



Rapport du projet d'analyse hilbertienne

Badr Sajid
Issam Habibi

ENSEEIHT — 2ème Année Sciences du Numérique - Parcours HPC Big Data
2020–2021

Contents

1	Introduction	2
2	Stratégie 1 : ACP classique	3
2.1	Classification des chiffres via ACP "classique"	3
2.2	Reconnaissance de chiffres et débruitage via ACP "classique"	4
2.3	Etude de la sensibilité des résultats	5
2.3.1	par rapports au choix de <i>PrecApprox</i>	5
2.3.2	par rapports au choix de variance du bruit	6
3	Stratégie 2 : généralisation aux RKHS	7
3.1	Principes	7
3.2	Classification des chiffres via Kernel ACP	8
3.2.1	Noyau linéaire	8
3.2.2	Noyau polynomial	8
3.2.3	Noyau Gaussien	9
3.3	Reconnaissance de chiffres et débruitage via Kernel ACP associé au Noyau Gaussien	10
3.4	Etude de la sensibilité des résultats	11
3.4.1	par rapports au choix de <i>PrecApprox</i>	11
3.4.2	par rapports au choix de variance du bruit	12
3.4.3	par rapports au choix de σ	13
3.4.4	par rapports au choix du nombre d'itérations	14

List of Figures

1	Classification des chiffres via ACP	3
2	Reconnaissance de chiffres via ACP pour une variance de bruit = 0.02 et une <i>PrecApprox</i> = 0.5	4
3	Reconnaissance de chiffres via ACP pour une <i>PrecApprox</i> = 0.8	5
4	Reconnaissance de chiffres via ACP pour une variance du bruit = 0.2	6
5	Classification des chiffres via Kernel ACP associé au Noyau lineaire	8
6	Classification des chiffres via Kernel ACP associé au Noyau polynomial avec $c = 3$ et $q = 2$	8
7	Classification des chiffres via Kernel ACP associé au Noyau Gaussien avec $\sigma = 6$	9
8	Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour <i>PrecApprox</i> = 0.8, variance de bruit = 0.02, $\sigma = 6$ et 4 itérations	10
9	Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour <i>PrecApprox</i> = 0.5	11
10	Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour une variance = 0.2	12
11	Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour $\sigma = 3$	13
12	Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour 10 itérations	14

1 Introduction

L'objectif est d'illustrer l'utilisation de l'analyse en composante principale et de sa variante RKHS pour la reconnaissance automatique de chiffres. Ce problème est rencontré, par exemple, par les services postaux - tri automatique du courrier depuis les codes postaux - ou dans le domaine bancaire - paiement par chèques.

Ce projet est réalisé sous Matlab et nous disposons d'une base de données supposées suffisantes pour effectuer l'apprentissage et donc la reconnaissance des chiffres. Nous utilisons dans notre cas, des images de 16 pixels par 16 pixels et représentant des chiffres. Chaque image est transformée en un vecteur de taille 256 et dont les composantes représentent le niveau de gris des pixels.

Le but de l'utilisation de l'ACP est de réduire la dimension du problème via la construction de sous-espaces représentatifs de chacune de ces classes qui facilitera donc la classification.

2 Stratégie 1 : ACP classique

2.1 Classification des chiffres via ACP "classique"

On trace la norme de la composante de l'image test dans l'orthogonal de l'espace engendré par les colonnes de chacune des matrices U_i , les bases ortho- normales de chacun des sous-espaces, cette norme est donnée par la relation suivante :

$$d_i = \frac{\|(I - U_i U_i^T)C\|_2}{\|C\|_2}, \quad \forall i \in \mathbb{N}_p. \quad (1)$$

En utilisant une variance de bruit de 0.02 et une PrecApprox de 0.5, on obtient le résultat suivant représentant le résultat de la classification des différents chiffres.

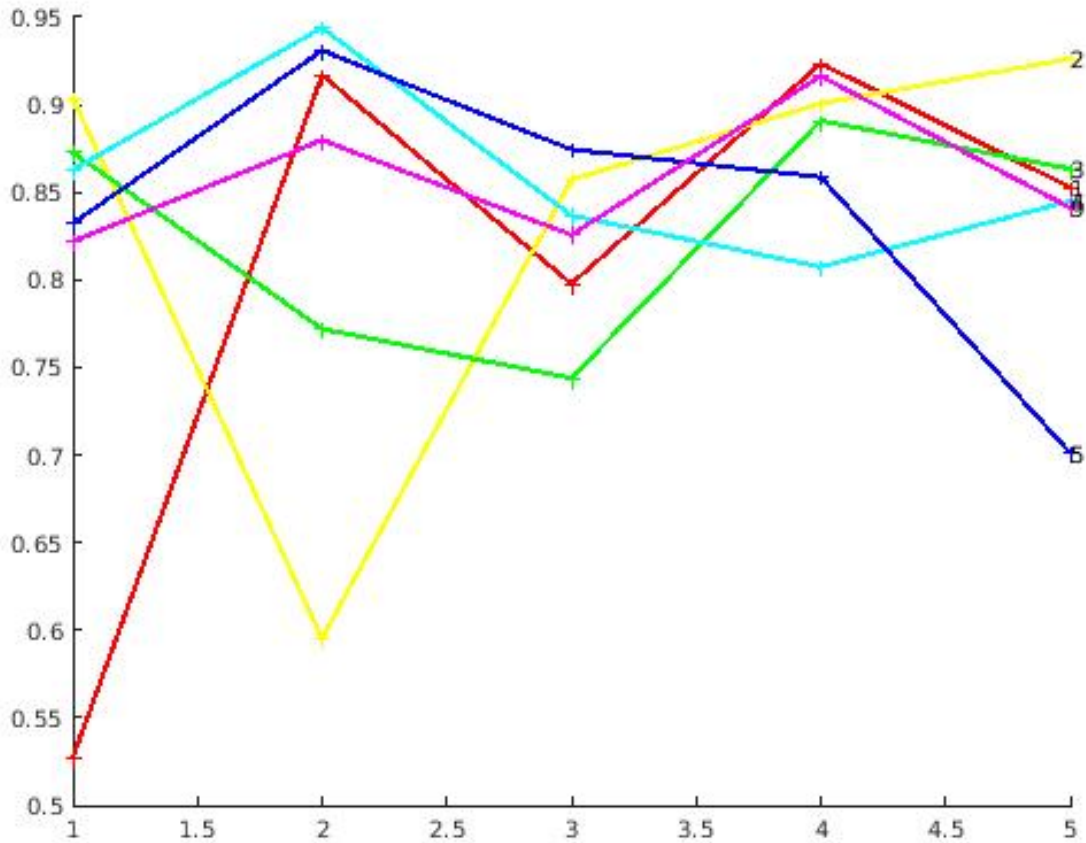


Figure 1: Classification des chiffres via ACP

2.2 Reconnaissance de chiffres et débruitage via ACP "classique"

Dans cette partie on reconstruit l'image à partir des composantes principales déterminées dans la première partie.

En utilisant une variance de bruit de 0.02 et une PrecApprox de 0.5, on obtient le résultat suivant représentant le résultat de la reconnaissance des différents chiffres.



Figure 2: Reconnaissance de chiffres via ACP pour une variance de bruit = 0.02 et une PrecApprox = 0.5

2.3 Etude de la sensibilité des résultats

2.3.1 par rapports au choix de *PrecApprox*

En utilisant cette fois une *PrecApprox* de 0.8, on obtient le résultat suivant représentant le résultat de la reconnaissance des différents chiffres.

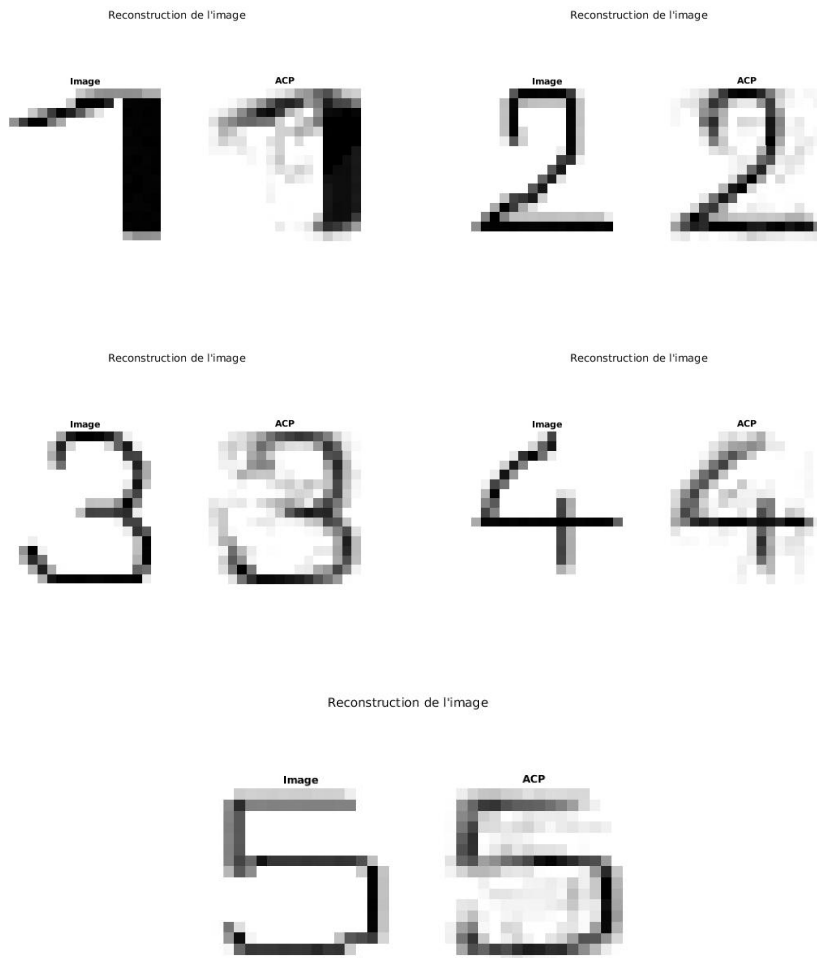


Figure 3: Reconnaissance de chiffres via ACP pour une *PrecApprox* = 0.8

On remarque que les images sont plus nettes que celle de pour une *PrecApprox* = 0.5. Cela est dû à l'utilisation de plus de données pour la reconnaissance. On conclue alors, que plus *PrecApprox* est hausse plus l'image est nette. Par contre pour une *PrecApprox* allant de 0.8 à 1 on a la même définition de l'image.

2.3.2 par rapports au choix de variance du bruit

En utilisant cette fois une variance du bruit de 0.2, on obtient le résultat suivant représentant le résultat de la reconnaissance des différents chiffres.



Figure 4: Reconnaissance de chiffres via ACP pour une variance du bruit = 0.2

On remarque que les images sont plutôt flou que celle de pour une variance du bruit = 0.02. Alors plus la variance du bruit est grande, moins bonne est la qualité de l'image reconstruite.

3 Stratégie 2 : généralisation aux RKHS

3.1 Principes

Le but de cette partie est d'appliquer le principe des noyaux reproduisant RKHS pour la reconnaissance des chiffres en utilisant toujours la méthode de l'analyse en composantes principale ACP.

À partir des données $(x_i)_{i \in [1, n]}$ et une feature ϕ , on construit la matrice K qu'on utilisera au lieu de la matrice variance-covariance est définie par :

$$K(i, j) = k(x_i, x_j) \quad (2)$$

avec :

$$k(x, y) = \langle \phi(x), \phi(y) \rangle \quad (3)$$

On construit ensuite $(u_i)_{i=1:n} \in \mathbb{R}^n$ une base orthonormée de vecteur propre de K , et les vecteurs $(\alpha_i)_{i=1:r}$ avec :

$$\alpha_i = \frac{1}{d_i} u_i, \quad \forall i = 1 : r \quad (4)$$

La reconstruction se fait alors par résolution d'un problème aux moindres carrés dans l'espace des features :

$$d_i^k = \frac{\|(\phi(x) - m_i) - \Pi_k(\phi(x) - m_i)\|^2}{\|(\phi(x) - m_i)\|^2} \quad (5)$$

Puisque $(\phi(x) - m_i)$ et $\Pi_k(\phi(x) - m_i)$ sont orthogonaux, alors :

$$d_i^k = 1 - \frac{\|\Pi_k(\phi(x) - m_i)\|^2}{\|(\phi(x) - m_i)\|^2} \quad (6)$$

On calcul séparément le numérateur et le dénominateur, on trouve :

$$\|\Pi_k(\phi(x) - m_i)\|^2 = \sum_{j=1}^k \left(\sum_{i=1}^n \alpha_i^j k(x, x_i) - \frac{1}{n} \sum_{h=1}^n k(x_h, x_i) \right)^2 \|v_j\|^2 \quad (7)$$

avec :

$$\|v_j\|^2 = \sum_{i=1}^n \sum_{l=1}^n \alpha_i^j \alpha_l^j k(x_i, x_l) \quad (8)$$

$$\|(\phi(x) - m_i)\|^2 = k(x, x) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j) - \frac{2}{n} \sum_{i=1}^n k(x, x_i) \quad (9)$$

3.2 Classification des chiffres via Kernel ACP

3.2.1 Noyau linéaire

On utilisant un noyau linéaire on retrouve le résultat suivant :

$$k(x, y) = x^T y \quad (10)$$

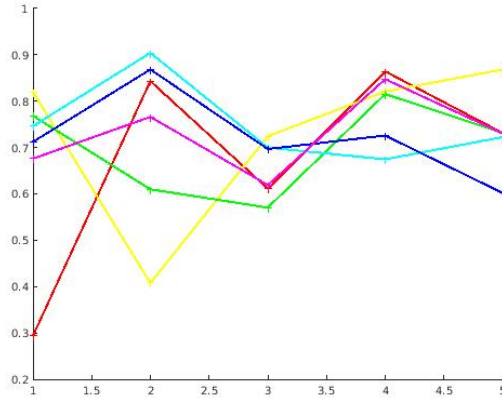


Figure 5: Classification des chiffres via Kernel ACP associé au Noyau linéaire

3.2.2 Noyau polynomial

On utilisant un noyau polynomial on retrouve le résultat suivant :

$$k(x, y) = (c + x^T y)^q \quad (11)$$

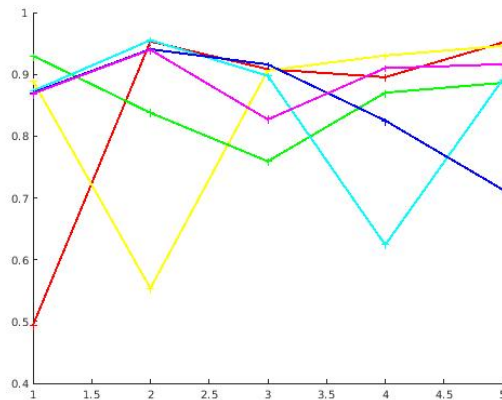


Figure 6: Classification des chiffres via Kernel ACP associé au Noyau polynomial avec $c = 3$ et $q = 2$

3.2.3 Noyau Gaussien

On utilisant un noyau Gaussien on retrouve le résultat suivant :

$$k(x, y) = \exp\left(\frac{-1}{2\sigma^2} \|x - y\|^2\right) \quad (12)$$

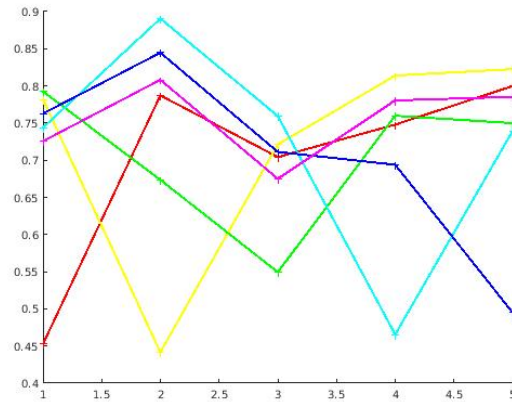


Figure 7: Classification des chiffres via Kernel ACP associé au Noyau Gaussien avec $\sigma = 6$

On remarque que les noyaux qu'en changeant le noyau on a un resultat différent en terme de précision par contre ils donne tous une classification pertinente.

De plus le noyau gaussien est le plus précis parmi les trois noyaux utilisés.

3.3 Reconnaissance de chiffres et débruitage via Kernel ACP associé au Noyau Gaussien

En utilisant une variance de bruit de 0.02, une PrecApprox de 0.8 et le Noyau Gaussien avec $\sigma = 6$, on obtient le résultat suivant représentant le résultat de la reconnaissance des différents chiffres la reconnaissance des chiffres via Kernel ACP associé à ce noyau.

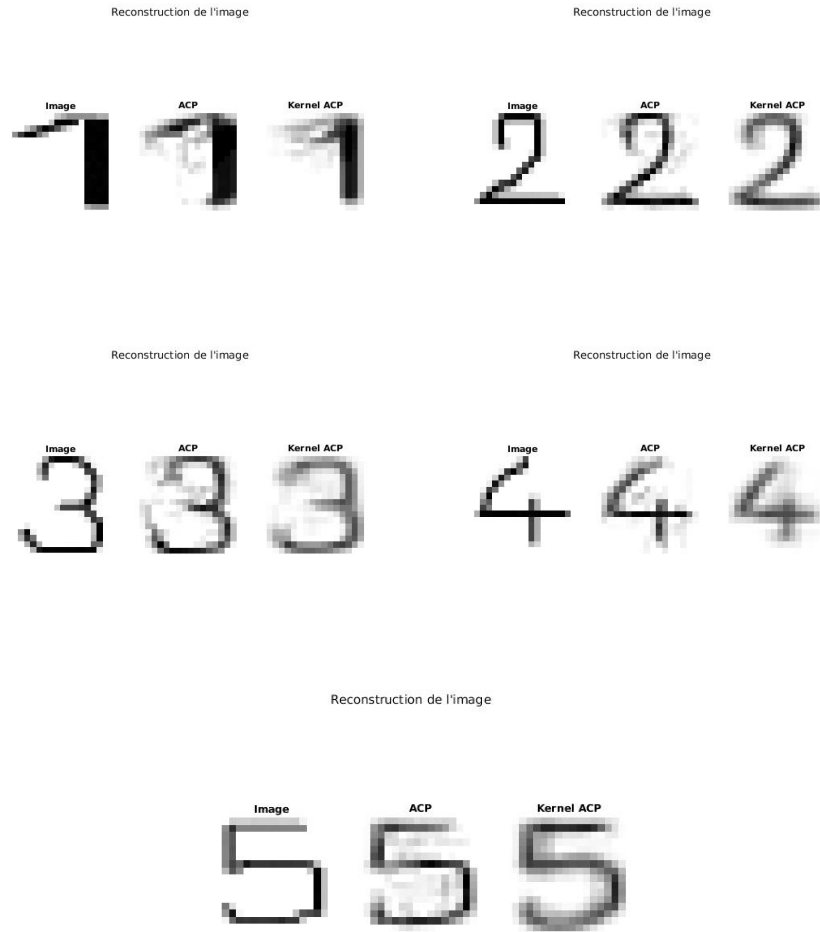


Figure 8: Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour PrecApprox = 0.8, variance de bruit = 0.02, $\sigma = 6$ et 4 itérations

3.4 Etude de la sensibilité des résultats

3.4.1 par rapports au choix de *PrecApprox*

En utilisant cette fois une *PrecApprox* de 0.5 et en fixant les autres paramètres, on obtient le résultat suivant :

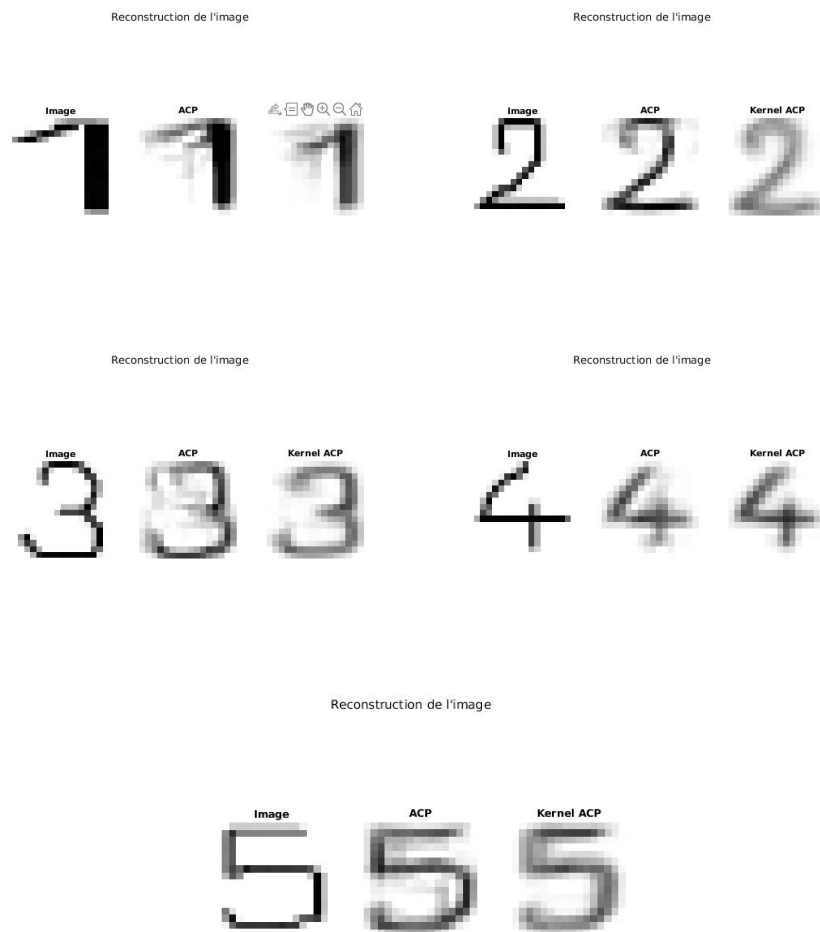


Figure 9: Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour $PrecApprox = 0.5$

On remarque que les images sont moins nettes que celle de pour une $PrecApprox = 0.8$. Cela est dû à l'utilisation de plus de données pour la reconnaissance. La précision est donc importante pour avoir des chiffres plus pertinents.

3.4.2 par rapports au choix de variance du bruit

En utilisant cette fois une variance du bruit de 0.2 et en fixant les autres paramètres, on obtient le résultat suivant :

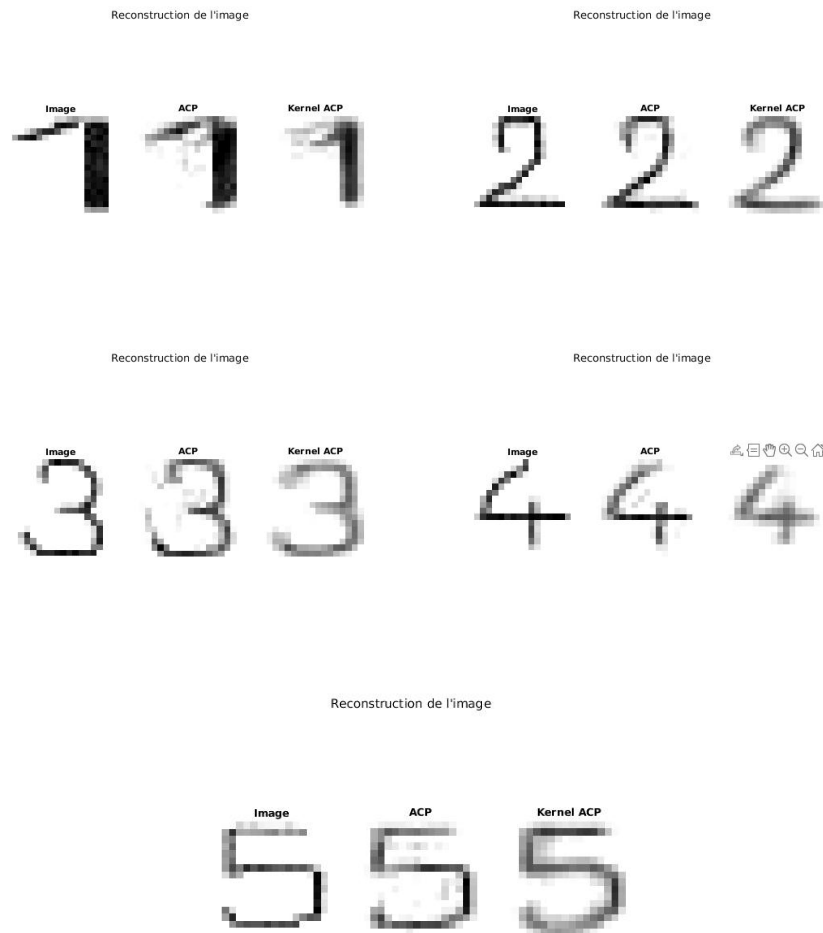


Figure 10: Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour une variance = 0.2

Comme dans la première partie on remarque que les image sont de plus en plus flous en utilisant un bruit de puissance élevé.

3.4.3 par rapports au choix de σ

En utilisant cette fois $\sigma = 3$ et en fixant les autres paramètres, on obtient le résultat suivant :

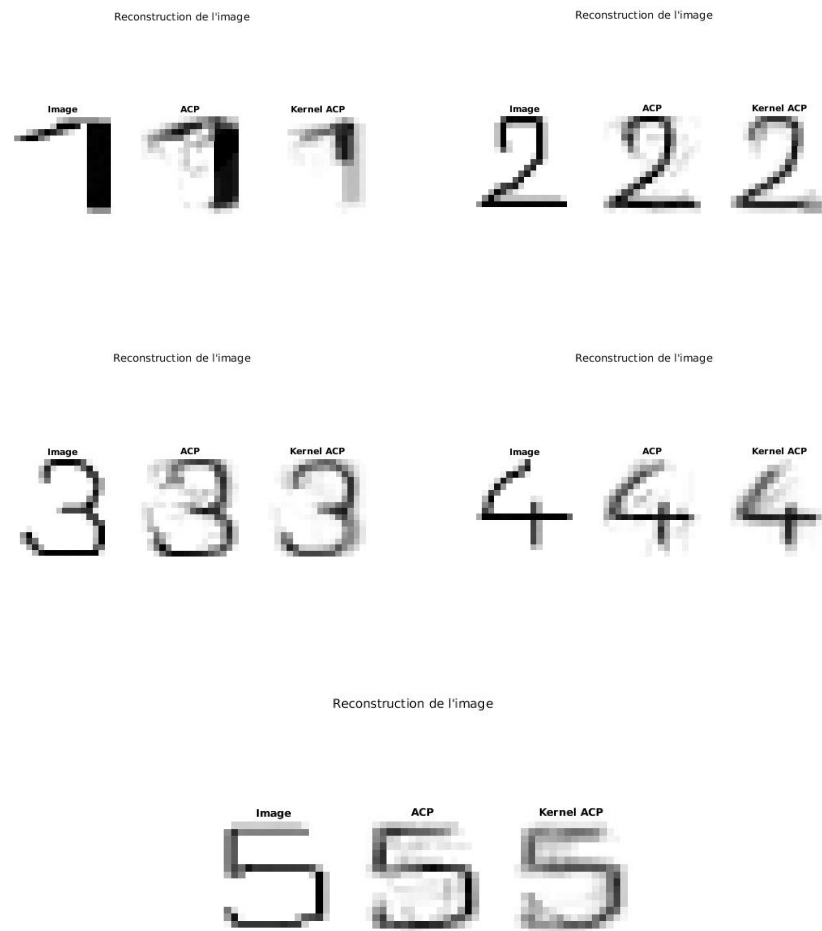


Figure 11: Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour $\sigma = 3$

3.4.4 par rapports au choix du nombre d'itérations

En utilisant cette fois 10 d'itérations et en fixant les autres paramètres, on obtient le résultat suivant :

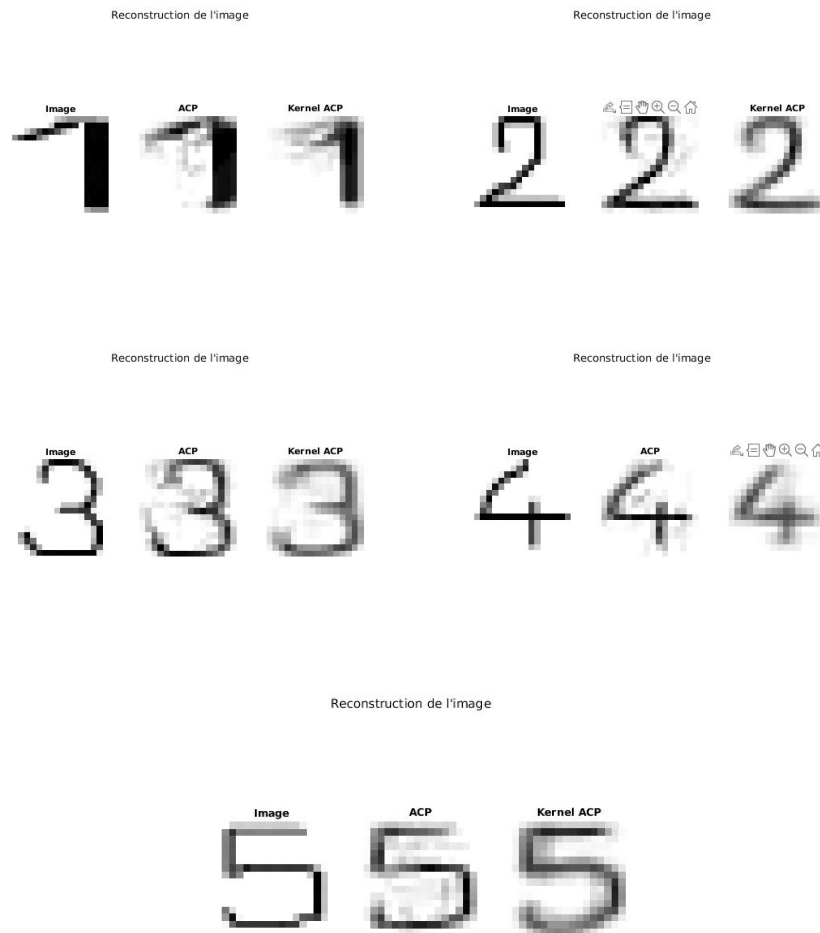


Figure 12: Reconnaissance de chiffres via Kernel ACP associé au Noyau Gaussien pour 10 itérations

Le nombre d'itérations rend l'exécution plus lente surtout si le nombre de classes est bien élevé, mais ce nombre permet de converger vers un point fixe qui représente l'image reconnu.