale celor două seturi de date Indian Pines și Pavia University.

Crop size	No. of PCs	C++ SC	C++ SC AVX2	C++ MC	C++ MC AVX2
40x40	1	0.947	0.206	0.169	0.047
40x40	3	2.858	0.620	0.480	0.120
40x40	5	4.749	1.035	0.796	0.211
80x80	1	15.147	3.317	2.502	0.663
80x80	3	45.274	9.942	7.534	1.674
80x80	5	75.451	16.389	12.457	3.012
100x100	1	36.834	8.070	6.108	1.605
100x100	3	110.570	23.834	18.452	4.560
100x100	5	184.189	40.409	30.627	7.871
145x145	1	162.853	35.185	27.017	6.797
145x145	3	491.379	105.084	81.027	18.816
145x145	5	814.208	175.127	135.510	32.400

Table 5.8: Timpii de execuție pentru implementările Indian Pines C++ Single Core (SC) vs. Single Core AVX2 (SC AVX2) vs. Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) (s).

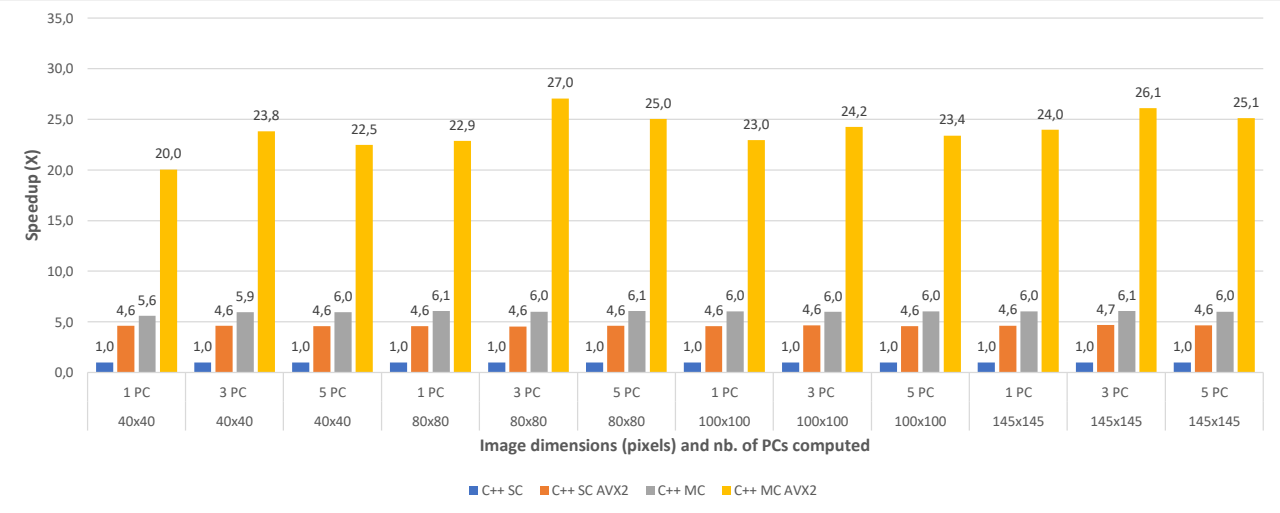


Figure 5.8: Accelerările pentru implementările Indian Pines C++ Single Core (SC) vs. Single Core AVX2 (SC AVX2) vs. Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) pentru diverse dimensiuni ale imaginilor.

5.3.4 C++ multi core vs. CUDA

Evaluarea comparativă finală a fost realizată între timpii de execuție ai celor trei implementări paralele C++: multi-core (MC), multi-core folosind AVX2 (MC AVX2) și multi-core folosind CUDA (MC CUDA). Tabelul 5.10 arată rezultatele de timp pentru cele trei implementări,

Crop size	No. of PCs	C++ SC	C++ SC AVX2	C++ MC	C++ MC AVX2
100x100	1	18.373	4.63084	3.030	0.805
100x100	3	55.185	13.9219	9.025	2.814
100x100	5	91.666	23.1688	15.277	4.565
200x200	1	293.311	74.8406	48.650	12.652
200x200	3	880.324	222.279	144.703	40.199
200x200	5	1472.080	371.005	243.616	68.642
300x300	1	1488.370	387.629	247.666	67.894
300x300	3	4489.640	1165.61	741.890	211.110
300x300	5	7438.200	1933.44	1240.140	363.606
610x340	1	7956.360	2053.12	1322.530	393.734
610x340	3	23962.794	6202.35	3975.840	1144.730
610x340	5	40190.385	10408.8	6605.060	1905.980

Table 5.9: Timpii de execuție pentru Pavia University C++ Single Core (SC) vs. Single Core AVX2 (SC AVX2) vs. Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) (s).

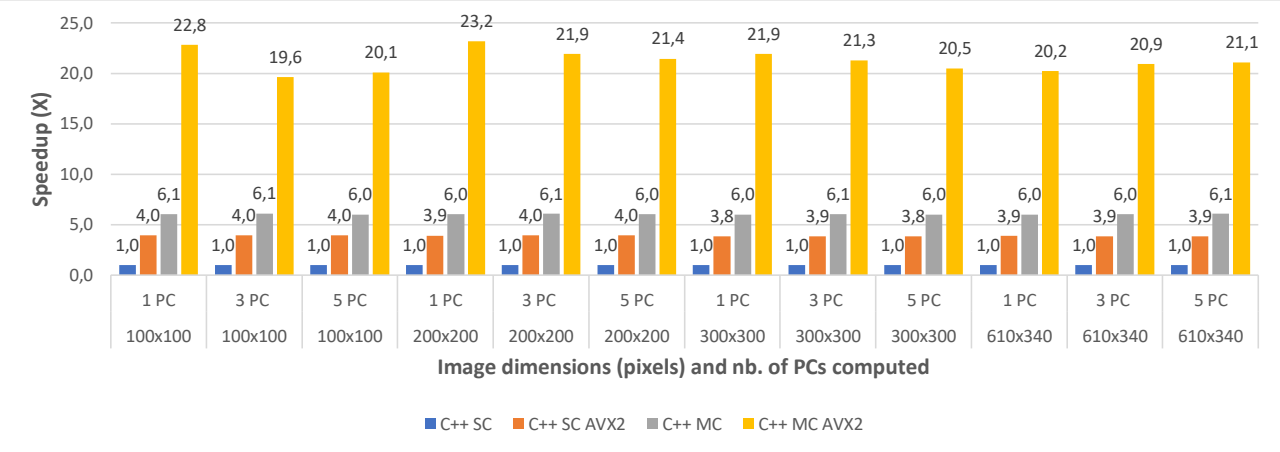


Figure 5.9: Accelerările pentru implementările Pavia University C++ Single Core (SC) vs. Single Core AVX2 (SC AVX2) vs. Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) pentru diverse dimensiuni ale imaginilor.

pentru fiecare dimensiune a imaginilor setului de date Indian Pines. În mod similar, Tabelul 5.11 prezintă rezultatele setului de date Pavia University.

Crop size	No. of PCs	C++ MC	C++ MC AVX2	C++ MC CUDA
40x40	1	0.169	0.047	0.113
40x40	3	0.480	0.120	0.202
40x40	5	0.796	0.211	0.239
80x80	1	2.502	0.663	0.585
80x80	3	7.534	1.674	1.619
80x80	5	12.457	3.012	2.654
100x100	1	6.108	1.605	1.324
100x100	3	18.452	4.560	3.835
100x100	5	30.627	7.871	6.343
145x145	1	27.017	6.797	5.609
145x145	3	81.027	18.816	16.690
145x145	5	135.510	32.400	27.770

Table 5.10: Timpii de excecuție pentru Indian Pines C++ Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) vs. Multi Core CUDA (MC CUDA) (s).

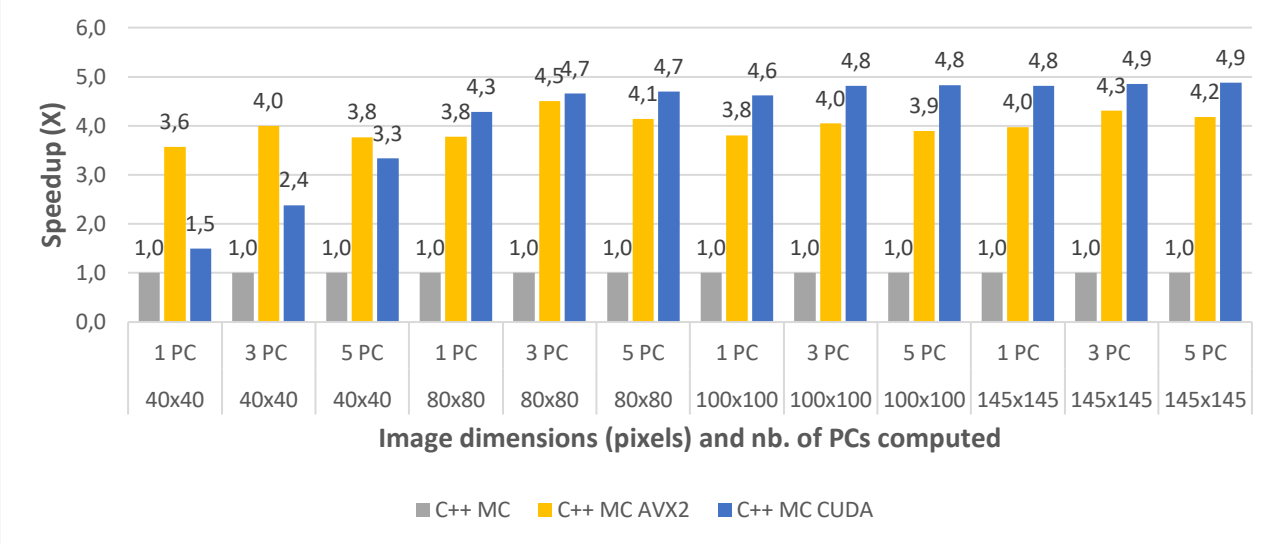


Figure 5.10: Accelerarea pentru Indian Pines C++ Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) vs. Multi Core CUDA (MC CUDA) pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) și diferite dimensiuni ale imaginilor.

Figura 5.12 arată o comparație a tuturor implementărilor în ceea ce privește accelerarea în comparație cu versiunea C++ SC luată ca referință. Această perspectivă comparativă confirmă faptul că cele două implementări bazate pe CUDA (C++ MC CUDA și PyCUDA) dau cele mai mari accelerări, urmate îndeaproape de versiunea C++ MC AVX2. C++ MC CUDA este mai rapid decât versiunea PyCUDA în medie cu 9,3% pentru Pavia și cu approx. 30% pentru Indian Pines.

Crop size	No. of PCs	C++ MC	C++ MC AVX2	C++ MC CUDA
100x100	1	3.030	0.805	0.689
100x100	3	9.025	2.814	1.916
100x100	5	15.277	4.565	3.143
200x200	1	48.650	12.652	9.989
200x200	3	144.703	40.199	29.699
200x200	5	243.616	68.642	49.068
300x300	1	247.666	67.894	50.700
300x300	3	741.890	211.110	151.082
300x300	5	1240.140	363.606	251.730
610x340	1	1322.530	393.734	267.495
610x340	3	3975.840	1144.730	801.950
610x340	5	6605.060	1905.980	1336.010

Table 5.11: Timpii de excecuție pentru Pavia University C++ Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) vs. Multi Core CUDA (MC CUDA) (s).

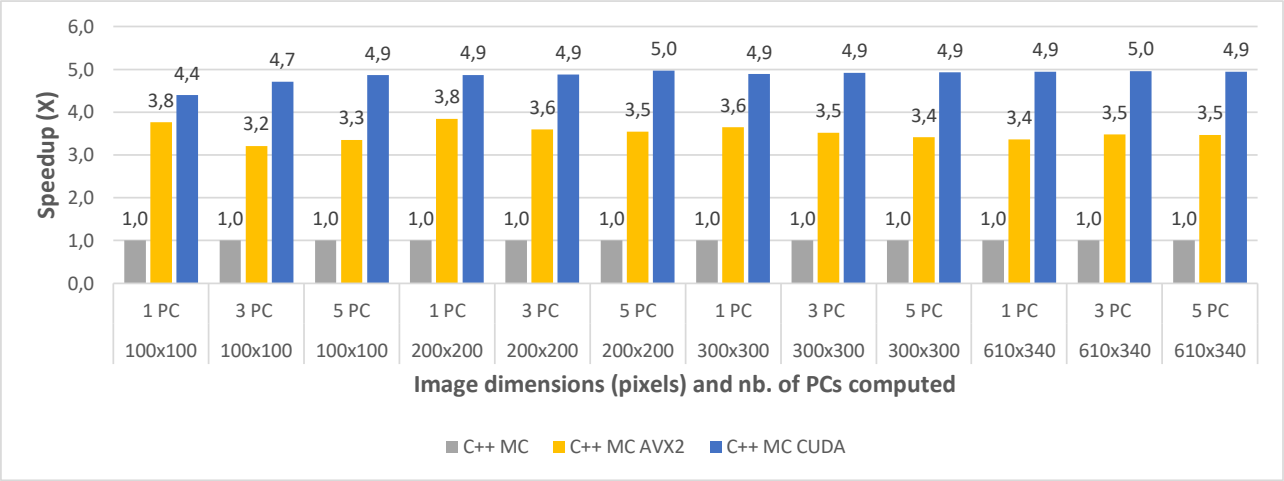


Figure 5.11: Accelerarea pentru Pavia University C++ Multi Core (MC) vs. Multi Core AVX2 (MC AVX2) vs. Multi Core CUDA (MC CUDA) pentru 1 PC (a) 3 PCs (b) și 5 PCs (c) și diferite dimensiuni ale imaginilor.

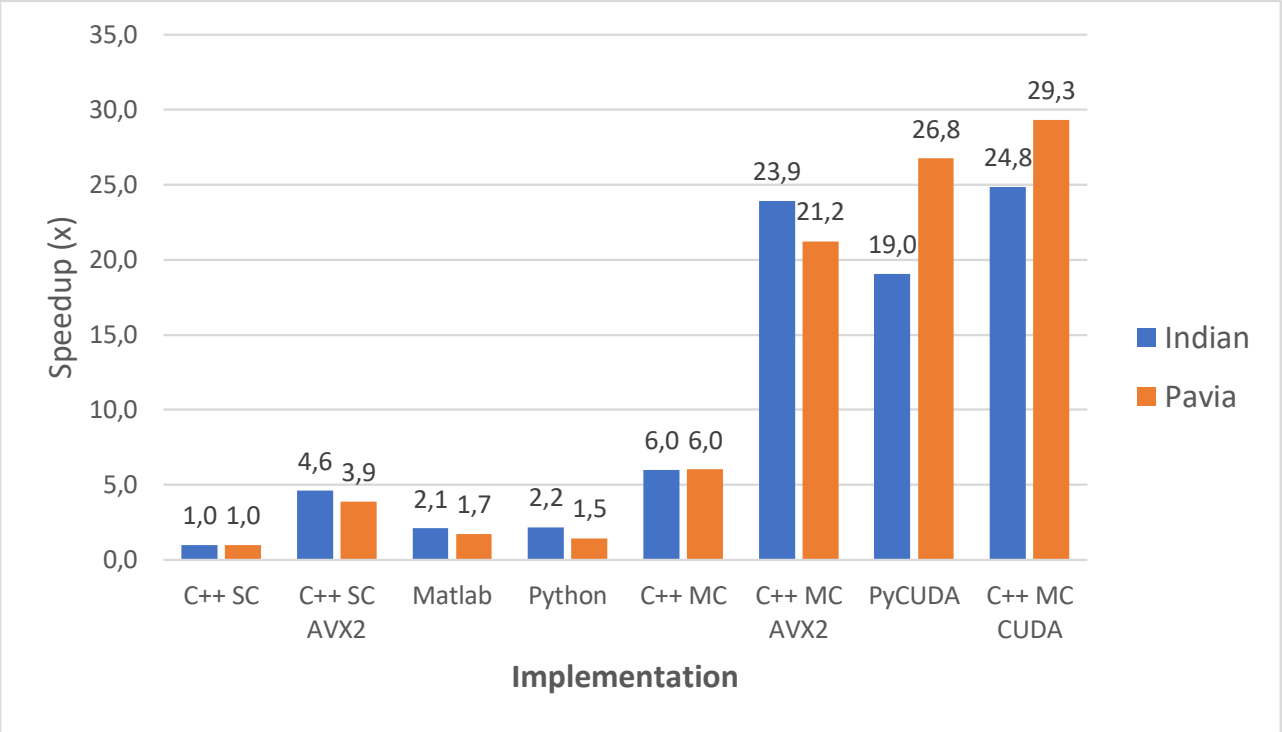


Figure 5.12: Rezultatele comparative ale accelerării diverselor implementări gaPCA.

5.3.5 Eficiența energetică

Pentru evaluarea consumului de energie au fost realizate măsurători în timpul executării fiecărei implementări a algoritmului pentru două cazuri de testare: Indian Pines 100×100 și Pavia University 200×200; în toate cazurile au fost calculate 5 componente principale. Consumul de energie al sistemului a fost măsurat folosind un dispozitiv PeakTech 1660 [10], care este echipat cu o conexiune USB și un software dedicat pentru achiziția de date, crescând astfel precizia și fiabilitatea rezultatelor măsurătorii.

Tabelul 5.12 arată consumul de energie pentru implementările Matlab, Python și PyCUDA comparativ cu timpul total de execuție; rezultatele energetice sunt afișate și în Figura 5.13. În mod similar, consumul de energie pentru implementările C++ este ilustrat în Tabelul 5.13 și în Figura 5.14.

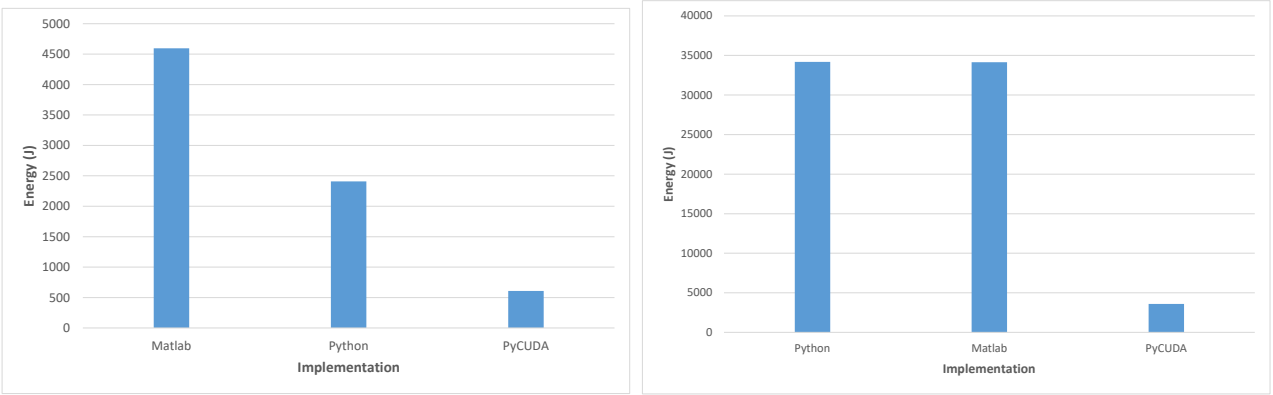
Implementarea C++ MC CUDA este dovedită a fi puțin mai puțin eficientă din punct de vedere energetic decât implementarea C++ MC AVX2 (consumând cu 24,74% mai multă energie în cazul testului Indian Pines și cu 4,89% mai multă energie în cazul de testare al Pavia University), dar are o viteză de execuție mai bună decât implementarea C++ MC AVX2 (fiind cu 24,07% mai rapidă pentru testul Indian Pines și cu 39,89% mai rapidă pentru Pavia University). Diferența între consumul de energie pentru cele două cazuri de testare (24,74% Indian Pines față de 4,89% Pavia University) poate fi explicată prin diferența numărului de benzi spectrale (200 pentru Pines Indian vs. 103 pentru Pavia University), ceea ce influențează numărul de thread-uri per bloc utilizate de nucleul CUDA (256 pentru Indian Pines vs. 128 pentru Pavia University). Aceste rezultate sunt confirmate și în cazul implementării PyCUDA, care folosește același nucleu CUDA.

Dataset	Size	No. of PCs	Implementation	Energy (J)	Time (s)
Indian	100x100	5	Matlab	4595.29	110.449
			Python	2409.06	68.647
			PyCUDA	609.23	8.004
Pavia U	200x200	5	Matlab	34139.77	957.494
			Python	34192.04	992.193
			PyCUDA	3589.8	53.496

Table 5.12: Măsurătorile de energie pentru implementările Matlab, Python și PyCUDA.

Dataset	Size	No. of PCs	Implementation	Energy (J)	Time (s)
Indian	100x100	5	C++ SC	3672	184.189
			C++ MC	1108.75	30.627
			C++ SC AVX2	792	40.409
			C++ MC CUDA	471.43	6.343
			C++ MC AVX2	378	7.871
Pavia U	200x200	5	C++ SC	27512.87	1472.080
			C++ MC	9242.40	243.616
			C++ SC AVX2	7431.87	371.005
			C++ MC CUDA	3491.43	49.068
			C++ MC AVX2	3328.63	68.642

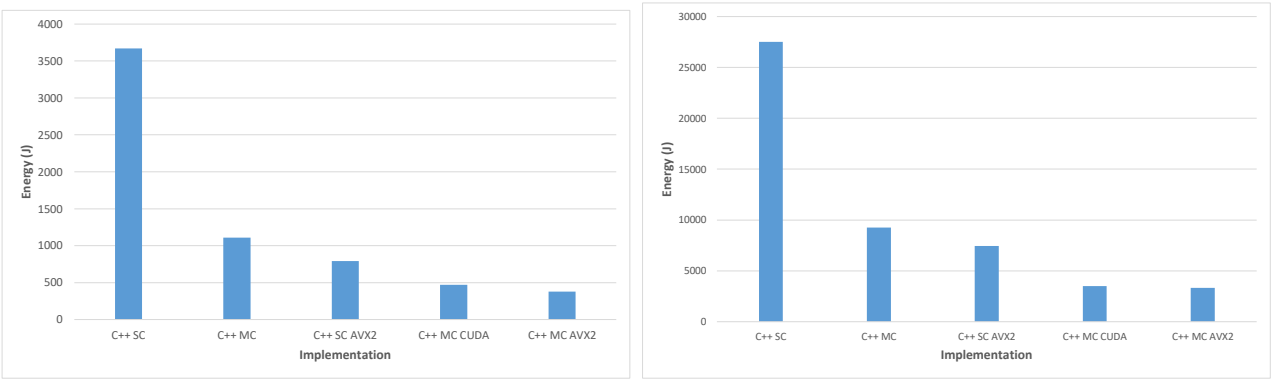
Table 5.13: Măsurătorile de energie pentru implementările C++.



(a) Indian Pines 100×100 5 PCs

(b) Pavia University 200×200 5 PCs

Figure 5.13: Consumul de energie în cazul seturilor de date Indian Pines (a) și Pavia University (b) pentru implementările Matlab, Python și PyCUDA.



(a) Indian Pines 100×100 5 PCs

(b) Pavia University 200×200 5 PCs

Figure 5.14: Consumul de energie în cazul seturilor de date Indian Pines (a) și Pavia University (b) pentru implementările C++.

5.4 Concluzii

Progresele recente din ultimii ani în domeniul Remote Sensing au dus la o creștere susținută a cantității de date geospațiale achiziționate. Acest lucru ridică provocări științifice legate de procesarea, analiza și vizualizarea datelor de Remote Sensing, cu accent pe algoritmi și metode de calcul capabile să extragă informații utile semnificative atât offline, cât și în timp real. Prin urmare, cele mai noi dispozitive și tehnici de calcul de înaltă performanță, cum ar fi calculul paralel, GPU-urile și seturile de instrucțiuni precum AVX2 reprezintă soluții capabile să ofere avantaje semnificative în ceea ce privește creșterea performanței aplicațiilor și algoritmilor ce procesează date de Remote Sensing.

În acest capitol s-a prezentat implementarea algoritmului gaPCA pe arhitecturi paralele de calcul: CPU single-core/multi-core, GPU și CPU multi-core folosind AVX2, împreună cu o evaluare comparativă a implementărilor în termeni de viteză de execuție și consum de energie. Evaluarea experimentală a arătat că toate implementările paralele au viteze mult superioare versiunii single core: C++ MC CUDA a fost în medie de 29,3× mai rapidă pe Pavia și de 24,8× mai rapidă în cazul Indian Pines, în timp ce versiunea C++ MC AVX2 a avut o accelerare medie de 21,2 × pentru Pavia și 23,9× pentru Indian Pines în comparație cu versiunea de referință C++ SC. Aceste rezultate arată nu numai avantajele utilizării programării CUDA în implementarea algoritmului gaPCA pe un GPU în termeni de performanță și consum de energie, dar și avantaje considerabile în implementarea acestuia pe procesoare multi-core folosind instrucțiuni AVX2. Aceste două implementări s-au dovedit a fi mult mai rapide decât implementarea standard multi-core și, de asemenea, cele mai eficiente în ceea ce privește consumul de energie. Versiunea C++ MC AVX2 s-a dovedit a fi cea mai eficientă, necesitând, în medie, de 8,26× mai puțină energie la experimentele pe Pavia și de 9,71 × mai puțin energie la experimentele pe setul de date Indian Pines în comparație cu implementarea de referință C++ SC. Versiunea C++ MC CUDA a avut doar rezultate foarte apropiate, fiind de 7,88× mai eficientă pe Pavia și de 7,78× mai eficientă pe Indian Pines decât implementarea C++ SC.

În consecință, acest capitol a evidențiat avantajele utilizării arhitecturilor de calcul paralel, instrucțiunilor AVX2 și paradigmelor de programare paralelă CUDA pentru accelerarea algo-

ritmilor de reducere a dimensionalității, cum ar fi gaPCA, pentru a obține viteze semnificative și eficiență energetică îmbunătățită comparativ cu implementările tradiționale cu un singur nucleu/thread sau cu mai multe nuclee CPU.

Cercetările și experimentele prezentate în acest capitol au fost validate și diseminate în următoarele publicații:

- [43] A. L. Machidon, C. B. Ciobanu, O. M. Machidon, and P. L. Ogrutan. On Parallelizing Geometrical PCA Approximation. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019 indexat și în IEEE Xplore Digital Library
- [47] A. L. Machidon, O. M. Machidon, C. B. Ciobanu, and P. L. Ogrutan. Accelerating a Geometrical Approximated PCA Algorithm Using AVX2 and CUDA. *Remote Sensing*, 12(12):1918, 2020

CAPITOLUL 6

Concluzii finale și contribuții originale

6.1 Concluzii finale

Această cercetare doctorală a fost axată pe conceperea, proiectarea, implementarea, testarea, validarea și optimizarea unei noi metode de reducere a dimensionalității, și anume metoda gaPCA.

În urma studiilor și experimentelor efectuate, prezentate în această teză, se pot trage următoarele concluzii:

- Noua metodă gaPCA se caracterizează prin mai multe avantaje specifice în comparație cu alte metode similare din familia Projection Pursuit. gaPCA are o capacitate îmbunătățită de a discrimina semnale sau obiecte mai mici de pe fundalul scenei și, având în vedere și designul algoritmic, este în mod natural ușor paralizabilă, ceea ce îi permite să fie accelerată pe arhitecturi de tip High Performance Computing. Cea mai consumatoare de timp subrutină în metoda gaPCA este calculul distanței euclidiene, care s-a dovedit a fi accelerabilă folosind platforme de calcul paralele [43].

- Validarea noii metode gaPCA folosind metrici de calitate pe datele de Remote Sensing a arătat că, în comparație cu PCA standard, ambele metode oferă rezultate similare în ceea ce privește contrastul și entropia, în timp ce în cazul metricii de energie, componentele gaPCA au o calitate spațială superioară a imaginii, ceea ce poate duce la rezultate mai bune de clasificare.

- În ceea ce privește performanțele obținute de gaPCA în activitatea de de clasificare pentru cele 4 seturi de date, această nouă metodă a depășit PCA în fiecare caz în ceea ce privește acuratețea totală calculată pentru fiecare set de date și a obținut, de asemenea, un nivel mai mare al acurateții decât PCA pentru majoritatea claselor. O analiză detaliată a rezultatelor în ceea ce privește acuratețea fiecărei clase a confirmat ipoteza inițială potrivit

căreia gaPCA, spre deosebire de metoda PCA standard, ia în considerare informația cu o contribuție mai mică la varianța totală a semnalului, care este considerată redundantă sau lipsită de importanță de către PCA. Prin urmare, gaPCA are o capacitate superioară de a discrimina obiecte mici sau clase similare, caracteristică confirmată de performanța sa în experimentele pentru clasele preponderent spectrale în fiecare set de date, unde a depășit PCA standard.

- În experimentul de recunoaștere facială folosind Eigenfaces, pentru primele trei seturi de date gaPCA a avut o acuratețe apropiată sau doar ușor inferioară (sub 10%) cu PCA canonic. Pentru cazul în care recunoașterea facială s-a efectuat folosind un clasificator cu rețea neurală, gaPCA a obținut din nou o acuratețe foarte apropiată de cea a omologului său, cu puțin sub 2% în medie comparativ cu PCA standard, cu rezultate chiar mai bune pentru gaPCA în cazul unor clase specifice. În plus, gaPCA a confirmat că are abilitatea de a scădea numărul de iterații de antrenament necesare pentru clasificatorul bazat pe rețea neurală.

- Algoritmul gaPCA a fost implementat pe arhitecturi paralele de calcul: CPU single-core/multi-core, GPU și CPU multi-core folosind AVX2, împreună cu o evaluare comparativă a implementărilor în termeni de viteză de execuție și consum de energie. Evaluarea experimentală a arătat că toate implementările paralele au viteze mult superioare versiunii single core: C++ MC CUDA a fost în medie de 29,3× mai rapidă pe Pavia și de 24,8× mai rapidă în cazul Indian Pines, în timp ce versiunea C++ MC AVX2 a avut o accelerare medie de 21,2 × pentru Pavia și 23,9× pentru Indian Pines în comparație cu versiunea de referință C++ SC. Aceste rezultate arată nu numai avantajele utilizării programării CUDA în implementarea algoritmului gaPCA pe GPU în termeni de performanță și consum de energie, dar și avantaje considerabile în implementarea acestuia pe procesoare multi-core folosind instrucțiuni AVX2. Experimentele au evidențiat așadar avantajele utilizării arhitecturilor de calcul paralel, instrucțiunilor AVX2 și paradigmelor de programare paralelă CUDA pentru accelerarea algoritmilor de reducere a dimensionalității, cum ar fi gaPCA, pentru a obține viteze semnificative și eficiență energetică îmbunătățită comparativ cu implementările tradiționale cu un singur nucleu/thread sau cu mai multe nuclee CPU.

- Rezultatele măsurării consumului de energie pentru diversele implementări gaPCA cu un singur nucleu, multi-core și bazate pe CUDA au arătat pe de o parte că toate soluțiile paralele consumă mult mai puțină energie totală decât implementarea cu un singur nucleu și, pe de altă parte, că dintre toate implementările paralele, cea mai eficientă a fost implementarea C++ MC AVX2, urmată îndeaproape de implementarea C++ MC CUDA și versiunea PyCUDA.

6.2 Contribuții originale

Printre contribuțiile originale ale acestei cercetări doctorale se numără următoarele:

- *Un studiu asupra stadiului actual în ceea ce privește algoritmii și metodele pentru analiza datelor multidimensionale*

- *Proiectarea și implementarea unei noi metode de reducere a dimensionalității [45]*
- *Validarea noii metode gaPCA pe date sintetice [44]*
- *Validarea noii metode gaPCA pentru vizualizarea datelor de Remote Sensing [44]*
- *Validarea noii metode gaPCA utilizând metrici de calitate pe datele de Remote Sensing [45]*
- *Validarea noii metode gaPCA pentru clasificarea imaginilor de Remote Sensing [45][46]*
- *Validarea noii metode gaPCA în recunoașterea facială [48]*
- *Paralelizarea noii metode gaPCA folosind multi-core Central Processing Units (CPU), Graphics Processing Units (GPU) și multi-core CPU cu AVX2 intrinsics [47][43]*
- *A fost realizată o evaluare a algoritmului gaPCA și a diverselor implementări ale acestuia în limbaje de programare/scripting și pe diverse platforme hardware multi-core CPU și GPU [47][43]*
- *Analiza consumului de energie a diferitelor implementări gaPCA single-, multi-core și CUDA [47]*

6.3 Direcții de cercetare viitoare

În ceea ce privește direcțiile de cercetare viitoare care pot valorifica și extinde rezultatele obținute, pot fi menționate următoarele:

- Extinderea domeniului de aplicare a metodei gaPCA la domeniul *target detection*.
- Extinderea gamei de aplicații a metodei gaPCA la alte tipuri de date multidimensionale, cum ar fi date biomedicale, înregistrări de date de la senzori, date de trafic etc.
- Extinderea comparației performanțelor metodei gaPCA cu alte metode de reducere a dimensionalității, cum ar fi Linear Discriminant Analysis, Minimum Noise Fraction, etc.

6.4 Diseminarea și validarea rezultatelor cercetării

Rezultatele cercetării doctorale obținute și prezentate în această teză au fost validate de comunitatea științifică internațională și valorificate prin publicarea în reviste de specialitate, precum și în volumele unor conferințe internaționale de prestigiu.

Articole științifice publicate în reviste indexate Web of Science:

- [45] A. L. Machidon, F. Del Frate, M. Picchiani, O. M. Machidon, and P. L. Ogrutan. Geometrical Approximated Principal Component Analysis for Hyperspectral Image Analysis. *Remote Sensing*, 12(11), 2020 (Impact Factor = 4.509)

- [47] A. L. Machidon, O. M. Machidon, C. B. Ciobanu, and P. L. Ogrutan. Accelerating a Geometrical Approximated PCA Algorithm Using AVX2 and CUDA. *Remote Sensing*, 12(12):1918, 2020 (Impact Factor = 4.509)

- [26] L. Fasano, D. Latini, A. L. Machidon, C. Clementini, G. Schiavon, and F. Del Frate. SAR Data Fusion Using Nonlinear Principal Component Analysis. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019 (Impact Factor = 3.833) de asemenea indexat în IEEE Xplore Digital Library

Articole științifice publicate în volumele unor conferințe internaționale indexate în Web of Science - proceedings paper:

- [43] A. L. Machidon, C. B. Ciobanu, O. M. Machidon, and P. L. Ogrutan. On Parallelizing Geometrical PCA Approximation. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019 de asemenea indexat în IEEE Xplore Digital Library

- [48] A. L. Machidon, O. M. Machidon, and P. L. Ogrutan. Face Recognition Using Eigenfaces, Geometrical PCA Approximation and Neural Networks. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pages 80–83, 2019 de asemenea indexat în IEEE Xplore Digital Library

- [44] A. L. Machidon, R. Coliban, O. Machidon, and M. Ivanovici. Maximum Distance-based PCA Approximation for Hyperspectral Image Analysis and Visualization. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018 de asemenea indexat în IEEE Xplore Digital Library

Articole științifice publicate în volumele altor manifestări științifice cu comitet de recenzori:

- [25] L. Fasano, F. Del Frate, D. Latini, A. L. Machidon, and C. Clementini. Classification of Urban areas by Means of Multiband SAR Data Fusion. In *European Space Agency 2nd Mapping Urban Areas from Space 2018 - MUAS 2018*. ESA, 2018

- [46] A. L. Machidon, M. Ivanovici, R. Coliban, and F. Del Frate. A Geometrical Approximation of PCA for Hyperspectral Data Dimensionality Reduction. In *The ESA Earth Observation Phi-week EO Open Science and FutureEO*. ESA, 2018

Contribuții la activitatea didactică în timpul studiilor doctorale:

- [53] P. L. Ogrutan, A. L. Machidon, and A. Dinu. Is There a Link Between Creativity and Multiculturalism in Education? *TEM Journal*, 8(2):577, 2019

Bibliografie

[1] AMD Ryzen 5 3600 Processor Specifications. https://www.amd.com/en/products/cpu/amd-ryzen-5-3600. Accessed: 2019-09-17.

[2] AVX2 Overview. https://software.intel.com/en-us/cpp-compiler-developer-guide-and-reference-overview-intrinsics-for-intel-advanced-v Accessed: 2019-01-22.

[3] Cambridge Database of Faces. https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. Accessed: January 28, 2019.

[4] ENVI online documentation. https://www.harrisgeospatial.com/Portals/0/pdfs/HG_ENVI_brochure_WEB.pdf. Accessed: December 07, 2018.

[5] FEI Face Database. http://fei.edu.br/cet/facedatabase.html. Accessed: January 08, 2019.

[6] Gen-Z: A New Approach. Gen-Z Consortium. https://www.youtube.com/watch?v=OTsk8B-EnmM. Accessed: February 14, 2019.

[7] Mathworks. Principal Component Analysis (PCA). Online documentation. https://www.mathworks.com/help/stats/principal-component-analysis-pca.html. Accessed: February 5, 2018.

[8] Multidimensional - In Merriam-Webster's Dictionary. https://www.merriam-webster.com/dictionary/multidimensional. Accessed: November 27, 2018.

[9] OpenMP. https://www.openmp.org/. Accessed: 2019-12-19.

[10] Peaktech Power Meter. https://www.peaktech.de/productdetail/kategorie/digital-leistungszangenmessgeraet/produkt/peaktech-1660.html/. Accessed: 2020-01-24.

[11] PyCUDA. https://mathema.tician.de/software/pycuda/. Accessed: 2019-08-19.

[12] Yale Face Database. `http://vision.ucsd.edu/content/yale-face-database`. Accessed: January 20, 2019.

[13] Y. Aït-Sahalia and D. Xiu. Principal component analysis of high-frequency data. *Journal of the American Statistical Association*, pages 1–17, 2018.

[14] I. S. Bajwa, M. Naweed, M. N. Asif, and S. I. Hyder. Feature based image classification by using principal component analysis. *ICGST Int. J. Graph. Vis. Image Process. GVIP*, 9:11–17, 2009.

[15] A. Barcaru. Supervised projection pursuit – A dimensionality reduction technique optimized for probabilistic classification. *Chemometrics and Intelligent Laboratory Systems*, 194:103867, 2019.

[16] D. Báscones, C. González, and D. Mozos. Hyperspectral Image Compression Using Vector Quantization, PCA and JPEG2000. *Remote Sensing*, 10(6):907, 2018.

[17] J. B. Campbell and R. H. Wynne. *Introduction to remote sensing*. Guilford Press, 2011.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[19] G. Cheng, J. Han, and X. Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017.

[20] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu. Big Data for Remote Sensing: Challenges and opportunities. *Proceedings of the IEEE*, 104(11):2207–2219, 2016.

[21] A. Davies, K. Lahiri, et al. A new framework for analyzing survey forecasts using three-dimensional panel data. *Journal of Econometrics*, 68(1):205–228, 1995.

[22] S. Ding, L. Chen, and J. Li. Dimension reduction of local manifold learning algorithm for hyperspectral image classification. In *Intelligent Image and Video Interpretation: Algorithms and Applications*, pages 217–231. IGI Global, 2013.

[23] Q. Du and J. E. Fowler. Hyperspectral image compression using JPEG2000 and principal component analysis. *IEEE Geoscience and Remote sensing letters*, 4(2):201–205, 2007.

[24] D. Eni, A. Iwara, and R. Offiong. Analysis of soil-vegetation interrelationships in a south-southern secondary forest of Nigeria. *International Journal of Forestry Research*, 2012, 2012.

[25] L. Fasano, F. Del Frate, D. Latini, A. L. Machidon, and C. Clementini. Classification of Urban areas by Means of Multiband SAR Data Fusion. In *European Space Agency 2nd Mapping Urban Areas from Space 2018 - MUAS 2018*. ESA, 2018.

[26] L. Fasano, D. Latini, A. L. Machidon, C. Clementini, G. Schiavon, and F. Del Frate. SAR Data Fusion Using Nonlinear Principal Component Analysis. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019.

[27] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on computers*, 100(9):881–890, 1974.

[28] V. Hlaváč. Principal Component Analysis (PCA) Application to images. `http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/11ImageProc/15PCA.pdf`. Lecture notes, Czech Technical University in Prague. Accessed: December 20, 2018.

[29] E. Homenauth, D. Kajeguka, and M. A. Kulkarni. Principal component analysis of socioeconomic factors and their association with malaria and arbovirus risk in Tanzania: a sensitivity analysis. *J Epidemiol Community Health*, pages jech–2017, 2017.

[30] G. B. Huang and E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pages 14–003, 2014.

[31] L. O. Jimenez and D. Landgrebe. Projection pursuit for high dimensional feature reduction: Parallel and sequential approaches. In *1995 International Geoscience and Remote Sensing Symposium, IGARSS'95. Quantitative Remote Sensing for Science and Applications*, volume 1, pages 148–150. IEEE, 1995.

[32] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[33] X. Kang, B. Zhuo, and P. Duan. Semi-supervised deep learning for hyperspectral image classification. *Remote Sensing Letters*, 10(4):353–362, 2019.

[34] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4(03):173, 2013.

[35] Q. Ke and T. Kanade. Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.

[36] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, 12(1):103–108, 1990.

[37] P. V. Krishna, M. R. Babu, and E. Ariwa. *Global Trends in Information Systems and Software Applications: 4th International Conference, ObCom 2011, Vellore, TN, India, December 9-11, 2011, Part II. Proceedings*, volume 270. Springer, 2012.

[38] T.-W. Lee. Independent component analysis. In *Independent component analysis*, pages 27–66. Springer, 1998.

[39] M. Lennon, G. Mercier, M. Mouchot, and L. Hubert-Moy. Independent component analysis as a tool for the dimensionality reduction and the representation of hyperspectral images. In *Geoscience and Remote Sensing Symposium, 2001. IGARSS'01. IEEE 2001 International*, volume 6, pages 2893–2895. IEEE, 2001.

[40] H. Lin and A. Zhang. Summarization of hyperspectral image visualization methods. In *2014 IEEE Int. Conf. Prog. Info. Comp.*, pages 355–358, May 2014.

[41] C. Lomont. Introduction to Intel Advanced Vector Extensions. *Intel white paper*, 23, 2011.

[42] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *Int. J. Remote Sens.*, 28(5):823–870, 2007.

[43] A. L. Machidon, C. B. Ciobanu, O. M. Machidon, and P. L. Ogrutan. On Parallelizing Geometrical PCA Approximation. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019.

[44] A. L. Machidon, R. Coliban, O. Machidon, and M. Ivanovici. Maximum Distance-based PCA Approximation for Hyperspectral Image Analysis and Visualization. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4, 2018.

[45] A. L. Machidon, F. Del Frate, M. Picchiani, O. M. Machidon, and P. L. Ogrutan. Geometrical Approximated Principal Component Analysis for Hyperspectral Image Analysis. *Remote Sensing*, 12(11), 2020.

[46] A. L. Machidon, M. Ivanovici, R. Coliban, and F. Del Frate. A Geometrical Approximation of PCA for Hyperspectral Data Dimensionality Reduction. In *The ESA Earth Observation Phi-week EO Open Science and FutureEO*. ESA, 2018.

[47] A. L. Machidon, O. M. Machidon, C. B. Ciobanu, and P. L. Ogrutan. Accelerating a Geometrical Approximated PCA Algorithm Using AVX2 and CUDA. *Remote Sensing*, 12(12):1918, 2020.

[48] A. L. Machidon, O. M. Machidon, and P. L. Ogrutan. Face Recognition Using Eigenfaces, Geometrical PCA Approximation and Neural Networks. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pages 80–83, 2019.

[49] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[50] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790, 2004.

[51] Y. Mori, M. Kuroda, and N. Makino. *Nonlinear principal component analysis and its applications*. Springer, 2016.

[52] A. Norko. Simple image classification using principal component analysis (PCA). *GMU Volgenau School of Engineering, Fairfax, VA, USA*, 9, 2015.

[53] P. L. Ogrutan, A. L. Machidon, and A. Dinu. Is There a Link Between Creativity and Multiculturalism in Education? *TEM Journal*, 8(2):577, 2019.

[54] A. Peleg and U. Weiser. MMX technology extension to the Intel architecture. *IEEE micro*, 16(4):42–50, 1996.

[55] D. Peña, F. J. Prieto, and J. Viladomat. Eigenvectors of a kurtosis matrix as interesting directions to reveal cluster structure. *Journal of Multivariate Analysis*, 101(9):1995–2007, 2010.

[56] S. A. Priyanka, Y.-K. Wang, and S.-Y. Huang. Low-light Image Enhancement by Principal Component Analysis. *IEEE Access*, 2018.

[57] A. A. Qahtan, B. Alharbi, S. Wang, and X. Zhang. A PCA-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944. ACM, 2015.

[58] S. K. Raman, V. Pentkovski, and J. Keshava. Implementing streaming SIMD extensions on the Pentium III processor. *IEEE micro*, 20(4):47–57, 2000.

[59] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Science*, 62(2):115–122, 2002.

[60] T. Roughgarden and G. Valiant. CS168: The Modern Algorithmic Toolbox Lecture# 7: Understanding and Using Principal Component Analysis (PCA). 2016.

[61] M. Scholz, F. Kaplan, C. L. Guy, J. Kopka, and J. Selbig. Non-linear PCA: a missing data approach. *Bioinformatics*, 21(20):3887–3895, 2005.

[62] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[63] C. O. S. Sorzano, J. Vargas, and A. P. Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014.

[64] L. van der Maaten, E. O. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. 2009.

[65] L. Wang, Z.-H. You, X. Yan, S.-X. Xia, F. Liu, L.-P. Li, W. Zhang, and Y. Zhou. Using two-dimensional principal component analysis and rotation forest for prediction of protein-protein interactions. *Scientific reports*, 8(1):1–10, 2018.

[66] M.-H. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. In *Image processing, 2000. proceedings. 2000 international conference on*, volume 1, pages 37–40. IEEE, 2000.

[67] S. Zhang and M. Turk. Eigenfaces. *Scholarpedia*, 3(9):4244, 2008.

ANEXA A

Abstract

The PhD thesis entitled "Algorithms and statistical methods for multidimensional data analysis" presents the design, implementation, testing, validation and optimization of a novel dimensionality reduction method, related to the Principal Component Analysis (PCA) technique, namely the gaPCA method (Geometrical Approximated Principal Component Analysis), and its applications in several domains like Remote Sensing and Face Recognition. The thesis is structured in six chapters, in addition to the Introduction. In the first chapter, the current state of the art on dimensionality reduction is presented, with focus on the Principal Component Analysis method: advantages and disadvantages, applications in various domains and adaptations. Chapter 2 introduces the gaPCA novel method based on a geometrical construction as an alternative to the canonical Principal Component Analysis. Chapter 3 presents the validation of gaPCA in the field of hyperspectral images for the purposes of image analysis and classification. The performance of the gaPCA method was evaluated using several metrics, for both image quality assessment, quality of the reconstruction, redundancy of the information, and accuracy of the classification and the results were compared with the ones from the canonical PCA. Chapter 4 describes the results of the gaPCA method in the field of face recognition and its performance in terms of accuracy of the recognition, with the canonical PCA as a benchmark. Chapter 5 presents the gaPCA implementations using parallel computing principles on different hardware architectures, for accelerating the computation time to obtain speed-ups and improved energy efficiency. Finally, Chapter 6 summarizes the conclusions of the previous chapters and reviews the original contributions made in this doctoral thesis research. It also presents several future research directions for extending the existing contributions.

Rezumat

Teza de doctorat intitulată "Algoritmi și metode statistice pentru analiza datelor multidimensionale" prezintă conceperea, proiectarea, implementarea, testarea, validarea și optimizarea unei noi metode de reducere a dimensionalității, inspirată de Principal Component Analysis (PCA), și anume metoda gaPCA (Geometrical Approximated Principal Component Analysis) și aplicațiile sale în diverse domenii precum Remote Sensing și recunoașterea facială. Teza este structurată pe șase capitole, pe lângă Introducere. În primul capitol este prezentat stadiul actual privitor la reducerea dimensionalității, cu accent pe PCA: avantaje și dezavantaje, aplicații în diferite domenii și diverse implementări. Capitolul 2 introduce metoda inovativă bazată pe o construcție geometrică gaPCA, ca alternativă la PCA. Capitolul 3 prezintă validarea gaPCA în domeniul imaginilor hiperspectrale cu accent pe analiza imaginilor și clasificare. Performanța metodei gaPCA a fost studiată prin evaluarea calității imaginii, calității reconstrucției, redundanța informației și acuratețea clasificării, iar rezultatele au fost comparate cu cele ale PCA. Capitolul 4 descrie rezultatele metodei gaPCA în domeniul recunoașterii faciale în ceea ce privește acuratețea recunoașterii, având ca reper rezultatele obținute cu metoda PCA. Capitolul 5 prezintă implementările gaPCA utilizând principii de calcul paralel pe arhitecturi hardware diferite, pentru accelerarea timpului de calcul și îmbunătățirea eficienței energetice. În cele din urmă, capitolul 6 rezumă concluziile capitolelor anterioare și trece în revistă contribuțiile originale realizate în această cercetare doctorală. Sunt prezentate de asemenea mai multe direcții viitoare de cercetare pentru extinderea contribuțiilor existente.