

MASTER 2 OSAE | SPÉ M : MÉTHODES NUMÉRIQUES

RAPPORT DE PROJET

"Reconnaissance faciale avec la PCA"

Encadrant :

BELMONT Renaud

Auteur :

ERBETTI Nicolas

Du 17 janvier au 11 février 2022

1 Introduction

L'objectif de cette partie est de présenter le projet ainsi que ses différents livrables.

1.1 Présentation du projet

Dans cette section, on rappelle les grandes lignes du projet ainsi que ses objectifs. Il consiste à développer deux algorithmes de reconnaissance faciale. Chacun travaille dans un espace différent pour comparer des visages tests à ceux issus d'un catalogue connu et tenter de 1) les reconnaître et 2) identifier si leur propriétaire est connu tout court ou non. Le premier utilise l'espace des pixels, tandis que le second utilise l'espace des composantes principales. La base de données avec laquelle on travaille est constitué de 40 personnes possédant chacune 10 photos de leurs visages représentés sous différents angles ou avec différentes expressions faciales. On va y extraire les images du catalogue de référence, à savoir les 9 premières photos des 39 premières personnes. La dernière photo de chaque individu permet de tester les capacités de l'algorithme à reconnaître un visage connu tandis que celles de la dernière personne permet de vérifier sa capacité à rejeter les inconnus. Les photos en elles-même sont en niveau de gris (256 valeurs possibles) et mesurent 92 pixels de large pour 112 de haut.

1.2 Livrables

Dans cette section, on présente les différents livrables ainsi que la manière dont ils sont organisés. En effet, en plus du présent rapport, du code python et sa documentation ont été produits. L'ensemble est fourni dans l'archive "Projet Méthodes Numérique". La documentation a été compilée grâce au logiciel Doxygen et présente les documentations technique et utilisateur du projet sous forme d'une page html dont l'index est présent dans le dossier "documentation". Le code source a quant à lui été développé sur Visual Studio sous Windows 10 et se trouve réparti dans les 4 fichiers "utils.py", "pixel_search.py", "make_pca.py" et "search_pca.py" du dossier "src". Enfin, l'environnement virtuel python que j'ai utilisé pendant le développement se trouve dans le dossier "projet-venv".

2 Lecture de la base et fonctions utiles

L'objectif de cette partie est de présenter les fonctions générales qui sont utilisées dans ce projet. En effet, peu importe la technique à l'oeuvre, il faudra lire les différentes images, constituer des catalogues et surtout chercher l'image la plus proche de l'image d'entrée. Les 3 "catalogues" considérés sont : celui qui contient les 9 visages connus des 39 personnes connues, celui qui contient leurs visages inconnus et celui qui contient les visages de la dernière personne. Les différentes fonctions que j'ai développées permettent cela et utilisent des variables globales qui dimensionnent notre base de données et qui modularisent mon programme. En effet, si l'on voulait rajouter une 11ème personne, ou rajouter des photos à chaque personne, il suffit d'ajouter les photos correspondantes dans le bon sous-répertoire du dossier "data" et de modifier les variables en conséquence. L'algorithme considérera toujours que la personne "inconnue" est la dernière de la liste, et que les visages "inconnus" des personnes connues sont leurs dernières photographies.

3 Comparaison dans l'espace des pixels

L'objectif de cette partie est de présenter l'algorithme derrière la méthode de comparaison des images dans l'espace des pixels ainsi que les résultats qu'il nous a permis d'obtenir.

3.1 Technique utilisée

Dans cette section, la méthode utilisée est assez brute. On utilise directement les méthodes définies dans `utils.py` pour comparer les valeurs des pixels de l'image brute aux images du catalogue pour trouver l'image correspondante dont la distance est la plus courte. Quand on parle de distance, on parle de la norme 2 du vecteur contenant les 10304 valeurs des pixels.

3.2 Résultats

Nous obtenons de bons résultats. Presque tous les visages connus sont effectivement reconnus tandis que ceux de la dernière personne ne sont pas assignés au bon id (ce qui est logique puisque le programme ne connaît pas du tout cet id). Cependant, certains visages ayant un angle particulièrement prononcé ou grimaçant ne sont pas reconnus correctement. C'est le cas de l'image test de la personne 32 disponible en figure 1. De plus, en moyenne, le temps de recherche d'un visage est d'un peu plus de 0,25 secondes.

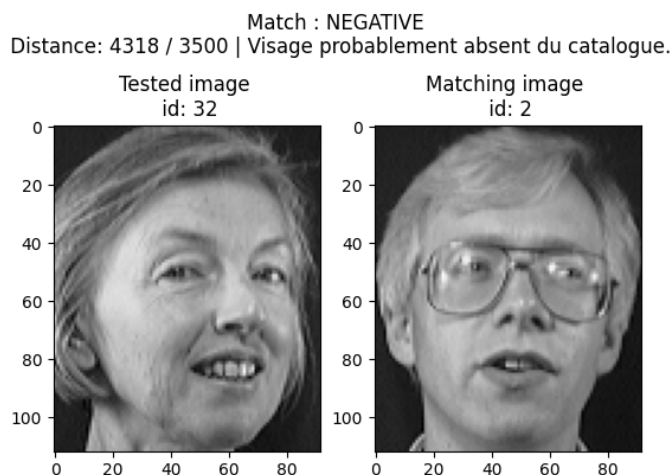


FIGURE 1 – Erreur de matching avec la méthode des pixels.

4 Comparaison dans l'espace des composantes principales

L'objectif de cette partie est de présenter l'algorithme derrière la méthode de comparaison des images dans l'espace des composantes principales ainsi que les résultats qu'il nous a permis d'obtenir.

4.1 Création d'un catalogue

Dans cette section, on définit les propriétés de la PCA et on crée un nouveau catalogue qui contient les images du catalogue de référence projetée dans cet espace. Normalement, cette méthode permet surtout de gagner en temps de calcul puisqu'on réduit la dimensionnalité du problème en traitant les images par le prisme de leurs composantes principales plutôt que par celui de leurs pixels directement. Pour l'appliquer, il m'a suffit de :

- Calculer la valeur moyenne de chaque pixels (le visage correspondant est disponible en figure 2).
- Retirer cette valeur aux valeurs des pixels des images du tableau.
- Calculer la décomposition SVD (pour singular value decomposition - décomposition en valeurs singulières) du catalogue.
- Tracer la proportion d'informations conservées en fonction du nombre de valeurs singulières considérées. On calcule pour cela le rapport entre la somme quadratique des nc premières valeurs singulières et la somme quadratique de toutes les valeurs singulières. Ce graphique est disponible en figure 3 et permet de déterminer qu'on conserve 80% de l'information de l'image originale en prenant les 43 premières valeurs ($nc = 43$ donc).
- Extraire les vecteurs propres correspondants à ces nc valeurs singulières, soient les composantes principales de la matrice $VexpT$ issues de la SVD. L'image résultante des 6 premiers vecteurs propre est disponible en figure ??.

Toutes ces informations sont pré-calculées grâce au script `make_pca.py` et sont stockées dans des fichiers textes qui seront lus plus tard par notre script de comparaison, `search_pca.py`.

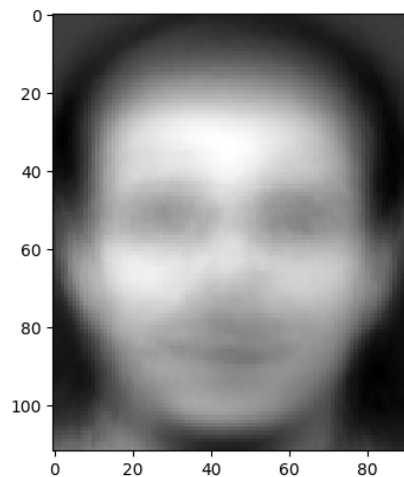


FIGURE 2 – Visage moyen obtenu en faisant la moyenne des images du catalogue.

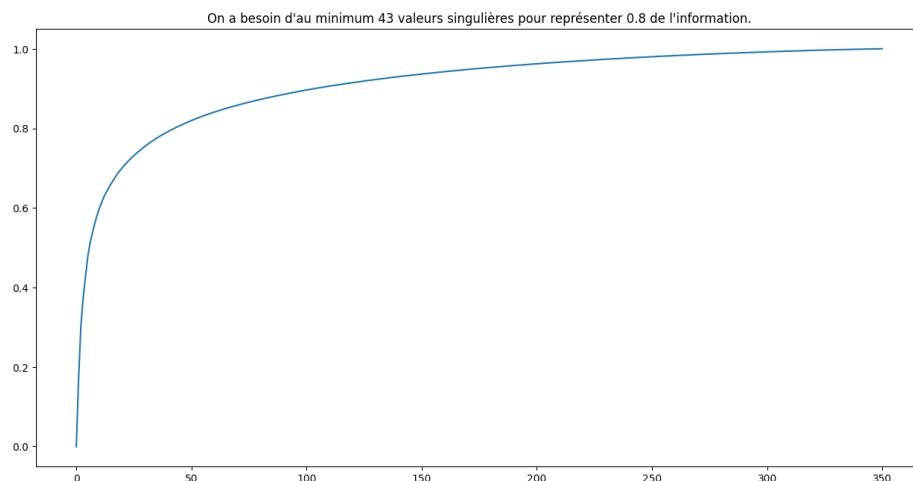


FIGURE 3 – Quantité d'information conservée en fonction du nombre de composantes principales.

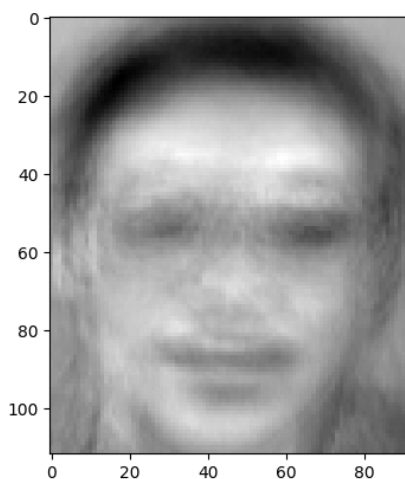


FIGURE 4 – Visage reconstitué à partir des 6 premières composantes principales.

4.2 Résultats

Nous obtenus plus ou moins les mêmes résultats qu'avec l'analyse par pixels. Cependant, les distances mesurées entre deux images matchées sont bien moins importantes qu'avec la méthode précédente. Ce résultat était attendu étant donné puisque la méthode des PCA consiste à projeter nos images dans une autre base. De plus, l'erreur de matching que nous avons pour l'image 32 a été corrigée (cf figure 5) et le temps de recherche est également réduit, passant de 0,25s à environ 0,23s.

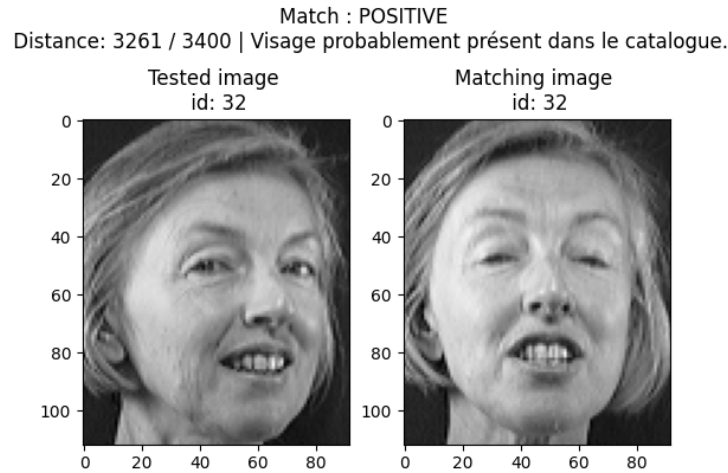


FIGURE 5 – Matching corrigé pour la personne 32

5 Identification

L'objectif de cette partie est de présenter la logique derrière le processus d'identification des différents visages comme étant connus ou non ainsi que les résultats de son application. La technique utilisée est la suivante : on commence par définir deux valeurs seuils (une pour la méthode "pixels" et une pour la méthode "pca", obtenus itérativement). Lors des tests de comparaison dans ces deux espaces, on compare la distance mesurée entre l'image de référence et l'image trouvée et ce seuil. Si la distance est inférieure au seuil, on peut raisonnablement penser que la personne représentée appartient bien au catalogue de base (et inversement).

Pour l'espace des pixels, le seuil optimal que j'ai trouvé après plusieurs itérations est d'environ 3500, mais on a toujours un faux négatif qu'on ne pourra jamais éviter avec cette méthode. En effet, on a un recouvrement des distances entre certains visages des bases connues et inconnues. Si on augmentait encore le seuil, on aurait des faux positifs dans le catalogue des visages inconnus, tandis qu'à ce niveau on a des faux négatifs. Ces deux visages sont ceux des figures 6 et 7.

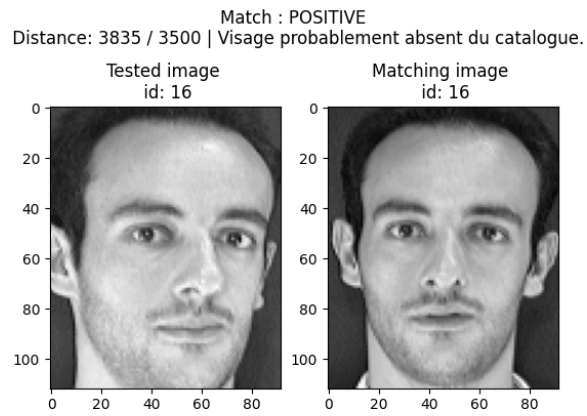


FIGURE 6 – Erreur d'identification dans la base des images connues (méthode pixel).

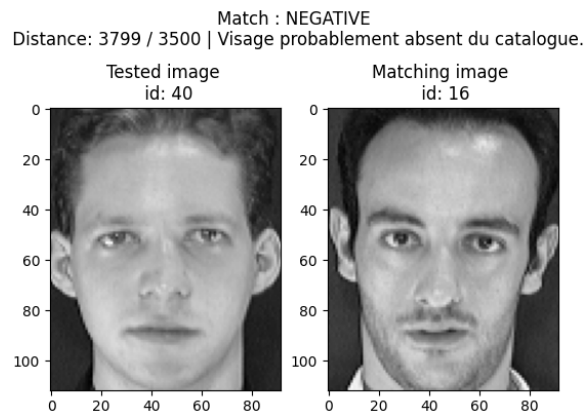


FIGURE 7 – Photo de la base inconnue responsable du recouvrement (méthode pixels).

Pour l'espace des composantes principales, le seuil optimal que j'ai trouvé après plusieurs itérations est d'environ 3400, mais nous avons toujours un peu de recouvrement avec cette méthode, issue des images pour lesquelles le matching est le plus compliqué. Même en augmentant le pourcentage d'informations conservées à 0.9, voir à 0.99, on n'arrive pas à éviter ce recouvrement (cf les figures 8 et 9). En le baissant, on obtient de moins bons matching, et les distances mesurées diminuent également (ce qui empire la plage de recouvrement).

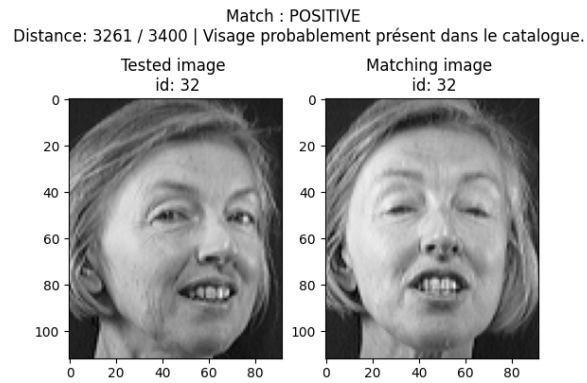


FIGURE 8 – Photo de la base connue responsable du recouvrement (méthode pca)

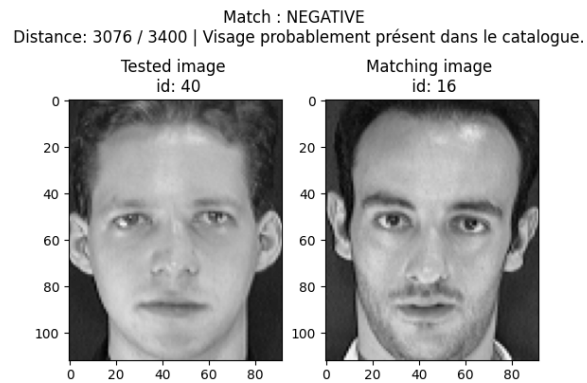


FIGURE 9 – Erreur d'identification dans la base des images inconnues (méthode pca).

6 Conclusion

Après avoir comparé les performances des deux méthodes, aucune ne se distingue clairement en termes de capacité d'identification. En effet, même si celle utilisant les PCA semble dotée d'une meilleure capacité de reconnaître les visages connus (si le pourcentage d'informations conservées est assez élevé), le gain est difficile à évaluer (un seul visage "corrigé"). Surtout, les capacités de détecter si le visage est connu tout court semblent équivalentes et les recouvrements sont impossibles à éviter dans une méthode comme dans l'autre peu importe le seuil choisi.

Cependant, les durées de calcul sont moins importants avec la méthode des PCA, pour peu qu'on ne compte pas le temps nécessaire à la construction du catalogue dédié. C'est le principal avantage comparé à la méthode des pixels, et c'est un résultat auquel on pouvait s'attendre puisque le principe des PCA consiste à réduire les dimensions (et donc le volume) des données à traiter. Elle peut également constituer un des éléments fondamentaux de méthodes plus performantes comme celles utilisant le machine learning.