



PS204  
TÉLÉCOMMUNICATIONS  
RAPPORT

---

## Introduction à l'apprentissage statistique

---

***Etudiants :***

Cyril HANNIER  
Romain BENQUET  
Nicolas BLANC

***Encadrant :***

Pascal VALLET

1 Avril 2022

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Principes de l'ACP</b>	<b>2</b>
2.1	Question 1 . . . . .	2
2.2	Question 2 . . . . .	3
2.3	Question 3 . . . . .	3
2.4	Question 4 . . . . .	3
2.5	Question 5 . . . . .	4
2.6	Question 6 . . . . .	4
<b>3</b>	<b>Réduction de dimension et eigenfaces</b>	<b>4</b>
3.1	N eigfaces issues de Training1 . . . . .	4
3.2	6 sujets de la base après reconstruction avec ACP . . . . .	5
3.3	Variation de la dimension l du facespace . . . . .	5
3.4	Ratio de reconstruction . . . . .	6
<b>4</b>	<b>Classification</b>	<b>6</b>
4.1	Classificateur K-NN . . . . .	6
4.1.1	Principe . . . . .	6
4.1.2	Implémentation . . . . .	7
4.1.3	Matrice de confusion . . . . .	7
4.2	Classificateur Gaussien . . . . .	8
4.2.1	Principe . . . . .	8
4.2.2	Implémentation . . . . .	9
4.2.3	Matrice de confusion . . . . .	9
4.2.4	Domination des composantes principales : . . . . .	9
4.2.5	Calcul matrice gaussienne différentes : . . . . .	10
4.3	Comparaison estimateur gaussien et K-NN . . . . .	10
<b>5</b>	<b>Conclusion</b>	<b>11</b>

## 1 Introduction

L'objectif de ce TP est d'introduire une notion d'intelligence artificielle à travers des méthodes statistiques. Le but est de se concentrer sur la reconnaissance de visage. Nous avons implémenté des algorithmes qui à partir d'une banque de visage d'une même personne, associe le visage type qui lui correspond. Nous avons étudié l'ACP qui consiste en une réduction de la dimension des données de départ afin d'avoir des images dites "réduites" pour ainsi se limiter uniquement sur leurs composantes principales. Suite à cela, nous avons travaillé sur deux types de classifieurs : Le classifieur KNN et le classifieur Gaussien.

## 2 Principes de l'ACP

L'ACP consiste à projeter les données  $(x_1, x_2, \dots, x_n)$  dans les  $l$  directions orthogonales qui contiennent le plus d'énergie au sens statistique.

### 2.1 Question 1

$$\begin{aligned} E_1 &= \frac{1}{n} \sum_{i=1}^n \langle u_1, x_i - \bar{x} \rangle^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(u_1^T (x_i - \bar{x}))]^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(u_1^T (x_i - \bar{x})) (x_i - \bar{x})^T u_1] \\ &= u_1^T \hat{R} u_1 \end{aligned}$$

On a alors un problème d'optimisation :  $\begin{cases} \max_{u_1} u_1^T \hat{R} u_1 \\ \|u_1\|^2 = 1 \end{cases}$

On note  $L(u_1, \lambda_1) = u_1^T \hat{R} u_1 - \lambda_1 (u_1^T u_1 - 1)$

$$\nabla L(u_1, \lambda_1) = 0$$

$$2 \hat{R} u_1 - 2\lambda_1 u_1 = 0$$

$$\hat{R} u_1 = \lambda_1 u_1$$

donc  $u_1$  est le vecteur propre de  $\hat{R}$  associé à la valeur propre  $\lambda_1$

$$\text{Ainsi, } E_1 = u_1^T \hat{R} u_1 = u_1^T \lambda_1 u_1 = \lambda_1$$

$$\text{or, } \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$$

On en déduit finalement que  $u_1$  est vecteur propre de  $\hat{R}$  maximisant l'énergie de projection.

## 2.2 Question 2

$$\begin{cases} \max_{u_2} u_2^T \hat{R} u_2 \\ \|u_2\|^2 = 1 \\ \langle u_1, u_2 \rangle = u_2^T u_1 \end{cases}$$

$$2(u_2, \lambda_2, \mu) = u_2^T \hat{R} u_2 - \lambda_2(u_2^T u_2 - 1) - \mu u_2^T u_1$$

$$\nabla L(u_2, \lambda_2, \mu) = 0$$

$$(1) \Rightarrow 2\hat{R}u_2 - 2\lambda_2 u_2 - \mu u_1 = 0$$

$$\Rightarrow 2u_1^T \hat{R} u_2 - 2\lambda_2 u_1^T u_2 - \mu u_1^T u_1 = 0$$

$$\Rightarrow 2u_1^T \hat{R} u_2 - \mu = 0$$

$$\Rightarrow 2\lambda_1 u_1^T u_2 - \mu = 0$$

$$\Rightarrow \mu = 0$$

$$(1) \Rightarrow \hat{R} u_2 = \lambda_2 u_2$$

$\Rightarrow u_2$  est le vecteur propre de  $\hat{R}$  associé à la valeur propre  $\lambda_2$  qui maximise l'énergie de projection.

$$\Rightarrow E_2 = \lambda_2$$

Plus généralement,  $\forall j \in \llbracket 1, l \rrbracket, E_j = \lambda_j$

## 2.3 Question 3

Le ratio de l'énergie de projection est donné par :

$$\begin{aligned} K(l) &= \frac{\frac{1}{n} \sum_{i=1}^n \|\pi_j(x_i - \bar{x})\|_2^2}{\frac{1}{n} \sum_{i=1}^n \|x_i - \bar{x}\|_2^2} \\ &= \frac{\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^l \langle u_j, x_i - \bar{x} \rangle^2}{\frac{1}{n} \sum_{i=1}^n \|x_i - \bar{x}\|_2^2} \\ &= \frac{\sum_{j=1}^l \lambda_j}{\sum_{i=1}^p \lambda_i} \end{aligned}$$

## 2.4 Question 4

Montrons que  $\text{rg}(X) \leq n-1$  :

$$X = \frac{1}{\sqrt{n}} [x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_n - \bar{x}] \text{ avec } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$X = \left[ x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_{n-1} - \bar{x}, \sum_{i=1}^n (x_i - \bar{x}) \right]$$

On somme toutes les colonnes sur la dernière colonne.

$$X = \begin{bmatrix} x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_{n-1} - \bar{x}, \underbrace{\sum_{i=1}^n (x_i - \frac{n}{n} \sum_{i=1}^n x_i)}_{=0} \end{bmatrix}$$

Donc  $\text{rg}(X) \leq n-1$ .

D'après l'énoncé,  $x_1, \dots, x_n$  sont linéairement indépendants donc on ne peut pas obtenir d'autres combinaisons linéaires entre  $x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_{n-1} - \bar{x}$ .

D'où  $\text{rg}(X) = n-1$ .

## 2.5 Question 5

D'après l'énoncé,  $v$  est vecteur propre de  $X^T X$ , ce qui signifie qu'il existe une valeur propre  $\lambda$  telle que :

$$\begin{aligned} X^T X v &= \lambda v \\ \Leftrightarrow X^T u &= \lambda v \\ \Leftrightarrow X X^T u &= \lambda X v \\ \Leftrightarrow X X^T u &= \lambda u \end{aligned}$$

Donc  $u$  est un vecteur propre de  $X X^T$  associé à la même valeur propre  $\lambda$ .

## 2.6 Question 6

On pose :

$$U = X V (V^T X^T X V)^{-1/2} = [u_1, \dots, u_l]$$

On a alors :

$$U^T U = [(V^T X^T X V)^{-1/2}]^T V^T X^T X V (V^T X^T X V)^{-1/2} \quad (1)$$

$$= [(V^T X^T X V)^{-1/2}]^T (V^T X^T X V)^{-1/2} \quad (2)$$

$$= [(V^T X^T X V)^T]^{-1/2} (V^T X^T X V)^{-1/2} \quad (3)$$

$$= (V^T X^T X V)^{-1/2} (V^T X^T X V)^{-1/2} \quad (4)$$

$$= I_n \quad (5)$$

On sait que  $U^T U[i, j] = \langle u_i, u_j \rangle$  donc on a les vecteurs  $u_1, \dots, u_l$  forment une base orthonormée c'est à dire que  $U$  est une matrice à colonne orthonormées. La question 5 donne ensuite que les  $u_k$  sont des vecteurs propres de  $R$ .

## 3 Réduction de dimension et eigenfaces

### 3.1 N eigfaces issues de Training1

Les  $N$  eigfaces de la première base d'apprentissage sont obtenues en effectuant un reshape afin de convertir les vecteurs colonnes en une image de taille  $192 \times 168$  :

Images issues de la base Training 1



### 3.2 6 sujets de la base après reconstruction avec ACP

Nous avons déterminé la dimension du sous espace vectoriel optimal afin d'avoir un ratio de reconstruction supérieur à 0.9. Après implémentation, il s'avère que cette dimension optimale est égale à 8. Nous obtenons pour ce  $l$  optimale, les 6 images de la base suivante :

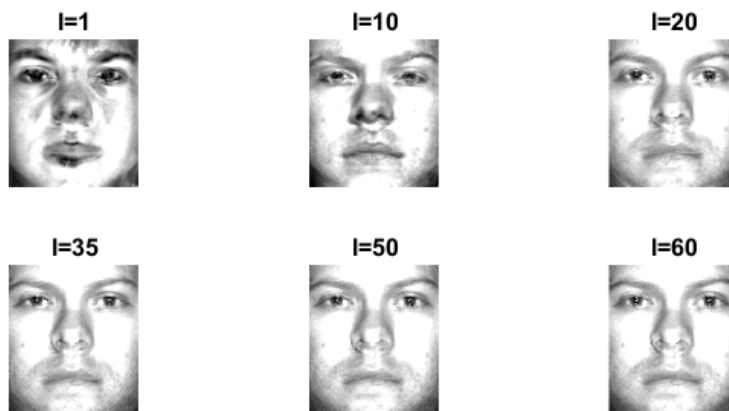
Differentes images de la base pour  $l=8$



### 3.3 Variation de la dimension $l$ du facespace

Après application du principe de l'ACP, nous obtenons des résultats différents en fonction du choix de  $l$  :

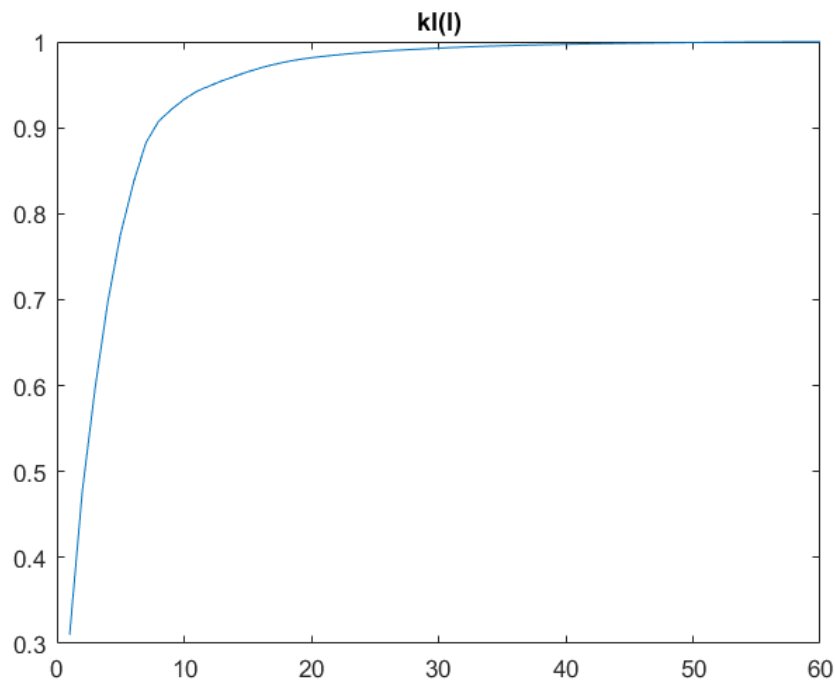
1 image de Training 1 pour different l



On remarque que plus le  $l$  est proche de  $n$  et plus l'affichage est de qualité.

### 3.4 Ratio de reconstruction

Le ratio de construction est déterminé à partir des  $n$  valeurs propres déterminées de  $XX^t$  s



## 4 Classification

### 4.1 Classificateur K-NN

#### 4.1.1 Principe

Tout d'abord, cette méthode repose sur le calcul du vecteur  $w(x)$ ,  $\forall x \in (x_1, x_2 \dots x_n)$

$$w(x) = \begin{pmatrix} \langle x - \bar{x}, u_1 \rangle \\ \vdots \\ \langle x - \bar{x}, u_l \rangle \end{pmatrix} \quad (6)$$

Après avoir calculé ce vecteur  $w(x)$  pour chaque élément de la base d'apprentissage, il faut calculer la norme suivante :

$$V(x) = \underset{(i_1, i_2, \dots, i_k) \subset [1, n]}{\operatorname{argmin}} \sum_{j=1}^k \|w(x) - w(x_{i_j})\| \quad (7)$$

Ainsi, on détermine les  $k$  indices correspondant aux plus petites différences en terme de norme entre le  $w(x)$  issue de la base de test et le  $w_{i_j}$  issue de la base d'apprentissage. Ainsi on repère quels sont les  $k$  images de la base d'apprentissage qui se rapprochent le plus de l'image test. Parmi ces  $k$  voisins déterminés, on attribue la classe à la l'image test comme la classe la plus présente parmi les  $k$  voisins.

#### 4.1.2 Implémentation

Dans un premier temps, nous avons commencé par calculer l'ensemble des vecteurs  $w(x_i)$  que nous avons placé dans une matrice. L'implémentation de  $V(x)$  a consisté à calculer la norme de la différence entre un vecteur  $w(x_i)$  et l'ensemble des autres vecteurs  $w$  de la base d'apprentissage. Pour identifier les  $k$  plus proches voisins, il a fallu déterminer l'ensemble des  $k$  minimums. Enfin, pour avoir la classe du vecteur, on regarde la classe qui revenait le plus parmi l'ensemble des  $k$  voisins.

Le choix de  $k$  doit suivre un compromis. En effet, il ne doit pas être trop élevé car si il est supérieur au nombre d'élément qui appartient à la même classe que le vecteur testé, il pourrait rendre erroné l'apprentissage. De même, si on prend un  $k$  trop petit, on pourrait perdre en précision.

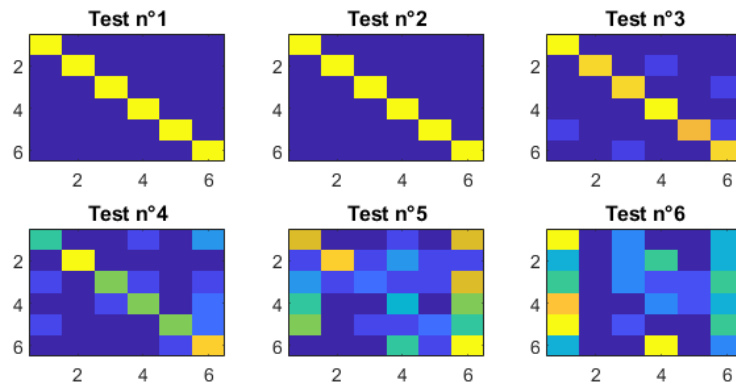
Après l'apprentissage, on passe à la phase de Test dans laquelle on va reprendre les vecteurs  $w$  issues de la phase d'apprentissage et refaire le même processus que précédemment.

#### 4.1.3 Matrice de confusion

Les matrices de confusion de l'ensemble des 6 bases de données sont obtenues à l'aide d'une fonction matlab fournie.



### Matrice de confusion méthode KNN



On remarque que les résultats sont plutôt satisfaisants même si la qualité de la classification diminue nettement à partir de la base 4. Cette diminution est due à la qualité de l'éclairage qui diminue également comme le montre les images suivantes.

Les images issues de la base 1 sont les suivantes :

Représentation de la matrice de data pour chaque colonne



Les images issues de la base 6 sont les suivantes :

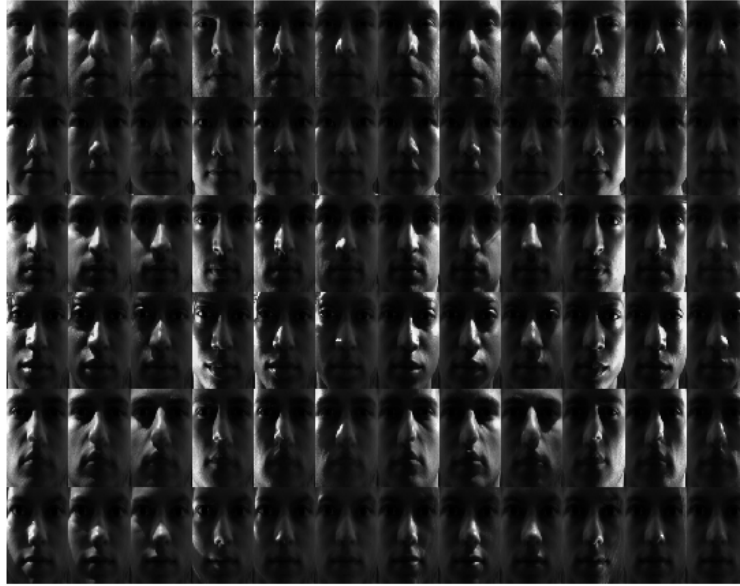
## 4.2 Classificateur Gaussien

### 4.2.1 Principe

Durant la phase d'apprentissage, nous calculons la moyenne pour chaque classe ainsi que la matrice de covariance afin de les utiliser dans la phase de test. Les formules utilisés sont les suivantes :

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} w(x_i) \quad (8)$$

Representation de la matrice de data pour chaque colonne



$$\bar{\Sigma} = \frac{1}{n} \sum_{j=1}^m \sum_{i \in C_j} (w(x_i) - \bar{\mu}_j)(w(x_i) - \bar{\mu}_j)^T \quad (9)$$

On considère ensuite le classifieur suivant par maximum de vraisemblance :

$$\phi_x = \operatorname{argmin}_{j \in [1, m]} \left\| \left( \sum \right)^{-\frac{1}{2}} (w(x) - \mu_j) \right\|_2^2 \quad (10)$$

#### 4.2.2 Implémentation

Pour implémenter cette méthode, nous avons déjà procédé au calcul de la moyenne pour chaque classe de la base d'apprentissage. Par la suite, nous avons procédé au calcul de la matrice de covariance à l'aide de la base d'apprentissage. Nous avons ensuite appliqué pour chaque élément de la base de test, le calcul de la norme donné par la formule donnée précédemment.

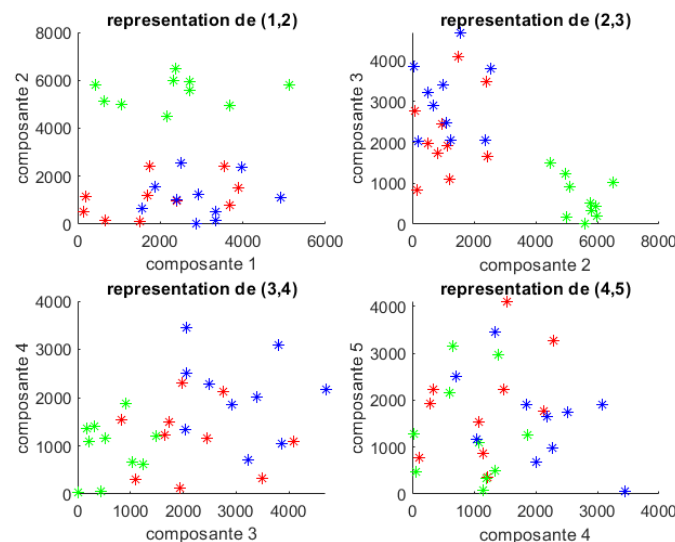
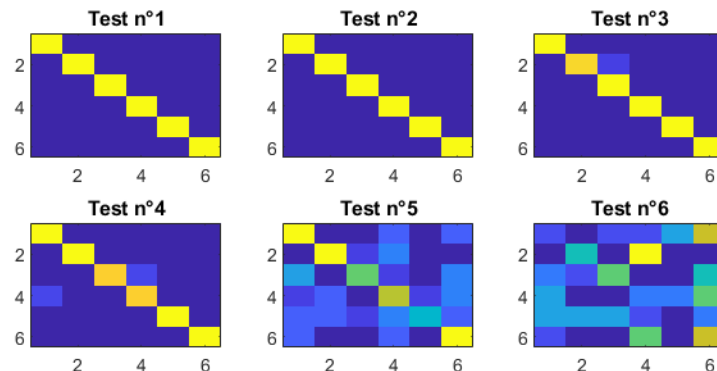
#### 4.2.3 Matrice de confusion

Les matrices de confusion pour les 6 bases de test sont données par la figure suivante. On remarque que les résultats sont satisfaisants. Ces résultats se dégradent surtout pour les 2 dernières bases de test. Ils se dégradent plus tard que pour le classifieur KNN.

#### 4.2.4 Domination des composantes principales :

La prochaine figure nous montrent que la projection sur les 1 premières composantes principales est pertinente. En effet, on remarque les images d'une même classe sont assez regroupés ce qui favorise leur classification par la suite.

### Matrice de confusion méthode Gaussienne



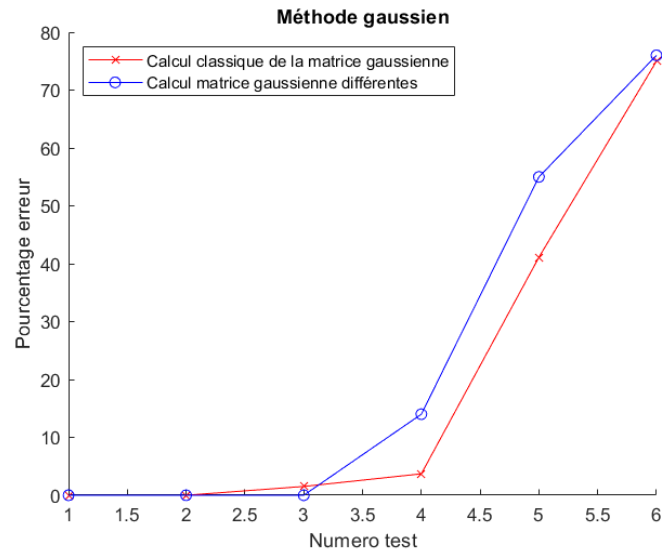
#### 4.2.5 Calcul matrice gaussienne différentes :

Nous avons calculé la matrice gaussienne d'une manière différente. En effet, au lieu de la calculer pour tous les éléments de la base d'apprentissage, nous avons calculé une matrice de covariance pour chaque classe. Les résultats visibles ci-dessous nous montrent que pour les premières bases de test, nous avons de meilleurs résultats alors qu'ils sont légèrement moins bons pour les autres bases de test.

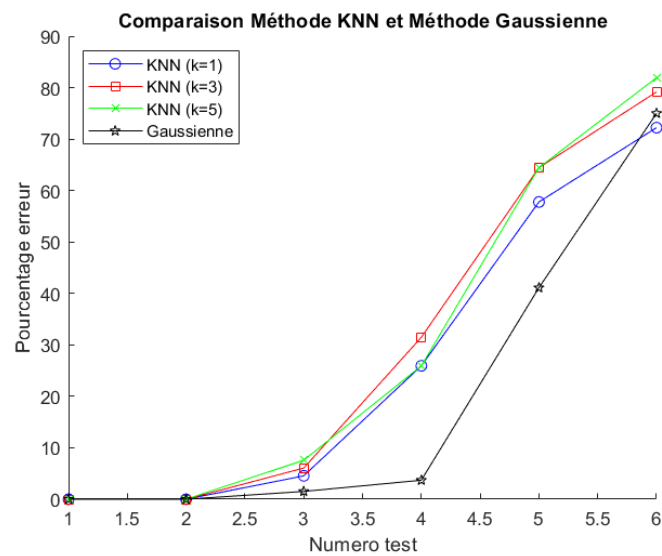
### 4.3 Comparaison estimateur gaussien et K-NN

Nous avons effectué une étude comparative afin de conclure sur l'efficacité de ces 2 classifieurs. A travers les 6 bases de tests, on peut facilement remarquer que le classifieur gaussien obtient de meilleurs résultats dans l'ensemble. Pour les 2 premières bases de test, les résultats sont semblables. Ensuite les écarts se creusent. En revanche, on remarque que pour la dernière base, les résultats semblent se valoir. Néanmoins, la qualité des images de la dernière base est tellement mauvaise, qu'il est difficile de dire que cela est représentatif.

Nous avons testé la méthode KNN avec différents  $k$ . On ne peut pas faire de conclusion pour des valeurs entre  $k=3$  et  $k=5$ . En revanche, on remarque que pour un  $k$  très faible, on a de



meilleures résultats.



## 5 Conclusion

A travers ce TP, nous avons constaté qu'il existait certaines méthodes statistiques pour faire de la reconnaissance faciale. Ces méthodes obtiennent des résultats intéressants même si ces résultats peuvent être parfois perturbés par une baisse de qualité des images fournies.