

# Présentation PFE : Analyse des composantes principales (ACP)

Etudes théorique et pratique

HANFAOUI Karim<sup>1</sup>

<sup>1</sup>FSBM — 3<sup>e</sup> année Licence SMA  
karim.hanfaoui@etu.univh2c.ma

Faculté des sciences de Ben M'Sick

Mars → Juin 2025

# Remerciements

*Chers membres du jury,*

Je tiens à vous remercier sincèrement pour votre présence et votre attention lors de ma soutenance de projet de fin d'étude. Cet aboutissement n'aurait pas été possible sans votre soutien et vos conseils tout au long de cette aventure.

Je souhaite également remercier Pr. ADNAOUI Khalid pour sa collaboration et son engagement dans ce projet, ainsi que pour sa patience à mon égard. *Nous avons travaillé ensemble avec passion et détermination pour offrir un travail de qualité.*

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

# Introduction Générale

L'Analyse des Composantes Principales (ACP) est une méthode statistique de réduction de dimensionnalité qui transforme des variables corrélées en un ensemble de composantes orthogonales maximisant la variance expliquée. Fondée sur l'algèbre linéaire et la théorie des matrices, elle permet d'extraire l'information essentielle tout en préservant la structure des données.

# Problématique

## Problématique

Comment peut-on réduire la dimension d'un ensemble de données représenté sous forme de matrice tout en conservant le maximum possible d'information ?

# Objectifs du PFE

La problématique se reformule ainsi : réduire la dimension de données volumineuses tout en conservant l'information clé (sans perte significative). Ce projet vise à :

- Maîtriser les fondements mathématiques et algorithmiques de l'ACP classique.
- Implémenter et expérimenter l'ACP sur des jeux de données réels (images).
- Évaluer la qualité de reconstruction obtenue à l'aide de métriques telles que le PSNR (rapport signal/bruit de crête) et le SSIM (indice de similarité structurelle) en fonction du nombre de composantes.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

## Covariance : définition

La covariance entre deux variables  $X$  et  $Y$  mesure comment ces variables varient ensemble. Elle est définie par la formule :

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

où  $X_i$  et  $Y_i$  sont les observations des variables  $X$  et  $Y$ , et  $\bar{X}$  et  $\bar{Y}$  sont leurs moyennes respectives.

La matrice de covariance  $C$  est définie par ses éléments :

$$C_{ij} = \text{Cov}(X_i, X_j) = \frac{1}{n} \sum_{k=1}^n (X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j)$$

# Multiplicateurs de Lagrange : rappel (I)

Soit un problème d'optimisation sous contrainte :

$$\max_{x \in \mathbb{R}^n} f(x)$$

sous la contrainte  $g(x) = 0$  ,

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  et  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  sont des fonctions suffisamment régulières.  
L'idée de la méthode consiste à introduire un multiplicateur de Lagrange  $\lambda \in \mathbb{R}$  pour incorporer la contrainte :

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x) ,$$

appelé le *Lagrangien* du problème.

## Multiplicateurs de Lagrange : rappel (II)

Les conditions d'optimalité (nécessaires, sous hypothèse de régularité) sont :

$$\begin{cases} \nabla_x \mathcal{L}(x, \lambda) = 0, \\ g(x) = 0. \end{cases}$$

Autrement dit, au point optimal, le gradient de la fonction objectif doit être colinéaire au gradient de la contrainte ; le multiplicateur  $\lambda$  ajuste ce facteur de proportionnalité.

# Décomposition en valeurs singulières

La SVD factorise toute matrice réelle  $X$  en composantes orthogonales et valeurs singulières.

## Définition (SVD)

Pour toute matrice  $X_{m \times n}$ , il existe des matrices orthogonales  $U_{m \times m}$  et  $V_{n \times n}$  et une matrice diagonale  $\Sigma_{m \times n}$ ,  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$  avec  $\sigma_1 \geq \dots \geq \sigma_r \geq 0$ , telles que

$$X = U \Sigma V^\top.$$

## Corollaire : calcul de $X^\top X$

La SVD donne immédiatement

$$X^\top X = V \Sigma^2 V^\top = V \text{diag}(\sigma_1^2, \dots, \sigma_r^2) V^\top,$$

ce qui offre les valeurs propres de  $X^\top X$  sans calcul supplémentaire (utile pour l'ACP).

# Méthode de la puissance (Power Iteration)

- ➊ Choisir un vecteur initial aléatoire non nul  $v^{(0)}$ .
- ➋ Pour  $k = 1, 2, \dots$  jusqu'à convergence, calculer

$$v^{(k)} = \frac{\sum v^{(k-1)}}{\|\sum v^{(k-1)}\|}.$$

- ➌ Lorsque  $\|v^{(k)} - v^{(k-1)}\| < \epsilon$ , estimer la plus grande valeur propre par

$$\lambda_{\max} \approx v^{(k)^\top} \Sigma v^{(k)}.$$

# Déflation

- Une fois  $\lambda_1$  (la plus grande valeur propre) et son vecteur propre  $v_1$  calculés, soustraire leur contribution :

$$\Sigma^{(1)} = \Sigma - \lambda_1 v_1 v_1^\top.$$

- Appliquer à nouveau la méthode de la puissance sur  $\Sigma^{(1)}$  afin d'extraire  $\lambda_2$ ,  $v_2$ , et ainsi de suite pour obtenir tout le spectre de  $\Sigma$ .

# Algorithme QR

- ➊ Initialiser  $A^{(0)} = \Sigma$  et une matrice orthonormale  $Q^{(0)} = I$ .
- ➋ Pour  $k = 1, 2, \dots$  :
  - ➌ Décomposer  $A^{(k-1)} = Q^{(k)}R^{(k)}$ .
  - ➍ Mettre à jour  $A^{(k)} = R^{(k)}Q^{(k)}$ .
- ➎ Lorsque  $A^{(k)}$  converge vers une matrice quasi diagonale, les éléments diagonaux de  $A^{(k)}$  approchent les valeurs propres de  $\Sigma$  et les colonnes de  $Q^{(k)}$  fournissent les vecteurs propres correspondants.

# Méthode de Jacobi (pour matrices symétriques)

- Choisir un couple d'indices  $(i, j)$  ( $i \neq j$ ) tel que l'élément hors-diagonale  $\Sigma_{ij}$  soit le plus grand en valeur absolue.
- Calculer l'angle  $\theta$  de rotation qui annule cet élément, et former la matrice de rotation  $J(i, j, \theta)$ .
- Mettre à jour la matrice :  $\Sigma \leftarrow J^\top \Sigma J$ .
- Répéter ces étapes jusqu'à ce que tous les éléments hors-diagonale de  $\Sigma$  soient inférieurs à un seuil fixé.
- La matrice finale sera quasi-diagonale : ses éléments diagonaux représentent les valeurs propres, et les rotations successives accumulées fournissent les vecteurs propres associés.

# Résumé des concepts clés

- **Covariance** : comprendre comment les variables sont corrélées entre elles.
- **Transformation linéaire** : réduire la dépendance entre les variables.
- **Diagonalisation** : identifier les directions principales dans les données via les vecteurs propres.
- **SVD** : utiliser cette décomposition pour réaliser une réduction dimensionnelle.
- **Centrage des données** : préparer les données avant l'ACP.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

# Matrice de covariance des données

- Données centrées :  $X \in \mathbb{R}^{n \times p}$ , chaque colonne a une moyenne nulle.
- La **covariance** entre deux variables  $j$  et  $k$  est

$$\text{Cov}(x_j, x_k) = \frac{1}{n-1} \sum_{i=1}^n x_{ij} x_{ik}.$$

- En notation matricielle, la matrice de covariance complète est

$$\boxed{\Sigma = \frac{1}{n-1} X^\top X} \quad (\Sigma \in \mathbb{R}^{p \times p}).$$

- Ainsi,  $\Sigma_{jk}$  mesure la co-variation entre les colonnes  $j$  et  $k$  de  $X$ .

# Variance projetée et problème d'optimisation

- Choix d'une direction  $w \in \mathbb{R}^p$  : projection des observations sur l'axe  $w$  :

$$z = Xw.$$

- Comme  $X$  est centré, la variance de  $z$  est

$$\text{Var}(z) = w^\top \Sigma w, \quad \Sigma = \frac{1}{n-1} X^\top X.$$

## Objectif

Trouver la direction la plus « informative » :

$$\max_{w \in \mathbb{R}^p} w^\top \Sigma w \quad \text{s.c. } \|w\| = 1.$$

- Pourquoi  $\|w\| = 1$  ?** Sans contrainte, on pourrait choisir  $w = c\tilde{w}$  avec  $c \rightarrow \infty$ . Alors  $\text{Var}(Xw) = c^2 \tilde{w}^\top \Sigma \tilde{w} \rightarrow \infty$ . La normalisation fixe l'échelle et fait dépendre le critère uniquement de l'*orientation* de  $w$ .

## Première composante principale : application des multiplicateurs

Dans le cadre de l'ACP, on cherche un vecteur unitaire  $w$  qui maximise la variance projetée :

$$\max_{w \in \mathbb{R}^p} w^\top \Sigma w \quad \text{sous la contrainte} \quad w^\top w = 1,$$

où  $\Sigma$  est la matrice de covariance centrée des données. On pose donc :

$$f(w) = w^\top \Sigma w, \quad g(w) = w^\top w - 1,$$

et l'on construit le Lagrangien :

$$\mathcal{L}(w, \lambda) = w^\top \Sigma w - \lambda (w^\top w - 1).$$

Les conditions d'optimalité donnent :

$$\nabla_w \mathcal{L}(w, \lambda) = 0 \implies 2 \Sigma w - 2 \lambda w = 0 \iff \Sigma w = \lambda w,$$

ce qui correspond à l'équation aux valeurs propres de  $\Sigma$ . On en conclut que  $w$  est l'autovecteur associé à la plus grande valeur propre de  $\Sigma$ .

*Remarque : La contrainte  $w^\top w = 1$  garantit l'unicité (à un signe près) de la solution et empêche de faire croître indéfiniment  $w$  pour augmenter  $f(w)$*

# Propriétés clés de l'ACP

## Origine mathématique

- *Orthogonalité* :  $\Sigma$  est symétrique ; par le théorème spectral, ses vecteurs propres  $w_1, \dots, w_p$  forment une base orthonormée de  $\mathbb{R}^P$ .
- *Variance totale conservée* :  $\sum_{i=1}^p \lambda_i = \text{tr}(\Sigma)$ .
- *Proportion de variance expliquée* pour la  $k$ -ième composante :  
$$\text{PVE}_k = \lambda_k / \sum_i \lambda_i.$$

## Applications directes

- *Sélection de composantes* : choisir  $K$  tel que  $\sum_{i=1}^K \lambda_i / \sum_j \lambda_j$  dépasse un seuil (p. ex. 90 %).
- *Réduction de dimension* : projeter  $X$  sur  $w_1, \dots, w_K$  tout en conservant l'essentiel de la variance.
- *Décorrélation* : les composantes principales  $z_i = Xw_i$  sont non corrélées, simplifiant régression, classification ou compression.

## Algorithme : Calcul des composantes principales

- ① Centrer les données :  $X_c = X - \bar{X}$ .
- ② Calculer la matrice de covariance :  $\Sigma = \frac{1}{n-1} X_c^\top X_c$ .
- ③ Calculer les valeurs propres  $\lambda_i$  et vecteurs propres  $w_i$  de  $\Sigma$ .
- ④ Trier les vecteurs propres par ordre décroissant de valeurs propres.
- ⑤ Projeter les données :  $Y = X_c W$  où  $W = [w_1 \cdots w_k]$ .

# Choix du nombre de composantes principales

- **Critère du coude** : identifier visuellement le point d'infexion dans le scree plot (courbe des valeurs propres).
- **Variance expliquée cumulée** : calculer la part de variance expliquée par les  $k$  premières composantes :

$$\text{Ratio}_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i}.$$

Par exemple, si  $\lambda_1 = 3$ ,  $\lambda_2 = 2$ ,  $\lambda_3 = 1$ ,  $\lambda_4 = 0.5$  (variance totale 6.5), alors pour  $k = 3$  :

$$\text{Ratio}_3 = \frac{3 + 2 + 1}{6.5} \approx 0.92 \quad (92\%).$$

On choisirait  $k = 3$  si l'objectif est d'expliquer au moins 80–90% de la variance totale.

- **Critère de Kaiser** : pour des données standardisées (variance de 1 par variable), on recommande de retenir les composantes ayant  $\lambda_i > 1$  (variance supérieure à celle d'une variable initiale). Par exemple, si  $\lambda_1 = 2.5$ ,  $\lambda_2 = 1.2$ ,  $\lambda_3 = 0.7$ ,  $\lambda_4 = 0.6$ , on ne conserverait que les deux premières composantes ( $\lambda_1$  et  $\lambda_2 > 1$ ).

# Interprétation géométrique de l'ACP

- Aligne le premier axe avec la direction de variance maximale.
- Aligne chaque axe suivant orthogonalement, en maximisant la variance résiduelle.
- Préserve les distances entre les points.

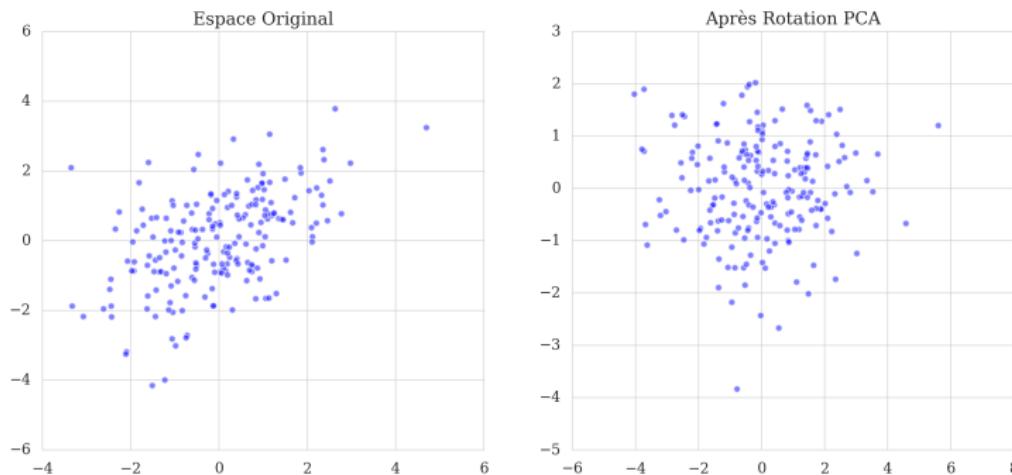


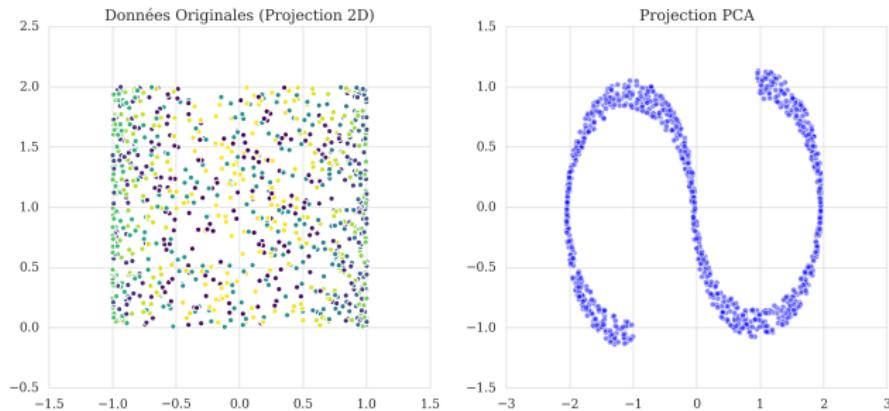
Figure – L'ACP peut être vue comme une rotation rigide de l'espace des données.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

## Limitation : Linéarité

L'ACP, de par sa nature linéaire, ne peut capturer que des relations linéaires entre les variables.



**Figure –** Données non linéaires pour lesquelles l'ACP échoue à restituer la structure (elle ne peut « dérouler » la spirale).

*Exemple : considérez des données disposées en spirale en 2D. L'ACP ne pourra pas dérouler cette spirale car elle se limite aux transformations linéaires.*

## Limitation : Sensibilité aux valeurs aberrantes

L'ACP classique est très sensible aux *outliers*, car elle se base sur la variance qui est elle-même sensible aux valeurs extrêmes.

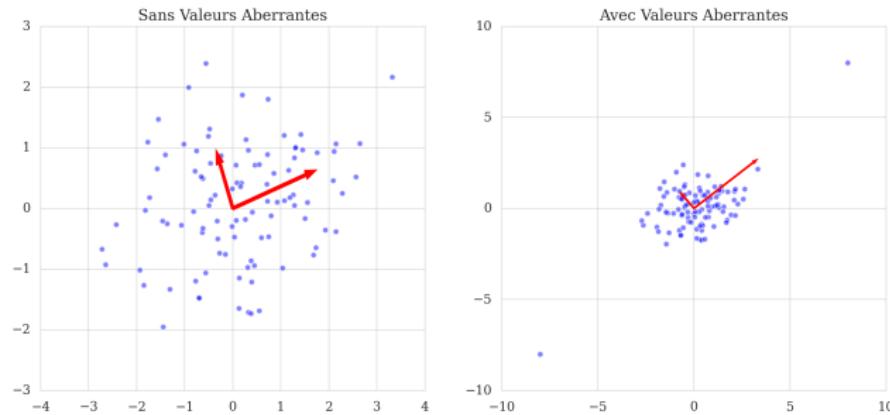


Figure – Impact de valeurs aberrantes (points rouges) sur les composantes principales. Un seul *outlier* peut fortement faire pivoter les axes principaux.

Exemple : dans un jeu de données financières, un seul point aberrant (p. ex. un crash boursier) peut complètement modifier l'orientation des composantes principales.

## Limitation : Hypothèse de normalité

L'ACP suppose implicitement que les données suivent une distribution normale multivariée, ce qui n'est pas toujours le cas en pratique.

*Exemple : pour des données de consommation électrique, les distributions sont souvent asymétriques avec des pics aux heures de pointe, violant l'hypothèse de normalité.*

## Limitation : Perte d'interprétabilité

Les composantes principales sont des combinaisons linéaires de toutes les variables initiales, ce qui peut rendre leur interprétation difficile.

*Exemple : dans une étude médicale avec 100 variables mesurées, chaque composante principale peut impliquer toutes ces variables avec des coefficients non nuls, rendant l'interprétation clinique complexe.*

## Solutions alternatives

- **Kernel PCA** : introduit des transformations non linéaires (noyaux) pour capturer des relations complexes.
- **Robust PCA** : intègre des méthodes robustes pour réduire l'influence des valeurs aberrantes.
- **Sparse PCA** : impose de la parcimonie pour obtenir des composantes plus facilement interprétables (quelques variables contributrices principales).
- **Probabilistic PCA** : formule l'ACP dans un cadre probabiliste pour tenir compte de l'incertitude (suppose un modèle statistique sous-jacent).

# Résumé des limitations et solutions

Limitation	Impact	Solution
Linéarité	Relations non linéaires mal captées	Kernel PCA
Outliers	Distorsion des composantes	Robust PCA
Interprétabilité	Difficile d'analyse	Sparse PCA
Normalité	Biais si données non gaussiennes	Probabilistic PCA

*NB : Comprendre ces limitations est essentiel pour choisir la variante d'ACP appropriée selon le contexte et la nature des données.*

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

# Matériel et méthodes

- **Données** : une image de test haute résolution ( $512 \times 512$  pixels, en niveaux de gris) issue d'une banque libre, et la base de visages ORL (400 images  $64 \times 64$ , en niveaux de gris).
- **Prétraitement** : centrage systématique des données (moyenne nulle par variable). Conversion des images en niveaux de gris et vectorisation. Normalisation supplémentaire (variance unitaire ou mise à l'échelle [0, 1]) appliquée si nécessaire pour homogénéiser l'échelle.
- **Outils** : langage Python (v3.x) avec bibliothèques NumPy (calcul matriciel, SVD), SciPy/*scikit-learn* (algorithmes d'ACP), Matplotlib (visualisation), *scikit-image* (lecture d'images, calcul de SSIM).
- **Environnement** : Notebooks Jupyter (distribution Anaconda) sous Windows 10.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP**
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

# Réduction de dimension et qualité visuelle (I)

L'ACP permet de projeter une image haute définition dans un espace latent de dimension beaucoup plus faible tout en préservant l'essentiel de l'information perceptuelle.

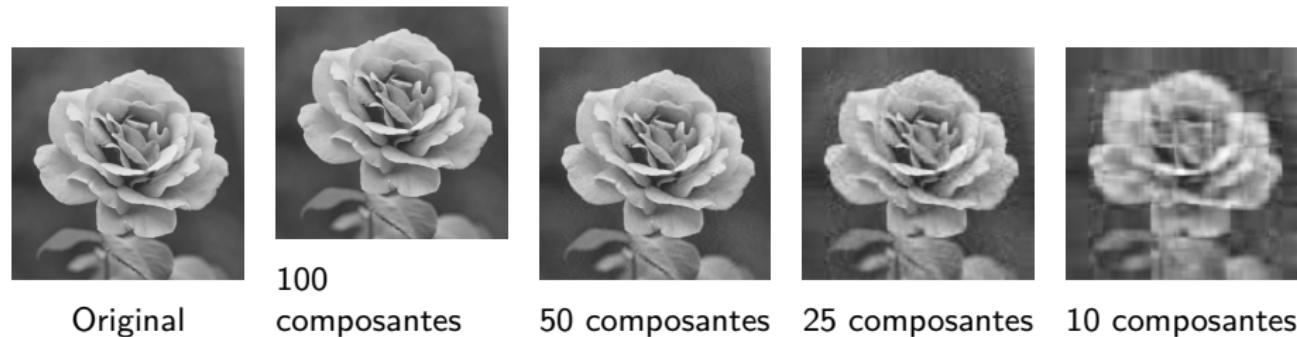


Figure – Effet du nombre de composantes principales  $k$  sur la qualité visuelle.

On observe une dégradation graduelle : de l'image originale ( $n$  dimensions) à des reconstructions avec  $k = 100, 50, 25$  et  $10$  composantes principales.

## Réduction de dimension et qualité visuelle (II)

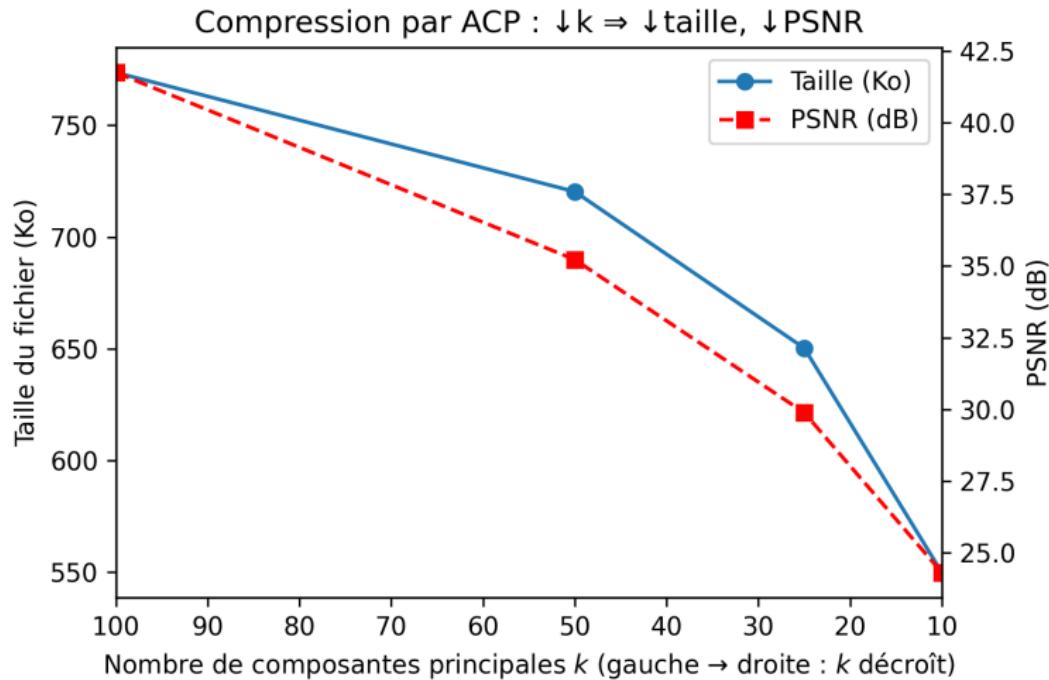


Figure – Relation entre le nombre de composantes  $k$ , la taille du fichier (Ko) et la qualité de reconstruction (PSNR en dB).

# Eigenfaces : Méthode de Reconnaissance Faciale

- On dispose d'un ensemble d'images de visages alignées et centrées.
- Chaque image  $I$  est convertie en vecteur de pixels, puis centrée :

$$I_{\text{centré}} = I - I_{\text{mean}}.$$

- L'ACP est appliquée à la matrice de données de dimension  $n \times p$  (images  $\times$  pixels).

# Rôle central de l'ACP dans Eigenfaces

- Soit un ensemble de  $n$  images vectorisées  $I_1, \dots, I_n \in \mathbb{R}^p$ , et leur moyenne

$$I_{\text{mean}} = \frac{1}{n} \sum_{i=1}^n I_i.$$

- Définissons la matrice de données centrée

$$X = [I_1 - I_{\text{mean}}, \dots, I_n - I_{\text{mean}}] \in \mathbb{R}^{p \times n}.$$

- Tout visage  $I$  s'encode alors par les coefficients

$$z = W^\top (I - I_{\text{mean}}) \in \mathbb{R}^K,$$

assurant :

- ▶ réduction de dimension :  $p \rightarrow K \ll p$ ,
- ▶ filtrage du bruit : on supprime les composantes de faible variance,
- ▶ extraction des traits dominants : chaque  $w_i$  capture une "direction" de variation.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

# SSIM : principe et application à l'ACP

## 1. Principe du SSIM

- **Objectif** : quantifier la similarité perçue entre deux images.
- Trois composantes locales (fenêtre  $11 \times 11$ ) :

$$\text{SSIM} = \underbrace{l(x, y)}_{\text{luminance}} \times \underbrace{c(x, y)}_{\text{contraste}} \times \underbrace{s(x, y)}_{\text{structure}}$$

avec

$$l = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} c = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

- $\text{SSIM} \in [-1, 1]$  : 1 identique, 0 aucune corrélation.

## 2. Comment l'utiliser avec l'ACP ?

### ① Compression PCA

- ▷ Réduire la dimension (images vectorisées).
- ▷ Reconstruire avec  $k$  composantes.

### ② Évaluation par SSIM

- ▷ Calculer  $\text{SSIM}(\text{img}_{\text{réelle}}, \text{img}_{\text{reconst}})$ .
- ▷ Tracer  $\text{SSIM}$  vs.  $k$  pour un compromis qualité/taille.

### ③ Sélection adaptative

Point où  $\text{SSIM} < 0.95$  pour fixer  $k$ .

### ④ Critère d'arrêt

Arrêter dès que  $\Delta \text{SSIM} < \varepsilon$ .

## Calcul de $\mu_x$ et $\mu_y$

- **Fenêtre** : généralement  $11 \times 11$  ou  $8 \times 8$ .
- **Filtre gaussien 2-D** ( $\sigma \simeq 1.5$ ) pour pondérer :

$$\mu_x(i, j) = \sum_{u,v} w(u, v) X(i+u, j+v), \quad \mu_y(i, j) = \sum_{u,v} w(u, v) Y(i+u, j+v)$$

- **Implémentation rapide** : convolution ou GaussianBlur.

Exemple Python (OpenCV) :

```
mu_x = cv2.GaussianBlur(X, (11, 11), 1.5)
mu_y = cv2.GaussianBlur(Y, (11, 11), 1.5)
```

# Constantes $C_1$ , $C_2$ , $C_3$

## Formules générales

$$C_1 = (K_1 L)^2, \quad C_2 = (K_2 L)^2, \quad C_3 = \frac{C_2}{2}$$

- $L$  : dynamique des intensités (8 bits  $\rightarrow$  255, flottant  $\rightarrow$  1).
- Paramètres conseillés :  $K_1 = 0.01$ ,  $K_2 = 0.03$ .
- **Images 8 bits** ( $L = 255$ ) :  $C_1 \approx 6.5025$ ,  $C_2 \approx 58.5225$ ,  $C_3 \approx 29.2612$ .
- **Images normalisées** ( $L = 1$ ) :  $C_1 = 1 \times 10^{-4}$ ,  $C_2 = 9 \times 10^{-4}$ ,  
 $C_3 = 4.5 \times 10^{-4}$ .

## À retenir

$\mu_x$ ,  $\mu_y$  : moyennes locales via un filtre gaussien.

$C_1$ ,  $C_2$ ,  $C_3$  : dérivés de la dynamique  $L$  et de deux petits coefficients  $K_1$ ,  $K_2$  pour stabiliser la division.

Toujours ajuster ces constantes à l'échelle choisie ([0, 255] ou [0, 1]).

# Évaluation quantitative

$k$	MSE	PSNR (dB)	SSIM
100	20	35.1	0.98
50	65	30.2	0.95
25	200	25.1	0.90
10	650	20.0	0.80

Métriques de reconstruction de l'image test en fonction du nombre de composantes principales  $k$ .

## Analyse des résultats

- La courbe du PSNR en fonction de  $k$  présente un *point de coude* : au-delà d'environ 25 composantes, les gains supplémentaires de PSNR deviennent très limités alors que la taille des données double (rendements décroissants).
- Le SSIM (indice de similarité structurelle) corrobore le PSNR : il demeure élevé ( $\approx 0.95$ ) pour  $k \geq 50$ , puis chute drastiquement lorsque  $k$  est trop faible (p. ex. 0.80 à  $k = 10$ ), traduisant une perte des structures perceptibles.
- La reconstruction par ACP surpasse une compression « naïve » : en tant qu'approximation de rang- $k$  optimale minimisant l'erreur quadratique moyenne, l'ACP fournit un PSNR supérieur pour une même taille de données comparée à une réduction de résolution ou une sélection aléatoire de pixels.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

## Conclusion & Perspectives

- **Bilan :** La mise en œuvre de l'ACP sur des images a démontré qu'une réduction drastique du nombre de variables (jusqu'à 90%) est possible sans altération majeure de la qualité visuelle. Les objectifs du projet ont été atteints : compréhension théorique, implémentation pratique et évaluation quantitative (PSNR et SSIM) ont permis de valider l'approche sur des cas concrets.
- **Limites :** L'ACP classique reste sensible aux données aberrantes et ne capture que les relations linéaires. Elle peut conduire à une perte d'interprétabilité des composantes. Ces aspects n'ont pas pu être pleinement traités dans ce travail.
- **Perspectives :** Affiner le choix du nombre optimal de composantes (critère du coude, validation croisée) et explorer d'autres méthodes de réduction de dimension. En particulier, des approches non linéaires comme t-SNE ou UMAP offriraient des visualisations plus fidèles de structures complexes.

**Testez-vous même**

**GitHub**



Figure – Code QR du Repo GitHub

# Bibliographie I

-  Pearson, K. (1901). *L'analyse des données*. Journal de Statistique.
-  Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*.
-  Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.
-  Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
-  Schölkopf, B., et al. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*.

# Webographie

- <https://scikit-learn.org/stable/modules/decomposition.html#pca> – Documentation *scikit-learn* sur l'ACP.
- <https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis> – Discussion approfondie sur l'ACP.
- <https://setosa.io/ev/principal-component-analysis/> – Visualisation interactive de l'ACP.
- [https://www.cs.otago.ac.nz/cosc453/student\\_tutorials/principal\\_components.pdf](https://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf) – Tutoriel complet sur l'ACP.
- <https://www.kaggle.com/code/arthurtok/interactive-intro-to-dimensionality-reduction> – Exemples pratiques d'ACP.

## Webographie (suite)

- <https://www.datasciencecentral.com/principal-component-analysis-for-dimensionality-reduction/> – Applications en data science.
- <https://mathworld.wolfram.com/PrincipalComponentAnalysis.html> – Aspects mathématiques détaillés.
- <https://towardsdatascience.com/a-complete-guide-to-principal-component-analysis-pca-in-machine-learning-6f9f105399c7> – Guide complet sur l'ACP.
- [https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis) – Article Wikipedia détaillé.
- <https://www.youtube.com/watch?v=FgakZw6K1QQ> – Vidéo explicative sur l'ACP.

# Table of Contents

- 1 Introduction Générale
- 2 Préliminaires Mathématiques pour l'ACP
- 3 Analyse en Composantes Principales (ACP)
- 4 Limitations de l'ACP classique
- 5 Matériel et méthodes
- 6 Application pratique de l'ACP
- 7 Résultats et Analyses
- 8 Conclusion & Perspectives
- 9 Annexes

## Extrait de code Python – ACP

```
import imageio
import numpy as np
# Chargement de l'image en niveau de gris
img = imageio.imread('images/original.png', as_gray=True)
# Décomposition SVD de l'image en composantes principales
U, S, Vt = np.linalg.svd(img, full_matrices=False)
k = 50 # nombre de composantes retenues
# Reconstruction approchée de l'image avec k composantes
img_rec = U[:, :k] @ np.diag(S[:k]) @ Vt[:, :]
# Calcul de l'erreur quadratique moyenne (MSE) et du PSNR
mse = np.mean((img - img_rec)**2)
psnr = 10 * np.log10((255**2) / mse)
print(f"PSNR={psnr:.2f}dB")
```

# Paramètres d'exécution

- **Machine** : PC portable sous MacOS, CPU Intel Core i9, 32 Go de RAM.
- **Environnement** : Anaconda Python 3.10, exécution du code dans des scripts sur VsCode.
- **Bibliothèques** : NumPy 1.21, SciPy 1.7, scikit-learn 1.2, Matplotlib 3.4, scikit-image 0.18.
- **Temps de calcul** : de l'ordre de la seconde pour les cas étudiés (p. ex.  $\approx 0.2$  s pour une image  $512 \times 512$ ,  $\approx 1$  s pour 400 matrice  $64 \times 64$ , sur CPU).