REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS –SETIF1-

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

Master-1-F3I

Rapport

de mini projet FDD

Reconnaissance de visages

Réalisé par :

- ✓ Aberkane Rania
- ✓ Gueddou Chaouki

Dirigé par :

- ✓ Moussaoui Abdelouahab

-2016/2017-

Table des matières

1. Introduction :

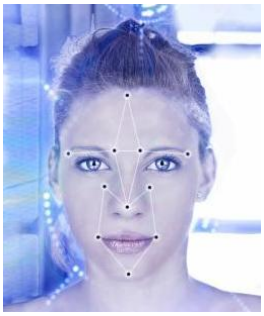
Identifier une personne à partir de son visage est une tâche aisée pour les humains. En est-il de même pour une machine ? Ceci définit la problématique de la reconnaissance automatique de visages, qui a engendré un grand nombre de travaux de recherche au cours des dernières années.

2. La reconnaissance de visage :

C'est une technologie très efficace qui est utilisée dans de nombreuses applications liées à la sécurité. Elle est par exemple un outil très fiable pour aider les forces de police à identifier des criminels, ou bien pour permettre aux services de douanes de vérifier l'identité des voyageurs. Actuellement, avec la numérisation des échanges, l'usage de cette technologie est en train de s'étendre au monde des entreprises. Utilisée dans des applications commerciales, la reconnaissance faciale permet par exemple de sécuriser des transactions en ligne. La reconnaissance faciale est sans contact et son utilisation ne nécessite aucun outil spécifique, ce qui en fait la solution idéale pour l'identification de personnes dans une foule ou dans des espaces publics.

3. Analyse biométrique des données du visage :

La **reconnaissance faciale** semble être la technologie biométrique la plus naturelle puisque nous nous reconnaissons en regardant notre visage. Les **systèmes de reconnaissance faciale** sont des **systèmes automatisés capables d'identifier des individus** en fonction des caractéristiques de leur visage telles que l'écartement des yeux, des arêtes du nez, des commissures des lèvres, des oreilles. menton, etc. Ces caractéristiques sont analysées puis comparées à une base de données existante afin d'**identifier une personne** ou de **vérifier son identité**.



4. Fonctionnement [1] :

Le principe de fonctionnement de base d'un système de reconnaissance faciale peut être résumé en **quatre étapes** : les deux premières s'effectuent *en amont* du système (**détection** et **normalisation** du visage) et les deux dernières représentent la *reconnaissance* à proprement dit (extraction et comparaison des caractéristiques).

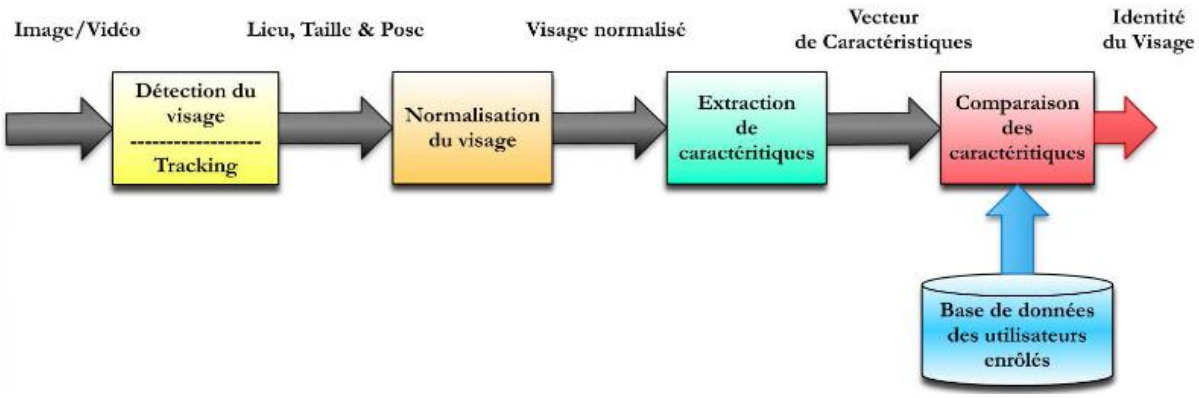


Figure 1: Principe de fonctionnement de base d'un système de reconnaissance faciale.

5. L'efficacité de la reconnaissance faciale dépend de plusieurs facteurs clés :

- **Qualité de l'image :** Les sujets coopérants sont ceux qui autorisent volontairement que leur image faciale soit capturée. Les sujets non coopérants sont ceux dont le visage est capturé par des caméras de surveillance ou par des témoins utilisant leur Smartphone.
- **Algorithmes d'identification** : le deuxième facteur de performance clé est la puissance des algorithmes utilisés pour déterminer des similitudes entre les traits des visages. Les algorithmes analysent la position relative, la taille et/ou la forme des yeux, du nez, des pommettes et de la mâchoire. Ces caractéristiques sont ensuite utilisées pour rechercher d'autres images présentant les mêmes caractéristiques.
- **Bases de données fiables** : enfin, la précision de la **reconnaissance faciale** dépend de la taille et de la qualité des bases de données utilisées : pour reconnaître un visage, on doit pouvoir le comparer à un autre. La difficulté est donc d'établir des points de correspondance entre la nouvelle image et l'image source, en d'autres termes, les photos d'individus connus. En conséquence, plus la base de données d'images ciblées est importante, plus il est possible d'y trouver des correspondances. [2]

6. Détection de visage :

L’efficacité des systèmes biométriques basés sur l’authentification de visage dépend essentiellement de la méthode utilisée pour localiser le visage dans l’image. Dans la littérature scientifique, le problème de localisation de visages est aussi désigné par la terminologie "détection de visages". Plusieurs travaux de recherches ont été effectués dans ce domaine. Ils ont donné lieu au développement d’une multitude de techniques allant de la simple détection du visage. [3]

Cependant, les solutions proposées jusqu’à maintenant sont loin d’être satisfaisantes car elles fonctionnent uniquement dans des environnements contrôlés, et par conséquent elles ne gèrent pas la variabilité des conditions d’acquisition de la vie quotidienne, notamment :

- **Variation de la pose**: où les images d'un visage changent en fonction de l’orientation de ce dernier.

Figure 2: Exemples de variation de poses.

- **La présence ou absence des composantes structurales** : les caractéristiques faciales tels que la barbe, la moustache, et les lunettes causent une grande variabilité des composantes structurales du visage, notamment au niveau de la forme, de la couleur, et de la taille.
- **Les occultations** : les visages peuvent être partiellement occultés par d'autres objets. En effet, dans une image contenant un groupe de personnes par exemple, des visages peuvent partiellement masquer d'autres visages.
- **Les conditions d'illumination** : des facteurs tels que l'éclairage (distribution de la source de lumière, son intensité, son spectre) et les caractéristiques de l'appareil photographique affectent l'aspect d'un visage dans l'image acquise.

7. Les méthodes de représentation de vissage :

Les systèmes de reconnaissance de visages sont très souvent classés à partir des conclusions d'études psychologiques sur la façon dont les hommes utilisent les caractéristiques faciales pour reconnaitre les autres. De ce point de vue, on distingue les trois catégories : les méthodes globales, les méthodes locales et les méthodes hybrides. [4]

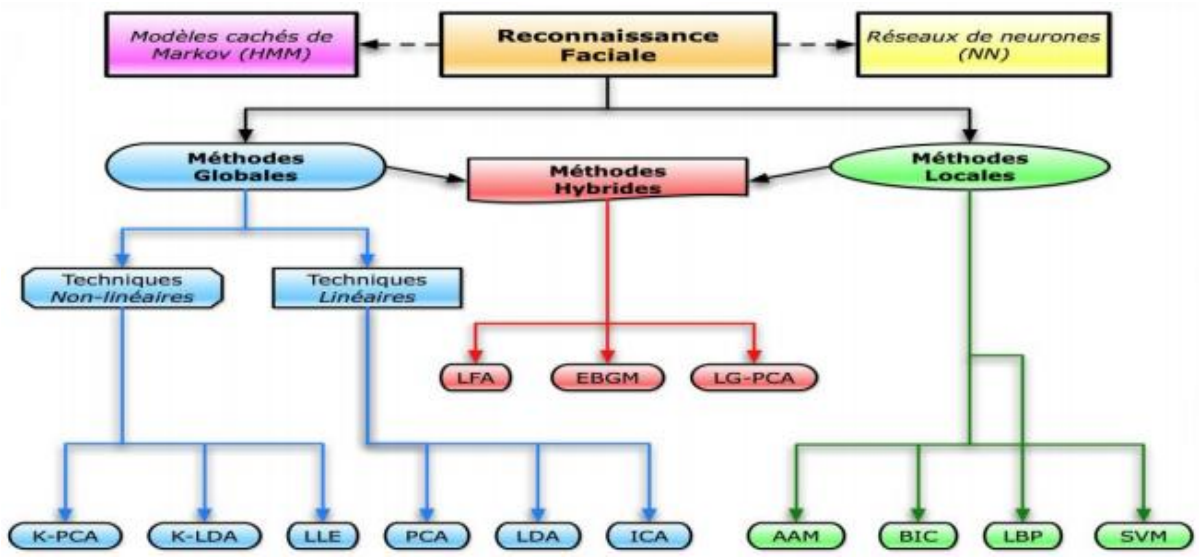


Figure 3: Classification des algorithmes principaux utilisés en reconnaissance faciale. [5]

7.1. Les méthodes globales :

Les méthodes globales basées sur des techniques d'analyse statistique bien connues. Dans ces méthodes, les images de visage (qui peuvent être vues comme des matrices de valeurs de pixels) sont utilisées comme entrée à l'algorithme de reconnaissance et sont généralement transformées en vecteurs, plus faciles à manipuler. L'avantage principal des méthodes globales est qu'elles sont relativement rapides à mettre en Suvre. En revanche, elles sont très sensibles aux variations d'éclairement, de pose et d'expression faciale. [6]

7.2. Les méthodes locales :

Les méthodes locales consistent à appliquer des transformations en des endroits spécifiques de l'image, le plus souvent autour de points caractéristiques (coins des yeux, de la bouche, le nez,...). Elles nécessitent donc une connaissance à priori sur les images. Ces méthodes sont plus difficiles à mettre en place mais sont plus robustes aux problèmes posés par les variations d'éclairement, de pose et d'expression faciale.

L'avantage des méthodes locales, est qu'elles prennent en compte la particularité du visage en tant que forme naturelle à reconnaître et un nombre réduit de paramètres en exploitant les résultats de la recherche en neuropsychologie et psychologie cognitive sur le système visuel humain.

La difficulté éprouvée c'est quand il s'agit de prendre en considération plusieurs vues du visage ainsi que le manque de précision dans la phase « extraction » des points qui constitue leur inconvénient majeur. [6]

7.3. Les méthodes hybrides :

Comme on a vu précédemment plusieurs approches ont été proposées pour la reconnaissance de visages, sauf qu'aucune d'elle n'est capable de s'adapter aux changements d'environnements tels que la pose, expression du visage, éclairage, etc. Les techniques hybrides combinent les deux méthodes précédentes pour une meilleure caractérisation des images de visages. [4]

8. Bases de données :

Plusieurs bases de données contenant des informations qui permettent l'évaluation des systèmes de reconnaissance de visages sont disponibles sur le marché. Toutefois, ces bases de données sont généralement adaptées aux besoins de quelques algorithmes spécifiques de reconnaissance, chacune d'elle a été construite avec des conditions d'acquisition d'images de visages diverse (changements d'illumination, de pose, d'expressions faciales) ainsi que le nombre de sessions pour chaque individu. Les bases les plus anciennes (ORL et YALE) ont été le plus utilisées et permettent de comparer plus facilement de nouvelles méthodes à celles de l'état de l'art. Les plus récentes (Color FERET, FRGC, CVL, AR et IV2) contiennent plus de personnes et sont donc utiles pour des évaluations à plus grande

échelle. D'autres bases de visages sont disponibles et destinées à des évaluations adaptées à certaines variabilités du visage telles que les bases UMIST, BANCA, PF01, Yale et PIE. Ces trois dernières bases par exemple (PF01, Yale et PIE) disposent d'un nombre important de poses différentes mais renferment seulement quelques dizaines de personnes acquises lors d'une seule session. [4]

9. Reconnaissance par Eigenface :

La reconnaissance de visages par eigenfaces est une approche de type « image ». Chaque image de visage est considérée comme un vecteur dans un espace ayant autant de dimensions que de pixels dans l'image. Les caractéristiques de l'image sont extraites par une méthode mathématique de réduction de dimensionnalité basée sur l'analyse en composantes principales (ACP).

9.1. Choix de la méthode :

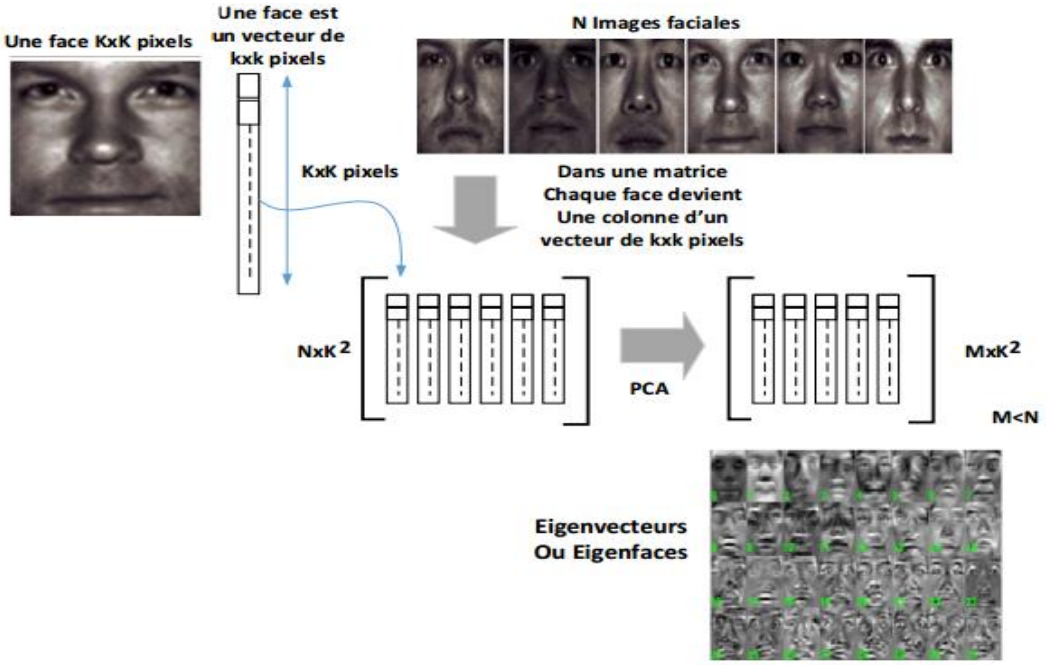
La méthode de reconnaissance de visage que nous allons utiliser pour implémenter notre projet, relève de la catégorie des méthodes globales.

L'ACP (Principal Component Analysis) est une approche proposée par Turk et Portland en 1991. Ce PCA applique des méthodes mathématiques afin de transformer un certaines nombre de variables corrélées dans un plus petit nombre de variables décolérées appelées principal components.

L'utilisation du PCA permet de convertir les images de la liste l'apprentissage en une somme d'"Eigenfaces" qui représente les principales différences entre les personnes ou les images de la liste d'apprentissage. [7]

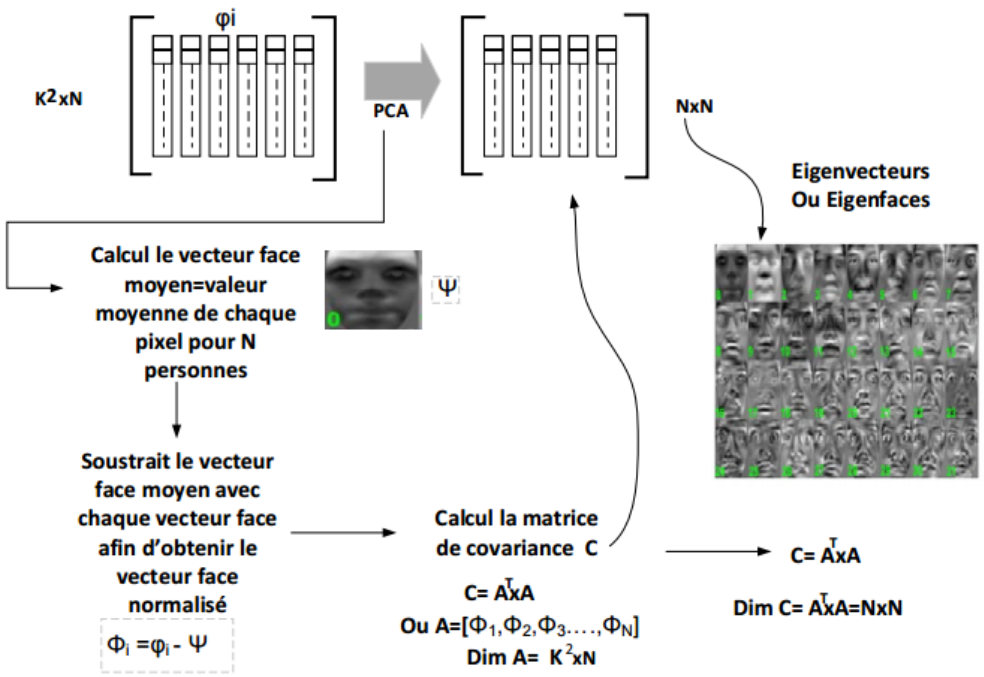
9.2. Le principe général :

Le PCA va permettre de transformer un espace vectoriel de N images dans un sous espace de dimension inférieure :



9.3. Le Fonctionnement du PCA :

Dans un premier temps le PCA trouvera la valeur moyenne de chaque pixel pour les N personnes. Les eigenfaces sont calculés en comparaison à cette face moyenne (normalisation des images), ou le premier eigenfaces est la différence de face la plus dominante, le second eigenfaces est la seconde différence de faces la plus dominante et ainsi de suite jusqu'à un nombre défini d'eigenfaces qui représente le plus de différence dans une liste d'images d'apprentissage. Le diagramme ci-dessous nous expose de façon plus complète les méthodes de calcul du PCA :

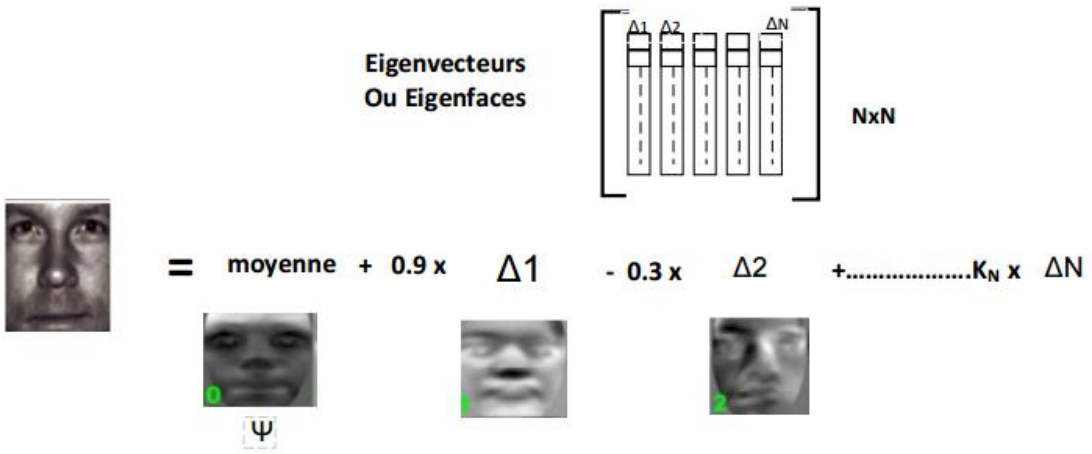


9.4.La combinaison linéaire :

On représente chaque image de la liste d'apprentissage comme la combinaison de ces différences: Par exemple une des images de la série peut être constituée des ratios suivants:

averageFace + (13.5% de eigenface0) - (34.3% de eigenface1) + (4.7% de eigenface2) + ... + (0.0% de eigenfacen).

Une fois ces combinaisons trouvées, le résultat est le suivant pour une image: {13.5, -34.3, 4.7, ..., 0.0}. Le diagramme suivant nous schématise la combinaison linéaire :



9.5. Objectifs de l'ACP :

- fournir des outils simples et lisibles de représentation des informations traitées, permettant de faire ressortir des données brutes les éventuels liens existant entre les variables (en terme de corrélation),
- donner des indications sur la nature, la force et la pertinence de ces liens, afin de faciliter leur interprétation et découvrir quelles sont les tendances dominantes de l'ensemble de données,
- réduire efficacement le nombre de dimensions étudiées (et ainsi simplifier l'analyse), en cherchant à exprimer le plus fidèlement possible l'ensemble original de données grâce aux relations détectées entre les variables. [8]

9.6. Avantages de l' ACP :

- ***Simplicité mathématique***: L'ACP est une méthode factorielle car la réduction du nombre des caractères ne se fait pas par une simple sélection de certains d'entre eux, mais par la construction

de nouveaux caractères synthétiques obtenus en combinant les caractères initiaux au moyen des "facteurs".

- ***Simplicité des résultats*** : Grâce aux graphiques qu'elle fournit, l'Analyse en Composantes Principales permet d'appréhender une grande partie de ses résultats d'un simple coup d'oeil.
- ***Puissance*** : L'ACP a beau être simple, elle n'en est pas moins puissante. Elle offre, en quelques opérations seulement, un résumé et une vue complète des relations existant entre les variables quantitatives d'une population d'étude, résultats qui n'auraient pas pu être obtenus autrement, ou bien uniquement au prix de manipulations fastidieuses.
- ***Flexibilité*** : L'ACP est une méthode très souple, puisqu'elle s'applique sur un ensemble de données de contenu et de taille quelconques, pour peu qu'il s'agisse de données quantitatives organisées sous forme individus/variables. Cette souplesse d'utilisation se traduit surtout par la diversité des applications de l'ACP, qui touche tous les domaines, comme exposé dans la partie précédente. [8]

10. Partie pratique :

10.1. Langage utilisée :

R est un logiciel dans lequel de nombreuses techniques statistiques modernes et classiques ont été implémentées qui possède une large collection d'outils statistiques et graphiques.

Les méthodes les plus courantes permettant de réaliser une analyse statistique telles que : statistique descriptive ; tests d'hypothèses ; analyse de la variance etc…

Donc, le langage R convient à l'algorithme choisit (PCA), du fait que le PCA est une méthode mathématique basée sur des notions d'algèbre linéaire et des statistiques.

10.2. La base de données utilisée :

c'est celle de YALE

10.3. Réalisation et résultats :

Dans cette partie, nous allons présenter les différentes étapes de réalisation de ce project qui est la reconnaissance de visage. Nous avant utilisé la méthode des faces propres « Eigenfaces », qui se base sur une analyse de composantes principales ACP.

La première étape consiste à installer le packages ("pixmap") des classes fournit des méthodes pour créer, compléter et convertir Images bitmap dans trois formats différents: pixmaps RGB, gris et indexés.

Après nous allant charger les donnes d'apprentissages et afficher les vues P00A+000E+00, P00A+005E+10, P00A+005E-10 et P00A+010E+00 pour toutes les matières. Ensuite nous convertissons chaque photo sur un vecteur pour construire une matrice qui nous permettons à appliquer l'algorithme PCA sur ces images.

```
mydir = "D:/Master 1 2 eme serie/FDD/Facial-Recognition-master"

setwd(mydir)
install.packages("pixmap")
library(pixmap)
```

```
pic_list = c(1:38)  # liste de taille 38
view_list = c( 'P00A+000E+00', 'P00A+005E+10', 'P00A+005E-10', 'P00A+010E+00')
dir_list_1 = dir(path="CroppedYale/",all.files=FALSE) # tous les fichie dans le c
# pour attribuer une liste vide
pic_data = vector("list",length(pic_list)*length(view_list))
# pour affecte une liste vide pour stocker la pgm pour le débogage
pic_data_pgm = vector("list",length(pic_list)*length(view_list))
# lire affiche une image
this_face = read.pnm(file = "CroppedYale/yaleB01/yaleB01_P00A+010E+00.pgm")
plot(this_face)
```

L' affichage :

```
# La fonction getChannels génériques renvoient des matrices
#numériques ou des tableaux contenant les canaux spécifiés.
this_face_matrix = getChannels(this_face)
this_face_matrix
original_size = dim(this_face_matrix)
pic_vector_length = prod(original_size)
pic_vector_length # 192 * 168
pic_mat = mat.or.vec(length(pic_list)*length(view_list),pic_vector_length)
# pour lire tous les photos et constuire la matrice de vissage original
for ( i in 1:length(pic_list) ){
  for ( j in 1:length(view_list) ){
    # Compile le nom de fichier correct
    this_filename = sprintf("CroppedYale/%s/%s_%s.pgm", dir_list_1[pic_list[i]] ,
    this_face = read.pnm(file = this_filename)
    this_face_matrix = getChannels(this_face)
    # Store pgm comme élément de la liste
    pic_data_pgm[[(i-1)*length(view_list)+j]] = this_face
    # Matrice de magasin en tant qu'élément de la liste
    pic_data[[(i-1)*length(view_list)+j]] = this_face_matrix
    # Faire le visage dans un vecteur et inclure dans la matrice de donnée
    # pic_mat c'est une matrice de taille 152 *32256
    pic_mat[(i-1)*length(view_list)+j,] =  as.vector(this_face_matrix)
  }
}
pic_mat_size = dim(pic_mat)
print(sprintf('The matrix of all faces has size %d by %d' , pic_mat_size[1] , pic
```

On passe maintenant à l'étape suivante qui est le calcule et l'affichage de visage moyenne et le calcule des valeur centrer pour tous les image:

```r
mean_face = colMeans(pic_mat)
# Maintenant, imprimez-le comme image
mean_face_matrix = mean_face
dim(mean_face_matrix) = original_size
mean_face_pix = pixmapGrey(mean_face_matrix)
plot(mean_face_pix)

# Supprimez le visage moyen
pic_mat_centered = mat.or.vec(pic_mat_size[1],pic_mat_size[2])
# calcler les valeur centrer  pour tous les image
for (i in 1:pic_mat_size[1]){
  pic_mat_centered[i,] = pic_mat[i,] - mean_face
}
```

En suit nous avons appliquée l'algorithme ACP nous utilisons la fonctions prcomp () pour effectue une analyse de composants principaux sur la matrice de données donnée et renvoie les résultats en tant qu'objet de classe prcomp.

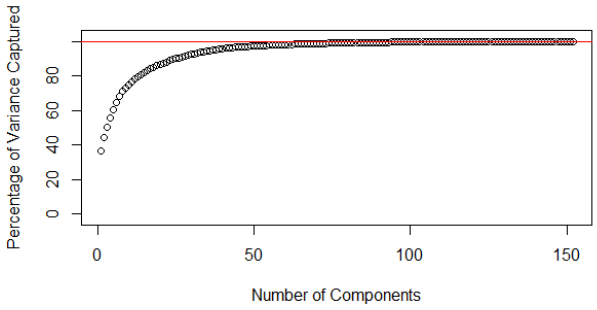
La classe "*prcomp*" contenant les composants suivants :

- ***sdev*** : les écarts-types des composants principaux c'est-à-dire les racines carrées des valeurs propres de la matrice de covariance / corrélation.
- ***rotation*** : la matrice des charges variables (i.e., une matrice dont les colonnes contiennent les vecteurs propres).
- ***x*** : si retx est vrai, la valeur des données tournées (les données centrées (et mis à l'échelle à la demande) multiplié par la matrice de rotation) est renvoyée. Par conséquent, cov(x) est le diag de matrice diagonale (de sdev ^ 2).

```r
pic_pca = prcomp(pic_mat_centered)
#Faire un vecteur pour stocker les variances capturées par les composants.
n_comp = length(pic_pca$x[,1])
pca_var = mat.or.vec(n_comp,1)
for (i in 1:n_comp){
  if (i==1){
    pca_var[i] = pic_pca$sdev[i]^2
  }else{
    pca_var[i] = pca_var[i-1] + pic_pca$sdev[i]^2
  }
}

pca_var = pca_var/pca_var[n_comp]*100
# Maintenant le tracer contre le nombre de composants
plot(pca_var,ylim=c(-2,102),xlab="Number of Components",ylab="Percentage of Varia
# Ajoute une ligne à 100 pour afficher le niveau max.
abline(h=100,col="red")
```

L'affichage :



Construire la matrice de composant principal est une image ‘Eigenfaces’ et afficher les 9 premières formes propres dans une grille 3 par 3 :

```
eigenface_mat = vector()
ind = 9
# Boucle les premières 9 formes propres.
this_face_row = vector()
for (i in 1:ind){
  # Créez le vecteur de la variable propre dans une matrice
  this_eigenface = pic_pca$rotation[,i]
  dim(this_eigenface) = original_size
  this_face_row = cbind(this_face_row,this_eigenface)
  if ((i %% 3)==0){
    # Faire une nouvelle ligne
    eigenface_mat = rbind(eigenface_mat,this_face_row)
    # clear row vector
    this_face_row = vector()
  }
}
# Plot the eigenfaces
eigenface_pgm = pixmapGrey((eigenface_mat-min(eigenface_mat))/(max(eigenface_mat)
plot(eigenface_pgm)
```

Maintenant nous avons utilisée les vissage pour reconstruire une image , on commence avec le visage moyen, on ajoute une seule image à la fois jusqu'à ce que vous atteigniez 24 formes propres. A la fin nous avons enregistrez les résultats dans une grille de 5 par 5. Encore une fois, en commençant par le visage

moyen, ajoutez cinq formes propres à la fois jusqu'à ce que vous atteigniez :

```r
# Effectue une fonction pour produire une matrice de faces 5 x 5
eigenface_add_by_face <- function(face_index, max_faces, by_faces, mean_face, pic_pca, original_size){
  # Initialiser la matrice pour les visages ajoutés dans une seule image à la fois
  face_by_eig_mat = vector() # vecteur nulle
  face_by_eig_row = vector()  # vecteur nulle
  face_by_eig_vector = mean_face
  # Stocker le visage temporaire en tant que matrice
  face_temp = face_by_eig_vector    # face_temp = mean face
  dim(face_temp) = original_size    # pour garder la mm taille de l image
  face_by_eig_row = cbind(face_by_eig_row,face_temp)

  # Ajoutez maintenant les formes propres   by_faces=1
  for (i in 1:24){ # nb de visage
    # Trouver les indices des propres propres à inclure
    ind_include = seq((i-1)*by_faces+1,i*by_faces,1)
    # Ajoutez le vecteur qui marque [j] x autoportance [j]
    eigenface_add = mat.or.vec(length(mean_face),1)
    for (j in 1:length(ind_include)){
      ind_temp = ind_include[j]
      eigenface_add = eigenface_add + pic_pca$x[face_index,ind_temp]*pic_pca$rotation[,ind_temp]
    }
    face_by_eig_vector = face_by_eig_vector + eigenface_add
    # Transformez-le en matrice et incluez
    face_temp = face_by_eig_vector
    dim(face_temp) = original_size
    face_by_eig_row = cbind(face_by_eig_row,face_temp)
    if ((i %% 5) == 4){
      # Démarrer une nouvelle ligne
      face_by_eig_mat = rbind(face_by_eig_mat,face_by_eig_row)
      face_by_eig_row = vector()
    }
  }
```

Nous avons boclée les 24 premiéres visage propre et leur resultas :

```r
# Bouclez les 24 premières formes propres
max_faces = 24
by_faces = 1
face_by_1 = eigenface_add_by_face(face_index, max_faces, by_faces, mean_face, pic_pca, original_size)
face_by_1_pm = pixmapGrey(face_by_1)

# Plot results
plot(face_by_1_pm)
```

En suit nous avons boclée les 120 premiéres visages propre et leur resultas dans une grille 5 par 5.:

```r
# Bouclez les 120 premières formes propres
max_faces = 120
by_faces = 5
face_by_5 = eigenface_add_by_face(face_index, max_faces, by_faces, mean_face, pic_pca, original_size)
face_by_5_pm = pixmapGrey(face_by_5)

# Plot results
plot(face_by_5_pm)
```

Maintenant nous avant faire l'inverse, nous utilisons les formes propres pour construire l'image originale, donc il faut utiliser la fonction prcomp () pour obtenir de nouveaux composants principaux. Apres en soustrayant le visage moyen et en projetant l'image restante sur les composants principaux.

```
pic_mat_mod = pic_mat[setdiff(1:pic_mat_size[1],1:4),]
pic_mat_mod_size = dim(pic_mat_mod)
# Recenter
mean_face_mod = colMeans(pic_mat_mod)
# Soustraire le visage moyen
pic_mat_mod_centered = mat.or.vec(pic_mat_mod_size[1],pic_mat_mod_size[2])
for (i in 1:pic_mat_mod_size[1]){
  pic_mat_mod_centered[i,] = pic_mat_mod[i,] - mean_face_mod
}
# Faites la même chose pour la photo de la requête
query_pic = pic_mat[4,]
query_pic_centered = query_pic - mean_face_mod
# Do PCA
pic_pca_mod = prcomp(pic_mat_mod_centered)
# Obtenez des scores pour la photo de requête
num_comp_mod = length(pic_pca_mod$x[,1])
scores_query = mat.or.vec(num_comp_mod,1)
for (i in 1:num_comp_mod){
  loading_temp = pic_pca_mod$rotation[,i]
  scores_query[i] = loading_temp %*% query_pic_centered
}
# Utiliser des charges pour reconstruire une image
query_reconstruct = mean_face_mod
for (i in 1:num_comp_mod){
  query_reconstruct = query_reconstruct + scores_query[i]*pic_pca_mod$rotation[,i]
}
# Tracer à côté de l'original pour la comparaison
dim(query_reconstruct) = original_size
original_query = query_pic
dim(original_query) = original_size
side_by_side = cbind(original_query,query_reconstruct)
side_by_side_pm = pixmapGrey(side_by_side)
plot(side_by_side_pm)
```

Conclusion :

Nous avons appris plusieurs choses à travers de ce projet car si la première fois que nous touchons un projet de ce type « la reconnaissance faciale ».

Ce projet été très utile pour nous car il nous a permis de découvrir comment développer un system de reconnaissance des visages en utilisant la méthode des faces propres « Eigenfaces », qui se base sur une analyse de composantes principales ACP.

Bibliography

[1] http://www.memoireonline.com/01/14/8585/m_Identification-des-personnes-par-reconnaissance-de-visage-pour-la-securite-d-une-institution-banca15.html .

[2] https://www.morpho.com/fr/reconnaissance-faciale.

[3] https://tel.archives-ouvertes.fr/tel-00623243/document.

[4] http://dspace.univ-tlemcen.dz/bitstream/112/4799/1/FaceRec.pdf.

[5] http://thesis.univ-biskra.dz/944/4/Chap%201%20Système%20RV%20sept%2012.pdf.

[6] http://www.memoireonline.com/02/13/6979/m_Reconnaissance-de-visages-par-Analyse-Discriminante-LineaireLDA-8.html.

[7] http://philpetitpa.890m.com/Eigenfaces.pdf.

[8] http://www.chambreuil.com/public/education/3.2/stat/projet/.