	ING1 – Génie Mathématique T.P. Noté : Reconnaissance faciale	
	Matière : **Analyse numérique**	*Date* : **12 avril 2021**
		Durée : **jusq. 15/04/21**
		Nombre de pages : **5**

1 Introduction

Ce TP a pour but de vous faire découvrir une des premières techniques de reconnaissance faciale, celle utilisant les **"eigenfaces".**

Cette technique de reconnaissance utilise la méthode d'analyse en composantes principales (ACP) ou la méthode de décomposition en valeurs singulières (SVD).

L'idée centrale est de diminuer la dimension de l'espace de travail pour simplifier les données et leur interprétation.

Une fois cette dimension réduite, la comparaison d'un nouveau visage à ceux de la base de données devient moins coûteuse ce qui permet d'espèrer un bon taux de réussite.

Les **"eigenfaces"** sont obtenues lors d'une première phase d'apprentissage. Il s'agit tout simplement des composantes principales de la matrice regroupant les images des visages d'une partie de la base de données.

Elles vont servir de base pour exprimer chacune des images. En se limitant aux premières composantes (une fraction qui représente un pourcentage suffisant de l'inertie totale) on réduit la dimension de représentation des images.

Ces composantes principales peuvent également être obtenues par la décomposition en valeurs singulières d'une matrice étroitement liée à celle de la base de données comme on va le voir dans la partie mathématique.

2 Notions mathématiques

Quelques rappels de cours :

Décomposition en valeurs singulières

Soit n et p deux entiers naturels non nuls et $A \in \mathscr{M}_{np}(\mathbb{R})$.

Alors il existe :

1. une matrice carrée orthogonale d'ordre n : U,
2. une matrice carrée orthogonale d'ordre p : V,
3. une matrice $\Sigma \in \mathscr{M}_{np}(\mathbb{R})$ diagonale,

telles que : $$A = U \, \Sigma \, V^\top$$

Construction d'une Décomposition en valeurs singulières

Soit n et p deux entiers naturels non nuls et $A \in \mathcal{M}_{np}(\mathbb{R})$ de rang r.

On note $(\sigma_i)_{1\leqslant i\leqslant r}$ les valeurs singulières de A.

Soit $(u_i)_{1\leqslant i\leqslant n}$ une base orthonormale de vecteurs singuliers à gauche telle que u_i soit un vecteur propre de $AA^\top$ associé à la valeur propre σ_i^2 si $i \leqslant r$. On note U la matrice de la base $(u_i)_{1\leqslant i\leqslant n}$.

Soit $(v_j)_{1\leqslant j\leqslant p}$ une base orthonormale de vecteurs singuliers à droite telle que v_j soit un vecteur propre de $A^\top A$ associé à la valeur propre σ_j^2. On note V la matrice de la base $(v_j)_{1\leqslant j\leqslant p}$.

Soit $\Sigma \in \mathcal{M}_{np}(\mathbb{R})$ la matrice définie par

$$\Sigma_{ij} = \begin{cases} \sigma_i & \text{si } i = j \text{ et } i \leqslant r \\ 0 & \text{sinon} \end{cases}.$$

$$\text{Alors } A = U\ \Sigma\ V^\top.$$

Composantes principales :

Lorsque l'on étudie m individus décrits par l variables, les données sont représentées dans une matrice $X \in \mathcal{M}_{ml}(\mathbb{R})$.

Chaque ligne représente un individu, et chaque colonne représente une variable.

Les composantes principales associées sont les vecteurs propres orthonormés de la matrice des variances/covariances C définie de la manière suivante.

Chaque colonne de X est centrée et réduite, c à d que chaque coefficient x_{ij} est remplacé par :

$y_{ij} = \dfrac{x_{ij} - \mu_j}{\sigma_j}$, où (μ_j, σ_j) représentent la moyenne et l'écart-type de la colonne j.

C'est ainsi qu'est construite la matrice Y normalisée de X.

C est ensuite calculée comme : $C = Y^T\ Y$.

La diagonalisation de C dans une base orthonormée se traduit par une écriture de la forme :

$$C = P\ D\ P^{-1} = P\ D\ P^T.$$

Les colonnes de P sont les composantes principales.

La variabilité de Y est concentrée dans les premières composantes principales. Si on décompose une ligne de Y dans les premières composantes de P, on peut donc réduire la dimension de la représentation sans perdre trop d'informations.

Lien entre ACP et SVD :

La décomposition en valeurs singulières de Y donne l'écriture suivante : $Y = U\ \Sigma\ V^\top$.

Il est alors facile de faire le lien entre les matrice P et D d'un côté, U, Σ et V de l'autre.

---

3 Travail mathématique demandé

1. Rappeler ce qu'est la norme matricielle subordonnée $\|.\|_2$.
2. Justifier pourquoi la matrice $C = Y^T\ Y$ est diagonalisable dans une base orthonormée.
3. Quel est le signe de ses valeurs propres?
4. Donner l'expression des matrices P et D en fonction des matrices U, Σ et V.
5. Sachant que les colonnes de P forment une base orthonormée, comment peut-on obtenir simplement les composantes d'un individu normalisé (une ligne de Y) dans cette base?

4 Phase d'apprentissage :

La première base de données (BDD1) est formée de plus de 1680 photos en niveaux de gris de visages de personnalités.

Vous en choisirez une centaine pour l'apprentissage.

Les images (visages) de la base de données sont formées de 64 pixels sur 64 pixels.

Leur lecture (imread) donne donc des matrices 64×64 d'entiers compris entre 0 et 255 correspondant au niveaux de gris.

1. La première étape consiste à transformer chaque matrice en une ligne de taille 4096.
2. Puis de les regrouper dans une matrice X de taille 100×4096.
3. X est normalisée comme expliqué plus haut (soustraction de la moyenne et division par l'écart-type de chaque colonne). On obtient ainsi la matrice Y.

   Les moyennes et les écart-types de chaque colonne (pixel) sont conservées, pour un usage ultérieur, dans un tableau ou une matrice M de taille 2×4096.
4. La décomposition en valeurs singulières de Y permet d'obtenir les matrices P et D. Les colonnes de P sont les fameuses "**eigenfaces**".
5. Pour le choix du nombre k de composantes à retenir, vous pouvez tester des valeurs entre 50 et 70.
6. Une fois k fixé, on désigne par P_k la matrice obtenue, à partir de P, en conservant uniquement les k premières colonnes.

   Les composantes de chaque image dans cette base partielle sont calculées par le produit : $Z = Y\ P_k$. Chaque ligne comporte les k coordonnées d'une image de la BDD dans la base formée des k composantes principales retenues "eigenfaces".

A la fin de la phase d'apprentissage, on doit conserver :

- La matrice M qui va servir à normaliser d'éventuelles nouvelles images.
- La matrice P_k des k composantes principales retenues, qui servira à obtenir les coordonnées d'une éventuelle nouvelle image par rapport à ces composantes.
- La matrice Z qui contient les coordonnées de toutes les images de la BDD par rapport aux composantes principes retenues.

5 Base de données à dimension réduite :

Grâce aux matrices M et P_k, obtenues par apprentissage sur les 100 images, on peut exprimer toutes les images de la BDD dans la nouvelle base P_k. C'est à dire en dimension $k \leqslant 100$ au lieu de 4096.

Il suffit de transformer la matrice des pixels en une ligne x_m, de la centrer et la réduire à l'aide de la matrice M pour obtenir y_m, puis de calculer $z_m = Y_m\ P_k$ qui donne les coordonnées souhaitées.

On pourra, d'ailleurs conserver le nom Z donné à la matrice des coordonnées de toutes les images de la BDD, et pas seulement des 100 qui ont servi pour l'apprentissage.

6 Reconnaissance faciale :

La deuxième base de données (BDD2) contient 100 photos différentes de certaines des personnalités qui figurent dans BDD1.
On peut donc partir d'une nouvelle image, photo d'un visage inconnu de BDD1 ou BDD2, on procède alors de la manière suivante :

1. Lecture de l'image et transformation de la matrice 64×64 en une ligne 4096 $\Longrightarrow x_p$.
2. Normalisation de cette image, c'est à dire centrage et réduction $\Longrightarrow y_p$.
3. Coordonnées de cette image dans la base des composantes principales retenues :
   $z_p = y_p P_k$.
4. Calcul des distances entre la nouvelle image et celles de la BDD :
   $d_i = \|z_p - Z_{i.}\|_2$.
5. Identification du visage par la recherche de la distance minimale.

7 Travail de codage demandé :

Ecrire un programme Scilab ou Python comportant les trois fonctions suivantes :

1. ***Apprentissage*** recevant comme entrées les cent images de la base de données et donnant à la sortie les matrices M, P_k et Z.

   VOUS EXPLIQUEREZ, DANS LE RAPPORT, VOTRE CHOIX DE LA VALEUR DE k, ET LE GAIN DE MÉMOIRE OBTENU EN %.
2. ***Ajout*** recevant comme entrée une nouvelle image ainsi que les matrices M, P_k, Z et donnant en sortie ses coordonnées dans la base P_k, qui seront rajoutées comme ligne supplémentaire de la matrice Z.
3. ***Reconnaissance faciale*** recevant comme entrées une nouvelle image ainsi que les matrices M, P_k, Z et donnant en sortie l'identité du visage reconnu (n° de la ligne de Z) et affichant les 2 images.

   VOUS DISCUTEREZ, DANS VOTRE RAPPORT, DE L'EFFICACITÉ DE CETTE MÉTHODE ET DE LA NÉCESSITÉ D'INSTAURER UN SEUIL DE DISTANCE AU DESSUS DUQUEL ON PEUT DÉCIDER QU'IL N'Y A PAS DE RECONNAISSANCE.

8 Consignes :

— Votre travail doit être rendu avant **jeudi 15 avril 23h59** dernier délai.

— Le rendu sera impérativement une archive .zip nommée **TPnote-Nom1-Nom2-Nom3.zip** comportant les codes sources Scilab ou Python et le pdf de votre rapport.

— Le rapport doit comporter impérativement l'analyse des méthodes utilisées. Aucun code informatique ne doit y être inséré, mais vous pouvez y insérer des captures d'écrans présentant des résultats ou sorties de programmes.

— Votre rapport doit être rendu sous format pdf.

— Sur la première page de votre rapport vous indiquerez les noms et prénoms des membres du groupe et le titre du TP.

— Les groupes seront formés de 2 ou 3 étudiants maximum, constitués à leur convenance. Toute absence de rendu entraînera une note de 0/20.

9 Outils nécessaires pour le TP :

Vous disposerez d'une première base de données (**BDD1**) formée de 1680 photos de visage de personnalités en niveaux de gris. C'est cette base qui servira pour l'apprentissage à l'aide de 100 de ces photos.

Vous disposerez également d'une deuxième base de données (**BDD2**) formée de 100 photos de visages de personnalités. Certaines de ces personnalités apparaissent dans BDD1 mais avec des photos différentes. BDD2 pourra donc servir pour la reconnaissance faciale.

Si vous travaillez avec Scilab, vous devez installer le package SIVP, via le menu Gestionnaire de modules - ATOMS .

Vous utiliserez, en particulier, les commandes **imread** et **imshow** dont il vous appartient de trouver l'usage dans la documentation.

N.B. :

Les pixels de l'image pouvant être codés via des entiers sur 1 octet, il est prudent de les convertir en double précision avant de faire des calculs, via la commande **double**.

Il est également possible que vous ayez à augmenter la pile de stockage des variables intermédiaires à l'aide de la comande **stacksize**.

Si vous travaillez avec Python, les packages matplotlib et Numpy contiennent déjà tout ce dont vous aurez besoin, notamment les commandes **imread** et **imshow**.