Génération et recalage de modèle 3D de visages à partir d'images

par

KEVIN BECQUET

RAPPORT DE PROJET PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE EN TECHNOLOGIE DE LA SANTÉ

MONTRÉAL, LE 30 AOÛT 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Prof. Carlos Vazquez, directeur de projet
Département de génie logiciel et TI à l'École de technologie supérieure

Prof. Simon Drouin, président du jury
Département des Technologies de la Santé à l'École de technologie supérieure

REMERCIEMENTS

Je remercie dans un premier temps mon directeur de projet Pr. Carlos Vazquez pour les conseils qu'il m'a apporté qui m'ont permis de de mener à bien ce projet.

Je souhaite également remercier Thierry Cresson et Antoine Dessolies, avec qui j'ai pu travailler sur ce projet. Ils m'ont été d'une grande aide et m'ont permis d'apprendre beaucoup au cours de ce projet.

J'aimerais ensuite remercier l'École de Technologie Supérieure (ÉTS) pour la formation et l'opportunité de réaliser ce type de projet qui m'a permis de gagner en expérience.

Je tiens enfin à remercier le Centre de Recherche du Centre Hospitalier Universitaire de Montréal (CRCHUM) de m'avoir accueilli et pour son environnement de travail ainsi que de proposer ces projets qui font avancer la recherche en santé.

GÉNÉRATION ET RECALAGE DE MODÈLE 3D DE VISAGES À PARTIR D'IMAGES

Kevin BECQUET

RÉSUMÉ

Ce projet, réalisé avec le Centre de Recherche du Centre Hospitalier de l'Université de Montréal, a pour vocation de venir en soutien aux patients ayant perdu une partie de leur visage (nez, bouche ou œil par exemple). Le but de ce projet est de réaliser un système capable de générer des modèles 3D de visages à partir d'image puis de les superposer sur ces images. Il s'implante dans un plus grand projet d'implémentation d'un système de réalité augmentée permettant aux patients de voir leur visage avec un modèle numérique de la prothèse faciale qui leur a été conçue pour combler la partie du visage qu'ils ont perdu. Ce système aurait accès au flux vidéo d'une caméra devant laquelle sera positionné le patient et afficherait sur un écran le visage du patient modifié pour y ajouter sa prothèse.

Il existe effectivement des méthodes de reconstruction de modèle de visage à partir d'image dans la littérature mais aucun ne parle de patients similaires à notre cas. Il est possible de réaliser cette reconstruction en utilisant des réseaux de neurones. Ces derniers pourront retrouver dans les images les caractéristiques du visage en replaçant un modèle basique puis en le modifiant pour s'adapter à ce dernier. Une autre méthode est de décrire le visage suivant une série de paramètres et laisser le réseau de neurone retrouver ces paramètres à partir de l'image. Ce sera la méthode implémentée et décrite dans ce rapport.

On présentera par la suite les résultats obtenus lors de nos essais comprenant deux réseaux différents et deux manières de représenter le visage. Ces résultats seront ensuite discutés, présentant les limites actuelles des tests effectués et les travaux futurs.

GENETATION AND IMAGE REGISTRATION OF A FACIAL 3D MODEL FROM AN IMAGE

Kevin BECQUET

ABSTRACT

This project, realized with the Centre de Recherche du Centre Hospitalier de l'Université de Montréal, is meant to support patients who have lost a part of their face, (nose, mouth or eyes for example). The goal of this project is the creation of a system that generates 3D model of a face from an image and then puts that model on that image. It is part of a bigger project, the implementation of an AR system allowing patients to have a preview of their face with the prosthesis of their lost facial part added to it. The system will have to do that by using the video flow from a camera, the patients will be placed in front of said camera, and display on a screen the patients' face, modified to add the prosthesis, for the patients to see.

We can find in the literature several methods to realize this project but only a few of them have been tested on a population similar to the one we are targeting. For example, some articles introduce the use of neural networks to directly reconstruct a facial 3D model by placing a basic model of the face in the image and modifying it to fit precisely the image. Some other articles present a way to describe the face with a few parameters defining its shape and then use neural networks to extract those parameters from the input image. This is the method that we used in this project and that will be described in this report.

After describing the method, we will introduce the results obtained from the tests made using it that include wo different neural networks and two ways of representing the face. Finally, these results and their limitations will be discussed and the future works on this project will be presented.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CNN : Convolutionnal Neural Network ou réseau de neurones convolutif

PCA : Principal Components Analysis ou Analyse en Composantes Principales

PnP : Perspective-n-Points

BD : Base de Données

VP : Vecteur Propre

MAE : Mean Average Error ou Erreur Moyenne Absolue

MSE : Mean Squared Error ou Erreur Moyenne Carrée

INTRODUCTION

Ce document est un rapport de projet réalisé en collaboration avec le Laboratoire de recherche en Innovation Ouverte en Technologies de la Santé (LIO-ÉTS) dans le cadre de ma maîtrise par projet en technologie de la santé à l'École de Technologie Supérieure (ÉTS).

Dans le cas d'un cancer au niveau des tissus du visage, l'oncologue peut parfois se retrouver dans l'obligation d'amputer, partiellement ou totalement, certains organes du visage tels que le nez, l'œil ou encore l'oreille. Cette intervention, modifiant grandement l'apparence du patient, peut laisser un impact psychologique lourd sur ce dernier. Pour pallier ce problème, nous pouvons fabriquer des épithèses, prothèses faciales en silicone, afin de rendre au patient une apparence la plus proche possible de son visage d'avant le traitement. Ces épithèses sont généralement imprimées en 3D ou moulées avec de la silicone à partir de modèles numériques de la prothèse. Il est donc compliqué de les modifier une fois physiquement fabriquée. Par conséquent, il serait donc pertinent de pouvoir simuler et observer le rendu du visage du patient portant l'épithèse avant que cette dernière ne soit physiquement construite. Cela permettrait à la fois au praticien de réaliser des retouches sur les modèles numériques et aux patients d'avoir un premier aperçu de leur apparence future.

C'est de cette demande qu'est né le projet d'implémenter un système en réalité augmentée permettant de visualiser des épithèses sur le visage des patients ayant subis des cancers au visage. Un système fonctionnant comme un miroir magique, le patient se placera devant un écran et une caméra capturant une vidéo et qu'il puisse voir, en temps-réel, son visage sur un écran auquel aura été ajouté un modèle numérique de sa future épithèse affiché sur l'écran. Nous serons deux à travailler sur ce projet. Antoine, étudiant à la maitrise par mémoire et moi-même, étudiant à la maitrise par projet.

Je porterai donc mon attention uniquement sur une partie du projet. Mon objectif sera de créer un système capable de générer un modèle 3D d'un visage à partir d'une image contenant ce

visage et de superposer le modèle généré sur le visage dans l'image dans la position dans laquelle il est représenté dans cette dernière.

Afin de mener à bien cette partie du projet, il sera nécessaire de disséquer notre objectif en plusieurs points clé qui seront soumis à certaines contraintes pour assurer sa réussite. En effet, la littérature sur les techniques de suivi de visage et de modélisation 3D devra être analysée afin de trouver la méthode la plus adaptée afin d'obtenir les meilleurs résultats possibles en termes de génération de visage.

Ces différents points sont soumis aux contraintes suivantes afin de pouvoir apporter une certaine satisfaction dans le rendu du système. Le suivi de visage se doit d'être robuste. En effet, les visages des patients présentent des caractéristiques peu communes par rapport à d'autres personnes. Il est important que ces dernières n'empêchent pas le bon fonctionnement du processus. Ensuite, la modélisation du visage doit être suffisamment précise pour que le patient, qui se regarde devant l'écran, puisse se reconnaitre. De même, la superposition du modèle sur les images doit être suffisamment précise pour bien suivre les mouvements de tête du patient et donc renforcer le sentiment d'incarnation ressenti.

Dans ce rapport, nous commencerons par présenter la revue de littérature puis introduirons la solution qui a été choisie et implémentée avant de porter notre attention sur les résultats obtenus et les discuter.

CHAPITRE 1

REVUE DE LITTÉRATURE

On cherche donc à réaliser un système simplifier de « miroir magique ». Un système capable d'estimer la forme et le positionnement d'un visage dans une image afin de générer un modèle 3D lui correspondant et de le replacer dans la même pose, c'est-à-dire la même position et orientation, que dans l'image. Dans la littérature, on a pu remarquer un certain nombre d'article partageant une méthode qui réalise exactement cela comme (Nguyen, Nguyen, Dakpé, Ho Ba Tho, & Dao, 2022), (J. Guo et al., 2020), (Hashemibakhtiar, Cresson, Nault, de Guise, & Vázquez, 2024) ou encore (Zielonka, Bolkart, & Thies, 2022) qui seront présentés dans la section suivante.

1.1 Génération de modèle 3D et recalage simultané

Les articles mentionnés ci-dessus utilisent une méthode capable de générer un modèle 3D de visage et de retrouver les informations de pose nécessaire à la réalisation du recalage en même temps (Nguyen et al., 2022), (J. Guo et al., 2020), (Hashemibakhtiar et al., 2024). Le principe de cette méthode repose sur l'expression des caractéristiques du visage qui est réalisée avec un nombre réduit de paramètres. Au lieu de retrouver un à un chacun des points du visage, on va en décrire les caractéristiques globales de sa forme qui serviront enfin à venir reconstruire on modèle 3D de visage grâce à un modèle d'Analyse en Composantes Principales (PCA).

1.1.1 Construction du modèle PCA

Commençons donc par présenter ce qu'est un modèle PCA. Cette méthode d'apprentissage machine dont le but va être de trouver la meilleure façon de décrire un ensemble de données relativement corrélées. L'idée va être de retrouver une transformation de données de manière à pouvoir réduire le nombre de dimension décrivant les données en perdant le moins d'information possible (Jolliffe & Cadima, 2016). Sur la Figure 1, on retrouve un ensemble de point corrélé, ils suivent tous une certaine droite. Le modèle PCA correspondant va donc

avoir cette dernière pour première dimension et l'axe perpendiculaire en seconde dimension. Comme on pourra le remarquer, projeter les points sur la première dimension du modèle PCA nous permettra de garder plus d'information sur leur dispersion qu'une projection sur l'axe des abscisses de ce graphe. Ces dimensions sont appelées vecteurs propres (VP). N'importe quel point de l'ensemble peut ainsi être décrit par une combinaison linéaire de l'ensemble des vecteurs propre du modèle statistique.

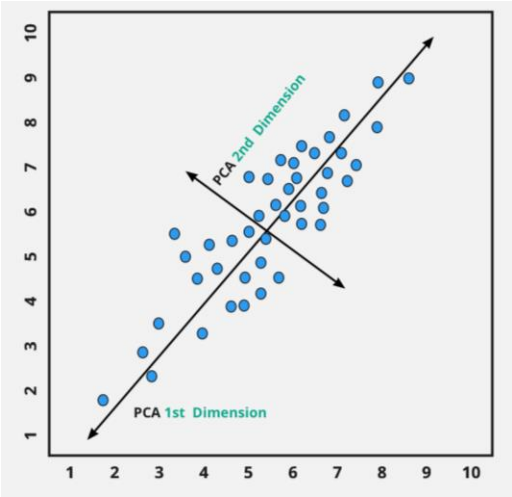

Figure 1 - Exemple de détermination de modèle PCA tiré de (Arif, 2021)

1.1.2 Réduction de paramètres

L'intérêt du modèle PCA est de pouvoir réduire le nombre de paramètres utilisé pour caractériser un même objet en perdant le moins d'information possible. Par exemple, en reprenant la Figure 1, les différents points peuvent être décrits par deux paramètres. Cependant, il est possible de les décrire avec un seul paramètre en sans perdre énormément d'information en en récupérant leur coordonnée sur l'axe de la première dimension du modèle PCA (Salem & Hussein, 2019). Dans le cas d'un modèle 3D de visage, cela permet de ne pas utiliser toutes les coordonnées de chaque point mais un nombre réduit de paramètres pour pouvoir le décrire.

1.1.3 Application de la méthode aux visages

Cette méthode va fonctionner de la même manière sur les visages. Bien que relativement différents, ces derniers suivent généralement une forme et partagent des caractéristiques similaires. (Nguyen et al., 2022) expliquent que la reconstruction d'un visage à partir d'un modèle PCA est une somme de la combinaison linéaire des vecteurs propres du modèle et du visage moyen qu'il représente. En effet la Figure 2 tirée de cet article explique qu'un visage 3D est une somme entre un visage moyen, ou meanface, et des produits de multiplication entre vecteurs propres appelés ici eigenfaces, et les paramètres statistiques α_i qui sont propres à chaque visage.

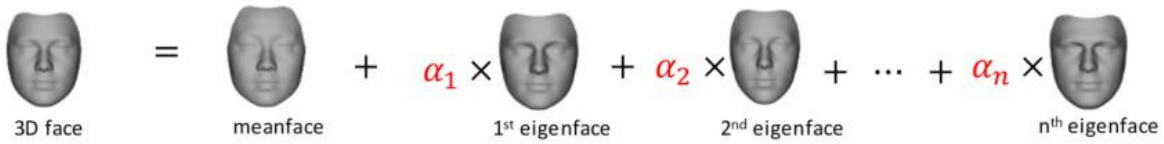

Figure 2 - Construction d'un modèle 3D de visage à partir d'un modèle PCA tiré de (Nguyen et al., 2022)

Utiliser une méthode basée sur un modèle PCA va donc demander de créer ce modèle statistique pour obtenir les « eigenfaces » présentées sur la Figure 2 d'un côté et entrainer un réseau de neurones afin de déterminer les paramètres statistiques α_i de l'autre comme le font (Nguyen et al., 2022) ou encore (J. Guo et al., 2020).

Plusieurs modèles PCA décrivant des visages existent déjà tel que le modèle de (Paysan, Knothe, Amberg, Romdhani, & Vetter, 2009) qui est réalisé avec les scans 3D de visages de 200 individus et mis à jour périodiquement ce qui le rend plutôt populaire. Cependant, ce modèle ne contient que des visages classiques, cela pourrait nuire à des utilisations similaires à la nôtre où les visages de patients ne correspondent pas à ces visages.

L'article de (Nguyen et al., 2022) a pour but de réaliser la reconstruction du visage de patients atteints de paralysie faciale. Par conséquent, utiliser un modèle PCA déjà construit uniquement sur des visages de personnes non atteintes par ce trouble n'est pas si pertinent du fait que les

données seront toutes différentes du cas d'utilisation. Les chercheurs ont donc décidé d'améliorer ce modèle en ajoutant des données de patients afin de créer leur modèle PCA plus spécifique qui pourra donc, par la suite, mieux reconstruire leurs visages. Ainsi, pour créer ce modèle PCA adapté à leur cas, leur base de données devra comprendre au minimum les modèles 3D des visages afin de créer leur modèle PCA adapté.

Une fois ces solutions implémentées il est nécessaire de pouvoir valider les résultats obtenus par le modèle PCA pour ce qui est de la génération de modèles 3D et ceux obtenus par les réseaux de neurones pour l'estimation des informations de pose et des paramètres statistiques des visages.

1.1.3.1 Estimation des paramètres statistiques et des informations de pose

Une fois le modèle PCA construit et validé, il est nécessaire d'entraîner un réseau de neurones afin d'extraire les paramètres statistiques utiles au modèle PCA pour reconstruire un visage à partir d'une image dudit visage. (Nguyen et al., 2022) utilisent pour cela d'un CNN basé sur un ResNet50 dont l'architecture est présentée à la Figure 3 qui est entraîné sur la base de données qu'ils ont créée.

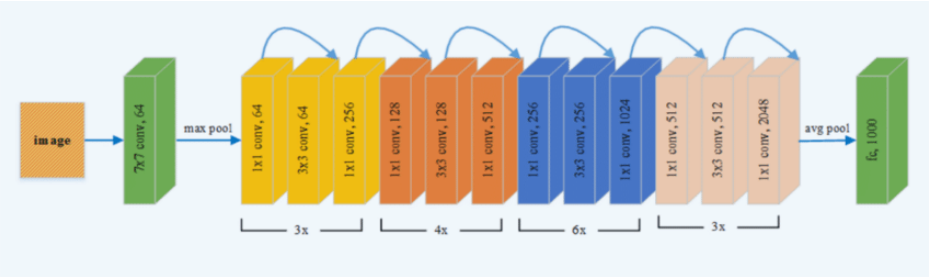

Figure 3 - Architecture du réseau ResNet50 tiré de (Albahli, 2022)

Ce réseau de neurones a pour particularité de retransmettre les cartes d'activations des couches précédentes aux couches suivantes afin de pouvoir créer un réseau plus profond pour extraire des composantes plus complexes qu'un réseau de convolution classique (He, Zhang, Ren, & Sun, 2015). Le réseau prend donc en entrée l'image du patient et retourne en sortie un vecteur

$[\boldsymbol{\alpha}\ \boldsymbol{p}]$ comprenant le set de paramètres statistiques du visage (les α_i de la Figure 2) et ses informations de pose. Ces informations de pose décrivent l'orientation et la position du visage dans l'image. (Nguyen et al., 2022) utilisent une représentation à 7 paramètres pour ces information : l'orientation est représentée par 3 angles utilisant un formalisme d'angles d'Euler, la position est représentée par 3 translations élémentaires suivant les axes x, y et z afin de placer le modèle dans un référentiel 3D et enfin un facteur d'échelle permettant d'agrandir ou rapetisser le visage selon la taille qu'il prend dans l'image.

$$\boldsymbol{p} = [\phi\ \gamma\ \theta\ t_x\ t_y\ t_z\ f] \tag{1.1}$$

Une fois toutes ces informations retrouvées par le réseau de neurones, le modèle PCA reconstruit un modèle 3D du visage en utilisant les paramètres statistiques comme expliqué sur la Figure 2 et ce dernier est replacé dans la même pose que dans l'image grâce aux inforamtions de pose. (Nguyen et al., 2022) arrivent à avoir une erreur de reconstruction moyenne de 3 mm en comparant point à point la reconstruction du visage par le modèle PCA et le scan 3D de patients test.

L'article de (Zielonka et al., 2022) vise également l'objectif de reconstruire des modèles 3D de visages à partir d'images en accentuant l'aspect de robustesse de la reconstruction. L'équipe de recherche démontre des résultats similaires à l'article précédent en termes d'erreur de reconstruction en utilisant une architecture de réseau ArcFace, basée sur le même principe de ResNet. En plus de cela, ce réseau montre une forte robustesse face aux occlusions. La Figure 4 montre un exemple de reconstruction sur des images d'un sujet portant ou non des lunettes de soleil ou le modèle en sortie semble inchangé entre ces images. Cependant, cet article ne montre que des résultats qualitatifs.

Figure 4 - Reconstruction faciale avec occlusion tirée de (Zielonka et al., 2022)

L'article de (J. Guo et al., 2020) présente une méthode similaire pour la reconstruction de visage en utilisant un modèle basé sur un CNN utilisant peu de poids afin de réduire le temps d'exécution, le MobileNet-V2. Ils utilisent ici une image de visage en entrée afin de ressortir un vecteur contenant les paramètres statistiques pour un modèle PCA et les informations de pose.

La différence avec les articles précédents va se trouver sur la représentation des paramètres de pose. En effet, l'utilisation d'angles d'Euler peut entrainer l'apparition du phénomène de Gimbal Lock (Hemingway & O'Reilly, 2018). Cette représentation de l'orientation divise la rotation en trois rotations élémentaire consécutives autour d'axes fixes par rapport à l'objet. Ainsi, dans certains cas de rotations, on peut se retrouver dans des configurations où deux axes se retrouvent alignés. La rotation de l'un ou l'autre crée donc le même effet sur l'objet. Cela peut être source de problème pour le modèle du fait que plusieurs valeurs d'orientation différentes pourraient être représentées de la même manière sur l'image et donc réduire la précision des prédictions.

(J. Guo et al., 2020) proposent de pallier ce problème par l'utilisation d'une matrice de transformation à la place d'un vecteur de pose. Cela rend explicite la transformation du modèle 3D du visage et donc contourne le problème de Gimbal Lock au prix de passer de 7 paramètres à retrouver (3 rotations, 3 translations, 1 facteur de mise à l'échelle) aux 12 paramètres d'une matrice 3x4 comme on peut le voir représenté aux équations (1.1) et (1.2). Cependant, cet article ne présente pas de résultats d'erreur sur la pose prédite par leur réseau.

$$\leftrightarrow T = \begin{bmatrix} \mathrm{T}_{11} & T_{12} & \mathrm{T}_{13} & \mathrm{T}_{14} \\ \mathrm{T}_{21} & \mathrm{T}_{22} & \mathrm{T}_{23} & \mathrm{T}_{24} \\ \mathrm{T}_{31} & \mathrm{T}_{32} & \mathrm{T}_{33} & \mathrm{T}_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix} \leftrightarrow T = [T_{11}\, T_{12} \ldots T_{43}] \tag{1.2}$$

1.1.4 Utilisation de la méthode dans d'autres domaine

Cette méthode de reconstruction et de recalage de modèles 3D n'est pas uniquement appliquée au visage. (Hashemibakhtiar et al., 2024) l'utilisent sur l'articulation de la cheville à partir d'image de radiologie. C'est-à-dire la génération d'un modèle 3D qui lui correspond et l'identification de la pose de la cheville afin de réorienter le modèle. Dans ces travaux, qui ont été également réalisé au LIO-ÉTS, le modèle est reconstruit en extrayant, à partir d'une image de radiographie, les paramètres statistiques de l'articulation qui pourra être reconstruite en utilisant un modèle PCA et les paramètres de pose qui permettront de replacer la cheville telle qu'elle est présentée dans l'image. Cela est réalisé en utilisant un réseau VGG16, un réseau CNN développé pour sa profondeur. Comme le montre la Figure 5, ce réseau va prendre en entrée une radiographie de la cheville et va ressortir un vecteur de 27 paramètres contenant les paramètres statistiques expliquant la forme de l'articulation et les informations de pose de cette dernière. L'expression de la pose est ici réalisée à l'aide d'angles d'Euler de manière similaire à l'équaation (1.1) mais dans ce cas-ci, le problème de Gimbal Lock n'est pas significatif dans car l'amplitude du mouvement en rotation de la cheville est relativement faible autour des axe qui ne sont pas son axes de flexion principal.

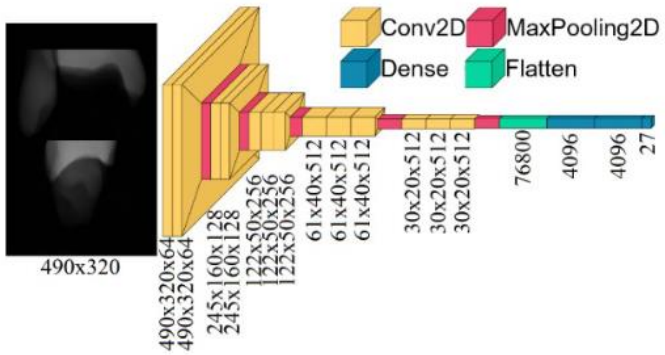

Figure 5 - Architecture du réseau VGG16 utilisée par (Hashemibakhtiar et al., 2024)

Cette méthode donne des résultats pour les angles semblant très précis. La rotation est exprimée en angles d'Euler et l'erreur de ce modèle sur ces derniers est inférieure à 1° en moyenne. L'erreur totale sur la reconstruction semble également précise, on retrouve une erreur moyenne inférieure à 1mm entre les modèles reconstruits et recalés et les modèles de référence.

La méthode de génération de modèle 3D et d'estimation de paramètres de pose en simultanée semble simple à mettre en place, elle ne nécessite qu'un entrainement de réseau de neurones et l'implémentation d'un modèle PCA. Cependant, elle demande de trouver une base de données contenant à la fois des images de visages, leurs modèles 3D correspondants et leurs informations de pose.

1.1.5 Validation du modèle PCA

Dans l'optique de valider le modèle PCA créé, (Heimann & Meinzer, 2009) proposent une série de trois tests. Ces tests sont le test de compacité, dont le but est de vérifier si le modèle est compact, c'est-à-dire à quel point il est capable de reconstruire un élément connu, le test de généralisation qui vient vérifier si le modèle peut reconstruire des données nouvelles à partir de son entrainement et le test de spécificité qui a pour objectif de valider la capacité du modèle à générer des visages aléatoires.

1.1.5.1 Test de compacité

Le test de compacité va évaluer la capacité du modèle statistique à reconstruire un élément de la base utilisée pour l'entrainement. La réalisation de ce test commence par choisir aléatoirement 10 données de la base utilisée pour entrainer le modèle. Ensuite, ce dernier est utilisé pour créer les sets de paramètres statistiques des données choisies puis retransformer ce set de paramètre en modèle de visage en faisant varier le nombre de vecteurs propres conservés dans le modèle PCA. Comme tous les vecteurs propres ne sont pas utilisés pour reconstruire les modèles 3D, une certaine quantité d'information est perdue. L'analyse de la moyenne des erreurs MSE point à point sur les 10 échantillons du test permet donc de valider si le modèle PCA est compact : cette moyenne est censée diminuer lorsque le nombre de vecteurs propres augmente en convergeant vers 0.

1.1.5.2 Test de généralisation

Le test de généralisation sert à évaluer les compétences de généralisation du modèle, c'est-à-dire à quel point il est capable de reproduire un set de données qu'il n'a jamais vu auparavant. Ce test utilise 10 sets de données qui n'ont pas servis à l'entraînement du le modèle PCA. La suite des opérations du test est similaire à celle effectuée lors du test de compacité. Le modèle transforme les sets de données en set de paramètres statistiques puis les retransforme, en faisant varier le nombre de VP conservés, en modèle 3D de visage afin d'effectuer la moyenne d'erreur MSE point à point en fonction du nombre de VP conservés. On s'attend également à voir l'erreur converger vers 0 avec l'augmentation du nombre de VP dans le modèle

1.1.5.3 Test de spécificité

Le test de spécificité a pour but de vérifier si le modèle statistique est capable de créer des données étant proches de celles utilisées lors de l'entrainement, on veut que le modèle soit suffisamment spécifique pour s'assurer qu'il ait trouvé les caractéristiques propres aux visages et qu'il puisse donc en reconstruire un.

Pour cela, on commence par générer 10 visages aléatoires de la manière suivante. On applique dans un premier temps notre modèle PCA sur l'ensemble des données sur lesquelles il a été entrainé pour obtenir leurs sets de paramètres statistiques. On retrouvera ensuite, pour chaque paramètre, la moyenne et l'écart-type sur l'ensemble des données. Ces valeurs permettent la génération d'un nouveau set de paramètres aléatoirement en suivant une loi normale de même moyenne et écart-type pour chacun de ces paramètres. Enfin, il suffit de réappliquer le modèle PCA sur ce nouveau set de paramètres afin d'obtenir le visage correspondant à ce dernier qui, au vue de la construction des paramètres, sera un nouveau visage généré.

Une fois les visages générés, On cherche le visage qui lui ressemble le plus dans la base d'entrainement du modèle statistique en réalisant la même opération de calcul de l'erreur que lors des tests précédents et en conservant la plus faible. Une moyenne de ces erreurs est réalisée sur les 10 échantillons générés et l'expérience est réitérée en faisant varier le nombre de VP conservé par le modèle. Contrairement aux tests précédents on ne s'attend pas à voir l'erreur converger vers 0 mais plutôt à ce qu'elle reste relativement stable. En effet, ici on cherche plus à voir si le modèle statistique est capable de générer de nouveaux visages. Une erreur de 0 signifierait que ce dernier reproduit uniquement des visages qu'il a déjà vu. De l'autre côté, on ne veut pas d'une erreur trop importante ou instable non plus, cela signifierait que le modèle n'est pas du tout capable de reproduire un visage.

1.1.6 Validation de la génération de modèle3D et du recalage

Pour analyser les résultats obtenus par cette méthode, les différents articles étudiés (Nguyen et al., 2022), (Hashemibakhtiar et al., 2024), (Bouaziz, Wang, & Pauly, 2013), (J. Guo et al., 2020) ont l'air de s'accorder sur une méthode similaire. Cette méthode consiste à mesurer la distance entre un point du modèle 3D de référence utilisé et le même point modèle 3D reconstruit par leur modèle PCA grâce au paramètres estimés par leurs différents réseaux de neurones. Ces articles calculent ensuite une erreur moyenne absolue (MAE) à partir de ces distances.

Ils ne parlent cependant pas de l'impact de la prédiction de la pose par rapport à celle de la reconstruction du modèle sur cette erreur.

1.2 Méthode de suivi de visage alternatives

Nous avons pu remarquer d'autres méthodes afin de retrouver le visage dans l'image et en retrouver la pose. Cela est généralement réalisé à l'aide de réseaux de neurones convolutifs (CNN) du fait de leur capacité à extraire des informations d'images (Minaee, Luo, Lin, & Bowyer, 2021). Ce type de réseau fonctionne en faisant passer une série de filtres sur une image afin d'en ressortir des formes et caractéristiques spécifiques (O'Shea & Nash, 2015). On entraîne ces réseaux de manière à ce que ces différents filtres s'adaptent pour retrouver les formes spécifiques que l'on veut retrouver dans l'image. Ici, ce sera le visage et ses points d'intérêts (contours des yeux, nez et bouche, …) (Minaee et al., 2021).

La méthode trouvée dans la littérature pour replacer le modèle 3D dans la bonne orientation est d'utiliser un réseau de neurones afin de déterminer les points du visage à partir d'une image et d'ensuite s'en servir pour replacer correctement le modèle dans l'image.

1.2.1 Retrouver des points d'intérêts sur un visage

Afin de retrouver des points sur un visage, on retrouve deux articles présentant une méthode similaire (Grishchenko, Ablavatski, Kartynnik, Raveendran, & Grundmann, 2020), (Prados-Torreblanca, Buenaposada, & Baumela, 2022).

La Figure 6 tirée de (Grishchenko et al., 2020) utilise un premier réseau, le Blaze Face Detector, pour retrouver un visage dans une image. Ensuite on y présente un réseau à graphe d'attention, Feature extractor dans la Figure 6, pour détecter différentes zones clés comme autour des yeux et la bouche et finalement y placer les différents points du visage en 3D par interpolation. Cette méthode permet d'obtenir des points en 3D d'un visage de manière rapide, l'article présente un temps d'exécution de 14ms (soit 71 FPS) sur un smartphone Pixel 2XL.

Les points ainsi estimés semble bien suivre le visage une fois projetés sur l'image 2D. On a cependant aucune indication quantitative des performances 3D de ce réseau.

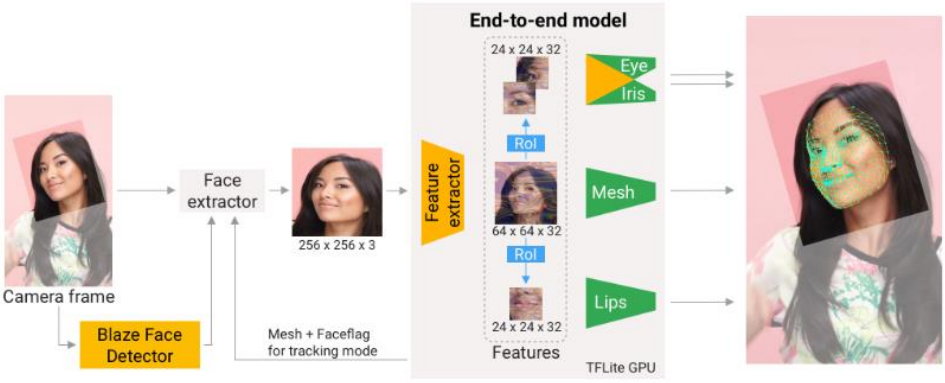

Figure 6 - architecture du réseau MediaPipe tiré de (Grishchenko et al., 2020)

Cette méthode semble intéressante pour ce qui est du suivi du visage, on voit que les points prédits par cette dernière sont bien situés sur le visage mais le manque d'informations sur ses performance en 3D nous empêche de nous prononcer pour ce qui est de son utilisation pour la génération de modèle 3D.

(Prados-Torreblanca et al., 2022) utilise également un réseau graphique, ici en combinaison à un CNN encodeur décodeur afin de retrouver un visage et ses points d'intérêts. Dans cette méthode, schématisée à la Figure 7 le réseau CNN fonctionne de la même manière que dans l'article de (de Lucena, Lima, Thomas, & Teichrieb, 2019). Il va retrouver les zones de l'images dans lesquelles sont situées les points d'intérêts du visage. Après cela, un set de points représentant un visage basique est projeté sur l'image. Le réseau graphique viendra modifier ce dernier en utilisant les zones d'intérêts du visage extraites par le CNN afin d'adapter le set de points au visage dans l'image. Après cette étape, les points sont supposés suivre correctement les contours du visage. L'article de (Prados-Torreblanca et al., 2022) présente justement cette méthode. Cet article présente des résultats intéressants en termes de précision, on retrouve une erreur moyenne normalisée de 4% la position des points entre les prédictions du réseau et la réalité.

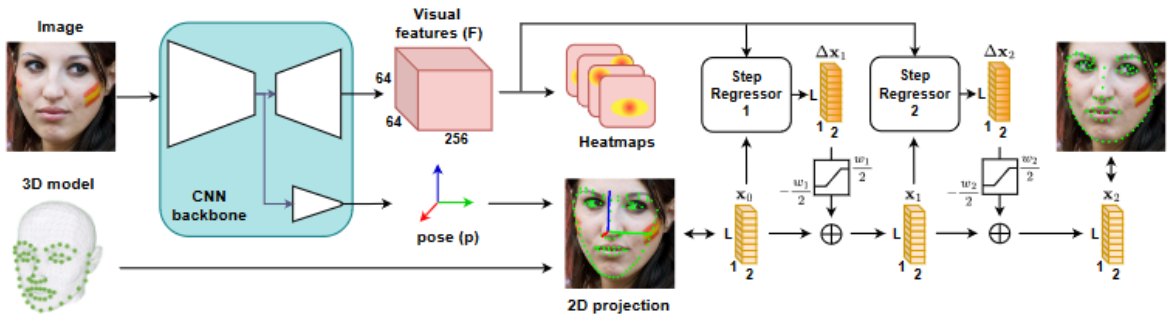

Figure 7 - Architecture de la méthode SPIGA tiré de (Prados-Torreblanca et al., 2022)

L'article de (de Lucena et al., 2019) présente un réseau CNN multitâche appelé MTCNN, basé sur les travaux de (K. Zhang, Zhang, Li, & Qiao, 2016). Ce réseau, divisé en trois parties, est capable, comme le montre la Figure 8 - Démonstration des étapes du réseau de (de Lucena et al., 2019), de proposer dans un premier temps des zones où se trouvent les différents points d'intérêts du visage, un grand nombre d'entre elles sont ainsi prédites mais peu précise. Une deuxième phase pour ensuite raffiner ces prédictions en gardant que les zones intéressantes. Enfin la dernière partie du réseau regarde ces différentes zones afin de placer précisément les points sur le visage.

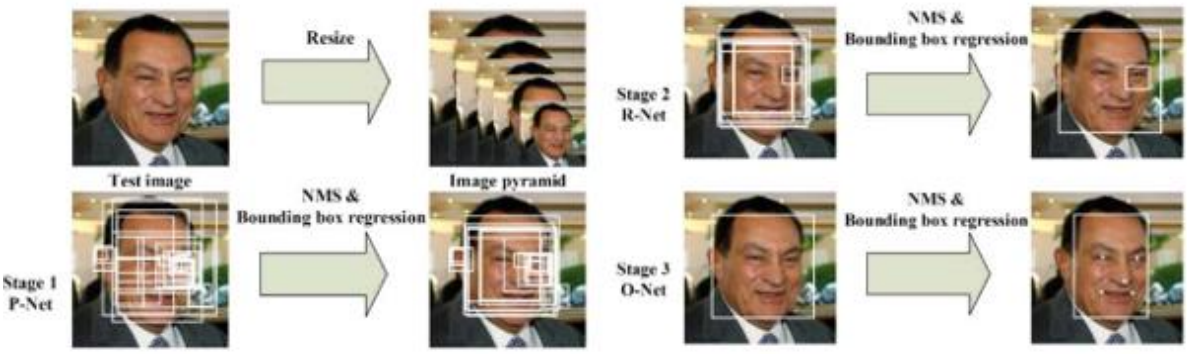

Figure 8 - Démonstration des étapes du réseau de (de Lucena et al., 2019)

1.2.2 Méthode de recalage

Une fois que l'on a pu retrouver des points du visage, il est possible de déterminer la position et l'orientation du visage dans l'image et donc d'y superposer le modèle 3D. Cette opération

est appelée recalage (Z. Zhang, 1993). Le recalage du modèle sur l'image va permettre de replacer le modèle 3D du visage dans la bonne position et orientation dans l'image. Une méthode existante pour retrouver l

Le problème de Perspective-n-Points (PnP) est un problème mathématique dans lequel on cherche initialement à retrouver la pose (position et orientation) d'une caméra dans un environnement connu à partir des points d'intérêt présent dans l'image qu'elle peut prendre depuis cette pose (Fischler & Bolles, 1981). On renverse généralement cette consigne afin de retrouver la position d'un objet connu dans une image. Dans ce problème, on n'a comme information uniquement la photo prise par la caméra, la géométrie 3D d'un objet présent dans celle-ci. Le but de la résolution de ce problème est donc de trouver des correspondances entre les points visibles de l'objet dans l'image et sa géométrie 3D afin de déterminer sa pose dans l'image.

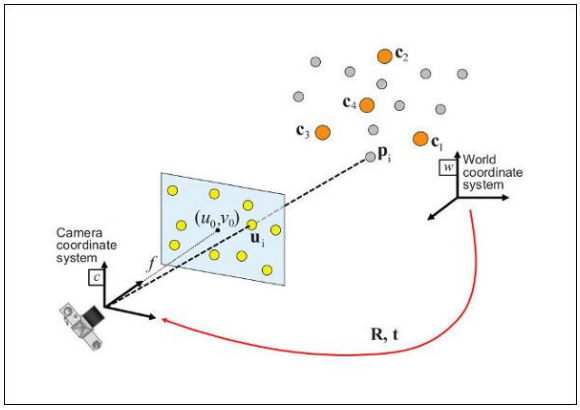

Figure 9 - Schéma présentant la méthode PnP tiré de (OpenCV, s.d.)

La Figure 9 représente le concept de la résolution de ce problème. On a un objet dont on connait les coordonnées 3D, exprimées dans un référentiel inconnu, de certains de ses points. On a également une image sur laquelle on peut voir les points décrits plus tôt dans le référentiel de la caméra. La résolution du problème PnP nous permettra donc d'obtenir la rotation et la translation nécessaire afin de passer du référentiel de la caméra à celui de l'objet.

Dans le cas théorique, il nous faut retrouver un minimum de 3 points visibles dans l'image correspondant à des points 3D de l'objet dans cette dernière afin d'obtenir un nombre fini de solutions au problème PnP. Résoudre le problème à trois points se fait en réalisant des projections entre le centre de projection de la caméra et les ponts dans l'image afin de contraindre l'objet dans un nombre fixe de position. Ces projections nous donneront un ensemble d'équation dont la résolution déterminera les configurations possibles de l'objet. Mais il existe aussi plusieurs autres algorithmes pouvant utiliser plus de points afin de trouver une solution unique (Lepetit, Moreno-Noguer, & Fua, 2009), (Terzakis & Lourakis, 2020). Ces méthodes ressortiraient les informations de pose entre l'image et le modèle 3D avec une erreur nulle à condition que les points sur l'image soient placés exactement aux mêmes endroits que sur le modèle.

Cependant, dans un cas plus pratique, les points de l'objet ne seront jamais parfaitement placés dans l'image, ce qui va entrainer une erreur sur la pose de l'objet dans cette dernière. (Lepetit et al., 2009) présente plusieurs algorithmes de résolution du problème PnP et l'influence de cette erreur de placement. Ce qui en ressort est que l'erreur augmente sur la pose en fonction de l'erreur sur le placement des points mais de manière moins significative si le nombre de points est suffisamment élevé. Il est également possible d'utiliser des algorithmes d'optimisation afin de choisir au mieux les points à utiliser pour réduire l'erreur.

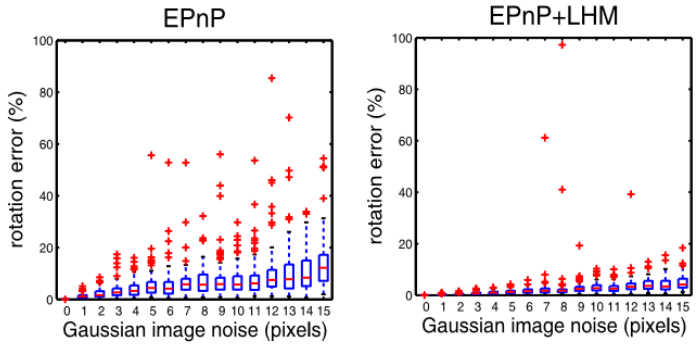

Figure 10 - Évolution de l'erreur normalisée (en %) sur la rotation en fonction de l'erreur de placement des points dans l'image (en px) tiré de (Lepetit et al., 2009)

La Figure 10 montre la relation entre la précision du placement des points sur l'image et l'erreur sur la rotation obtenue en sortie d'algorithme. Comme on peut le voir ici la résolution du problème PnP est sensible au bruit en position des points dans l'image. En effet, moins le placement d'un point dans l'image est précis, moins la correspondance entre ce dernier et le modèle géométrique de l'objet sera forte ce qui laissera place à l'erreur.

Utiliser cette méthode dans notre cas demanderait de réaliser une étape de détection de visage très précise afin de placer un certain nombre de points sur le visage dans l'image avec une erreur faible car dans le cas contraire cette méthode deviendrait également inefficace. Il nous faudrait donc entrainer un premier réseau pour générer un modèle 3D de visage et un autre pour réaliser la détection du visage, tous deux pouvant augmenter l'erreur de recalage si leur résultat n'est pas suffisamment précis.

1.3 Bases de données

Dans les dernières étapes on a vu que réaliser des modèles statistiques ou entraîner des réseaux de neurones sont les solutions généralement retenues pour ce type de projet. Cependant l'implémentation de ces derniers demandent des données. Voici donc quelques exemples de bases de données qui ont été utilisées dans la littérature.

1.3.1 WFLW

Cette base de données conçue par l'équipe de (Wu et al., 2018) regroupe 10 000 images, dont 7 500 sont conservée en tant que base de test, contenant au moins un visage. Ces images sont prises dans des situations diverses (luminosité élevée, visage maquillé, image floue, …) et les visages sont annotés de 98 points d'intérêt en 2D. Les coordonnées de ces points sont exprimées en pixels. Cette base de données est régulièrement utilisée pour réaliser la détection de visage comme dans l'article de (Prados-Torreblanca et al., 2022).

1.3.2 AFLW2000-3D

Cette base de donnée (Zhu, Lei, Liu, Shi, & Li, 2019) contient 2000 sets de données provenant de la base de données Annotated Facial Landmarks in the Wild (Gupta, Thakkar, Gandhi, & Narayanan, 2018). Les données ont été augmentée par rapport à la base de données de départ afin d'ajouter au image de visage et sets de points d'intérêt 2D de départ, des modèles 3D de 68 points des visages ainsi que les informations de pose de ces derniers et leurs paramètres statistiques obtenus avec le modèle de (Paysan et al., 2009) présenté précédemment.

De plus, la base de données contient des codes qu'ils ont utilisés pour replacer les modèles 3D sur les images. Ces derniers contiennent les informations nécessaires pour reproduire le recalage des modèles 3D et donc expliquer le vecteur de pose. Ce dernier contient 7 paramètres tels que :

$$p = [\phi\ \gamma\ \theta\ t_x\ t_y\ t_z\ f] \tag{1.3}$$

, où $[\boldsymbol{\phi}\ \boldsymbol{\gamma}\ \boldsymbol{\theta}]$ représente la rotation du modèle suivant les angles de tangage, lacet et roulis respectivement, $[\boldsymbol{t_x}\ \boldsymbol{t_y}\ \boldsymbol{t_z}]$ représente la translation suivant les axes x, y et z et $\boldsymbol{f}$ est le facteur d'échelle.

La base de données n'indique pas l'unité dans laquelle sont exprimées les coordonnées des points des modèles 3D. En effet, les coordonnées suivant les axes x et y correspondent aux valeur en pixel de la position du visage dans l'image à laquelle le modèle 3D fait référence mais aucune mention de la dimension de profondeur n'est faite dans l'article (Zhu et al., 2019).

1.3.3 3DFAW

La première base de données 3D Face Alignment in the Wild a été conçue dans le cadre d'un premier challenge de reconstruction faciale (Jeni, Tulyakov, Yin, Sebe, & Cohn, 2016). Elle contient 23 000 sets de données contenant une image de visage et le modèle 3D correspondant composés de 66 points créés en annotant une série d'images 2D puis en réalisant une opération

de photogrammétrie pour obtenir le modèle 3D. Une autre base de données 3DFAW a par la suite été créée contenant des modèles 3D avec plus de points mais nous n'avons pas pu y avoir accès (Pillai et al., 2019).

1.4 Discussion de la revue

De notre revue de littérature il semblerait ressortir une méthode très bien établie et utilisée dans plusieurs domaines qui est d'entrainer un réseau de neurones à retrouver dans l'image d'un visage, ses paramètres statistiques et ses informations de pose. Cette méthode montre des résultats semblant plutôt satisfaisant dans la littérature et d'autres personnes du laboratoire l'utilisent ce qui simplifiera sa mise en place pour notre cas spécifique. La seule limite que l'on pourrait y trouver est sur les données à utiliser. En effet, nous avons besoin d'une base de données contenant toutes les informations nécessaires (image, modèle 3D et pose du visage). La base de données AFLW2000-3D présentée à la section 1.3.2 contient justement toutes ces données.

Une autre méthode pour le recalage des modèles 3D est effectivement utilisée dans la littérature consistant à retrouver les points du visage dans l'image et de les utiliser pour repositionner le modèle 3D correctement mais ces méthodes sont moins utilisées et le résultat de la pose dépend grandement de la précision du placement des points sur l'image.

De plus, le principe d'utiliser des paramètres statistiques afin de reconstruire un modèle de visage semble donner de bons résultats même sur des visages pouvant parfois être asymétriques dû à une condition physique (Nguyen et al., 2022). Cette application se rapprochant de l'objectif de ce projet, il semblerait intéressant de vérifier si ces résultats sont transposables à notre cas. L'utilisation d'une matrice de transformation pour pallier le problème des angles d'Euler réalisée par (J. Guo et al., 2020) semble également intéressante. Nous comparerons donc l'utilisation de ces deux représentations de la pose pour le recalage de modèles 3D de visages.

CHAPITRE 2

MÉTHODOLOGIE

L'objectif que l'on se fixe pour ce projet est la génération d'un modèle 3D de visage à partir d'une image et son recalage sur cette dernière. Ce chapitre présentera la solution retenue de la revue de littérature et les méthodes de validation qui y ont été appliquées afin d'arriver à ce résultat

2.1 Solution retenue

Ainsi, notre système va suivre la méthode présentée dans la Figure 11 dont nous présenterons chaque bloc à la suite de ce chapitre. On a en entrée une image d'un visage traitée par un réseau de neurones qui en ressortira les paramètres statistiques du visage dans l'image et ses informations de pose. Les paramètres statistiques sont ensuite utilisés par le modèle PCA que l'on a créé afin de générer un modèle 3D du visage. Le modèle est ensuite recalé sur l'image à l'aide des informations de pose.

On commencera créer un modèle PCA à l'aide des données d'AFLW2000-3D pour obtenir les paramètres statistiques extraits des modèles 3D de visages nécessaires à l'entrainement du réseau de neurones. Enfin, ce dernier réalise un entrainement d'un réseau ResNet50 (Figure 3) utilisant l'image d'entrée et les informations de pose et les paramètres statistiques des visages en sortie. La base de données utilisée pour entrainer le réseau de neurones est également AFLW2000-3D (Zhu et al., 2019).

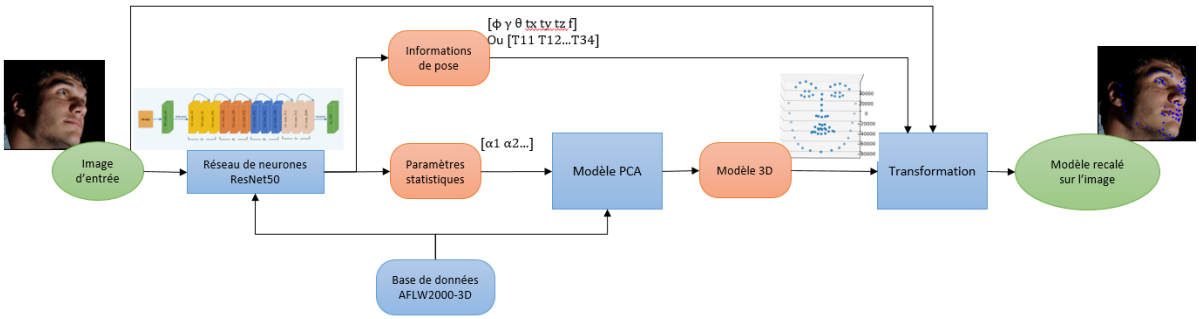

Figure 11 - Diagramme bloc de l'infrastructure de notre système

Dans ce projet, les codes ont été réalisés en langage Python, langage semblant très pertinent du fait de sa popularité dans le domaine de l'intelligence artificielle. Des bibliothèques sont déjà implémentées et des forums rapportent et documentent largement des problèmes que l'on pourrait rencontrer ce qui rendra l'implémentation plus simple.

2.2 Construction du modèle 3D

Cette partie présente comment le modèle PCA a été choisi et comment ce dernier parvient à générer des modèles de visage.

2.2.1 Modèle PCA

Afin d'implémenter le modèle PCA, on utilise les données de la base de données AFLW2000-3D présentée plus tôt. Comme dit précédemment, cette base de données contient des modèles 3D de visages composés de 68 points. Ce sont ces points qui sont utilisés pour créer le modèle PCA. Cependant, ces derniers sont stockés dans la position qu'ils ont dans leur image respective. Les modèles de visages n'étant pas alignés sur les mêmes axes, ils ne sont pas directement utilisables pour implémenter un modèle PCA. Il va donc falloir appliquer une transformation à nos différents modèles 3D pour les aligner. Cette transformation sera introduite dans une section plus tardive de ce rapport (voir section 2.3, équation (2.3)). Elle utilise les informations de pose associées au modèle 3D de visage qui sont présente dans la BD.

Une fois le modèle PCA entraîné, il suit une série de tests dans le but de valider ses capacités. L'entraînement du modèle est réalisé sur l'intégralité des données de la base de données exceptés 10 sets de données choisis aléatoirement qui nous serviront lors de la validation.

Ensuite, il suit les processus de réduction de dimension permettant de régler le nombre de vecteurs propres que l'on va pouvoir conserver dans le modèle. En effet, la majorité de l'information d'un modèle PCA est concentrée dans les premiers vecteurs propres. Il nous est donc nécessaire de trouver combien d'entre eux il est acceptable d'ignorer tout en gardant une reconstruction relativement précise. Pour cela on va se concentrer sur la quantité de variance expliquée par les vecteurs propres du modèle. On conserve les vecteurs propres nous permettant d'atteindre un taux de variance expliquée d'au moins 95%. Le modèle PCA ainsi établit permettra de reconstruire des modèles 3D de visages à partir de leurs paramètres statistiques de la manière suivante :

$$\boldsymbol{M = \mu + \alpha_0 \cdot A_0 + \alpha_1 \cdot A_1 + \cdots + \alpha_n \cdot A_n} \quad (2.1)$$

$$\boldsymbol{M = \mu + \alpha \cdot A} \quad (2.2)$$

Avec $\boldsymbol{\mu}$ $\boldsymbol{et}$ $\boldsymbol{A}$ représentant le visage moyen et la matrice contenant les n vecteurs propres conservés dans le modèle PCA et $\boldsymbol{\alpha}$ les paramètres statistiques décrivant le visage M dans le modèle.

2.3 Expression de la transformation appliquée aux modèles 3D

La transformation d'un modèle 3D est l'étape qui consiste à déplacer le modèle entre le référentiel de départ et un référentiel cible. Dans notre cas, les modèles de visages générés par le modèle PCA sont tous alignés dans le même référentiel. Il est donc nécessaire de réaliser cette opération de transformation afin de les superposer à leur image dans la bonne pose comme le montre la Figure 12.

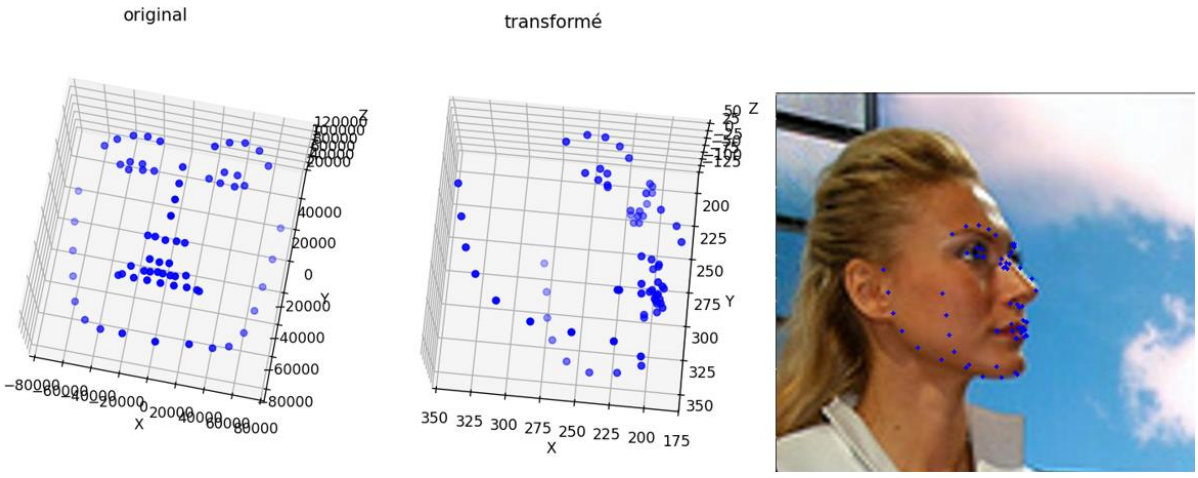

Figure 12 - Exemple de transformation : l'image de gauche représente un modèle 3D de visage dans l'espace dans une configuration neutre, l'image centrale représente le modèle 3D transformé et à droite est représenté le modèle transformé projeté sur son image de référence

Cette transformation est appliquée à un ensemble de points 3D en utilisant une matrice de transformation construite à partir d'un vecteur de pose de la même manière que dans la base de données AFLW2000-3D. Cependant, cette transformation n'est pas détaillée dans l'article de (Zhu et al., 2019) présentant la base de données. Mais cette dernière contient un fichier de code dans laquelle il est possible de reprendre les étapes nécessaires à la réalisation de la transformation.

Elle est réalisée en suivant consécutivement ces étapes :

- Rotation du modèle : Le modèle est orienté suivant le sens du visage dans l'image correspondante
- Mise à l'échelle : On utilise f pour changer la taille du modèle
- Translation : Le modèle est déplacé jusqu'à la position du visage dans l'image
- Retournement vertical : l'axe y des modèles 3D est croissant vers le haut (dans la direction allant du menton vers le front) tandis que l'axe y des images est orienté dans le sens inverse. Il faut donc retourner le set de points afin de bien le replacer sur l'image.

Chacune de ces étapes donne lieu à une matrice de transformation qui, placées à la suite les unes des autres permettent de mettre en équation le recalage du modèle 3D du visage à sa place dans l'image :

$$\boldsymbol{M_{image}} = \boldsymbol{T} \cdot \boldsymbol{M_{base}} \tag{2.3}$$

Avec,
$$\boldsymbol{T} = \boldsymbol{F}_{(h)} \cdot \boldsymbol{t}_{(t_x,t_y,t_z)} \cdot \boldsymbol{S}_{(f)} \cdot \boldsymbol{R}_{(\phi,\gamma,\theta)} \tag{2.4}$$

Où $\boldsymbol{M_{image}}$ et $\boldsymbol{M_{base}}$ sont les modèles exprimés respectivement dans les référentiels de l'image et le référentiel de base. Ces matrices sont de la forme :

$$\boldsymbol{M} = \begin{bmatrix} x_0 & x_n \\ y_0 & \dots \; y_n \\ z_0 & z_n \\ 1 & 1 \end{bmatrix}$$

Et, $\boldsymbol{F}, \boldsymbol{t}, \boldsymbol{S}, \boldsymbol{R}$ sont les matrices représentant les transformations subies par le modèle correspondant aux explications données plus tôt dans l'ordre inverse (**F** représente le retournement vertical et **R** la rotation du modèle).
Ces matrices sont de la forme :

$$\boldsymbol{F}_{(h)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{T}_{(t_x,t_y,t_z)} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{S_f} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{R}_{(\phi,\gamma,\theta)} = \boldsymbol{R}_{\phi} \cdot \boldsymbol{R}_{\gamma} \cdot \boldsymbol{R}_{\theta} \tag{2.6}$$

Avec,

$$\boldsymbol{R_{\phi}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{R_{\gamma}} = \begin{bmatrix} \cos(\gamma) & 0 & -\sin(\gamma) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\gamma) & 0 & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$\boldsymbol{R_{\theta}} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Réaliser la multiplication de ces matrices donnera la matrice de transformation pour passer du référentiel de base vers le référentiel de l'image. Pour faire la transformation inverse et donc

passer du référentiel de l'image au référentiel de base, il suffit de retrouver la matrice inverse de cette matrice de transformation. :

$$\boldsymbol{M_{base} = T^{-1} \cdot M_{image}} \tag{2.8}$$

2.4 Estimation des paramètres PCA et de pose par réseaux de neurones

Comme montré sur la Figure 11, le bloc réseau de neurones réalise l'estimation des paramètres statistiques du visage présent dans l'image ($\boldsymbol{\alpha}$ dans l'équation (2.2)) ainsi que ses paramètres de pose. Nous testons ici l'utilisation du réseau ResNet50 utilisé par (Nguyen et al., 2022). Ces réseaux prendront donc en entrée l'image contenant le visage à reconstruire et donnera en sortie un vecteur [$\boldsymbol{p\ \alpha}$] avec p les informations de pose et α les paramètres statistiques à donner au modèle PCA pour construire un modèle 3D du visage de l'image.

2.4.1 Expression des informations de pose et des paramètres statistiques

Les paramètres de pose extraits de la base de données sont écrits sous la forme d'angles d'Euler et sont donc sensibles au phénomène de Gimbal Lock qui apparaît lorsque deux des axes de rotations du modèle se retrouvent confondus (Hemingway & O'Reilly, 2018). Dans ce cas, une même configuration peut être représentée par plusieurs valeurs d'angles différentes. Ce qui peut poser problème au réseau qui pourrait prédire un set de valeurs d'angles représentant correctement la configuration mais n'étant pas identique aux angles de référence ce qui augmenterait l'erreur dans les prédictions suivantes. Pour régler ce problème (J. Guo et al., 2020) proposent d'utiliser la matrice de transformation directement, nécessitant 12 paramètres au lieu de 7 mais, en contrepartie, ne laissant pas la place à la confusion apportée par les angles d'Euler.

Ainsi, pour la suite du projet, la pose pourra être estimée soit en utilisant le vecteur de pose (voir équation (1.3)) soit en utilisant la matrice de transformation T de l'équation (1.2). On compare l'utilisation de ces deux derniers dans nos tests.

Notre réseau de neurone prend donc en entrée une image de visage et reourne en sortie un vecteur : $\boldsymbol{v} = [\boldsymbol{p}\ \boldsymbol{\alpha}]$ avec **p** étant le vecteur de pose utilisant les angles d'Euler comme à l'équation (1.1), le ResNet50 avec **p** étant la matrice de transformation **T** de l'équation (1.2).

Les paramètres statistiques sont eux obtenus par l'utilisation du modèles PCA créé dans la section 1.1.1 sur les modèles de visages d'AFLW200-3D.

Les données utilisées pour l'entrainement de ces réseaux proviendront de la base de données AFLW2000-3D lors de différents tests. Les données de cette base seront divisées aléatoirement en une base d'entrainement, de validation et de test correspondant respectivement à 70%, 15% et 15% de AFLW2000-3D.

La fonction de perte utilisée pour l'entrainement est l'erreur carrée moyenne ou Mean Squared Error (MSE). Cette fonction de perte va permettre d'exacerber les différences aberrantes entre les prédictions et les valeurs de référence (Y. Guo, Zhang, Cai, Jiang, & Zheng, 2018).

$$L = \sum_i \left| v_i^{ref} - v_i^{pred} \right|^2 \tag{2.9}$$

Où $\boldsymbol{v^{ref}}$ est le vecteur de sortie de référence contenant les informations de pose et les paramètres statistiques du visage et $\boldsymbol{v^{pred}}$, le vecteur prédit par le modèle.

Le taux d'apprentissage utilisé lors de l'entraînement est variable. Il diminue au cours des epochs afin de réaliser des modifications importantes des poids dans les premières epochs mais d'affiner ces changements sur la fin, lorsque l'on se rapproche d'un minimum local de la fonction de perte. La fonction décrivant la variation de ce taux est la suivante :

$$\begin{cases} lr_i = lr_{i-1} \cdot e^{-0.1}\ ; i > 10 \\ lr_i = lr_0 = 0.001\ ;\ i \leq 10 \end{cases} \tag{2.10}$$

Où i est le numéro de l'epoch actuelle et lr_0 le taux d'apprentissage initial. La diminution du taux d'apprentissage s'applique à chaque epoch à partir de la dixième. Ces données sont les données de base utilisées par la bibliothèque d'apprentissage machine Tensorflow (Tensorflow, s.d.).

L'entraînement se déroule sur un maximum de 200 epochs mais est arrêté avant par un planificateur qui, à la fin de chaque epoch, vérifie l'amélioration des résultats sur la fonction de perte entre l'epoch qui vient de se terminer et les 5 précédentes. Si la perte n'a pas diminué, l'entraînement est arrêté.

2.4.2 Prétraitement des données

Dans un premier temps on va regrouper tous nos paramètres utiles dans un même fichier afin d'en simplifier la récupération. On récupère les points 3D et leurs informations de pose d'AFLW2000-3D. Ensuite, on réaligne les points 3D en utilisant les poses de manière à ce qu'ils soient placés comme lors de la création du modèle PCA (voir section 2.4.1). Enfin, on en ressort les paramètres statistiques à l'aide du modèle PCA que l'on aura sélectionné après l'étape de validation.

Ensuite, on créera les vecteurs de sortie que le réseau utilisera lors de l'apprentissage. Les vecteurs auront la forme suivante : $[\boldsymbol{\phi\ \gamma\ \theta\ t_x\ t_y\ t_z\ f\ \alpha_0\ \alpha_1 \dots \alpha_n}]$,où les 7 premiers paramètres reprennent ceux du vecteur de pose et les suivants sont les n paramètres statistiques conservés après la réduction de paramètres du modèle PCA. Si on veut prédire une matrice de transformation et non un vecteur de pose, on recrée cette dernière avec les 7 premiers paramètres du vecteur précédent en suivant la méthode de transformation de la section 2.3 et l'on remplace les paramètres du vecteur par la matrice de transformation ainsi créée. Le vecteur de sortie du réseau aura donc pour forme $[\boldsymbol{T\ \alpha_0\ \alpha_1 \dots \alpha_n}]$où T est un vecteur comprenant les 12 paramètres de la matrice de transformation telle que construite à l'équation (2.4).

Une fois ces vecteurs construit, on réalise une opération de normalisation par Z-score sur chacun des champs du vecteur (Cabello-Solorzano, Ortigosa de Araujo, Peña, Correia, & J. Tallón-Ballesteros, 2023). En récupérant tous les vecteurs ainsi créés pour la base d'entraînement, on détermine la moyenne et l'écart-type de chaque paramètre pour réaliser l'opération suivante :

$$p_{norm} = \frac{p - \mu}{\sigma} \tag{2.11}$$

Où μ est la moyenne et σ l'écart-type de ce paramètre dans la base d'entraînement.

Cette opération est nécessaire du fait que les différents champs du vecteur de sortie se trouvent de base dans des ordres de grandeurs différents. En effet, les translations sont généralement exprimées en pixels et peuvent facilement se situer a des valeurs absolues de 10^2px tandis que les rotations sont exprimées en radians, leurs valeurs sont donc plus situées entre $-\pi$ et π (rad). Quant aux paramètres statistiques, on s'attend à ce qu'ils se situent en moyenne autour de 0 mais leur écart-type peut être bien plus grand.

Cela pose des problèmes car lors du calcul de l'erreur, les erreurs que le réseau fera sur les caractéristiques à plus hautes valeurs auront un impact plus fort sur la modification des poids que celles sur les caractéristiques à valeurs plus faibles. Par conséquent, la normalisation fait en sorte que chaque champ du vecteur de sortie ait autant d'importance que les autres dans le calcul de l'erreur (Cabello-Solorzano et al., 2023).

2.4.3 Fichiers de configuration

Lors de la phase d'implémentation un certain nombre de tests différents vont devoir être lancés. Cependant, réécrire et/ou modifier le code pour chaque test prend du temps et augmente le risque d'erreur lors du lancement d'un entraînement de réseau de neurones. Or un entraînement de ce type est également coûteux en temps et il est probable que les erreurs potentielles incorporées lors des modifications du code ne soient remarquées qu'à la fin de l'entraînement ce qui rendrait ce dernier inutilisable.

Par conséquent le code a été implémenté en utilisant des fichiers de configurations qui permettront de changer les paramètres de l'entrainement sans modifier le code. Il modifiera les paramètres suivants :

- La base de données utilisée et son lieu de stockage
- Les pourcentages de répartition des données entre entraînement, validation et test
- Le nombre maximum d'epochs pour l'entrainement
- Le réseau de neurone utilisé
- Son taux d'apprentissage
- La méthode de représentation des informations de pose
- Le lieu de stockage des données pour le test

2.5 Méthodes de Validation

Une fois ces solutions implémentées il est nécessaire de pouvoir valider les résultats obtenus par le modèle PCA pour ce qui est de la génération de modèles 3D et ceux obtenus par les réseaux de neurones pour l'estimation des informations de pose et des paramètres statistiques des visages.

2.5.1 Validation du modèle PCA

La validation du modèle PCA se fait en réalisant les tests de compacité, généralisation et spécificité de (Heimann & Meinzer, 2009) présentés à la section 1.1.3.1.

2.5.2 Validation des réseaux de neurones

Pour valider les entrainements de réseaux, on commencera par regarder les courbes de perte sur les bases d'entrainement et de validation. Elles sont censées être décroissantes et converger.

Ensuite, pour valider les résultats, on regardera les différents modèles 3D de visages que l'on peut obtenir à partir des sorties du réseau. On veut valider ses capacités à extraire d'un côté les informations de pose et de l'autres les paramètres statistiques du visage dans l'image d'entrée.

Comme indiqué dans la section 1.1.6, les articles étudiés ne différencie par l'erreur liée à l'estimation de la pose à celle liée à la reconstruction du modèle. Il semblait donc intéressant de s'intéresser à cela.

C'est pour cela que l'on va réaliser trois tests qui nous permettront de valider la précision de la reconstruction et de la pose. Pour le premier, on commence par reconstruire des modèles de visages de la base de test à l'aide du modèle PCA conçut précédemment puis on replacera ces derniers sur le référentiel de l'image pour ensuite calculer l'erreur moyenne absolue (MAE) entre les points du modèle original et ceux juste reconstruit (Hashemibakhtiar et al., 2024). Comme expliqué dans la section 1.3.2 présentant la base de donnée utilisée tout au long du projet, l'unité dans laquelle sont exprimées les coordonnées des points des modèles de visages n'est pas claire (on peut exprimer les coordonnées x et y en pixel mais pas la coordonnée suivant l'axe z). Par conséquent, la MAE est exprimée sans unité dans les résultats. Le calcul de cette erreur se fait sur l'intégralité des données de la base de test pour trouver un résultat moyen général et que sa valeur ne dépende pas de la qualité de la prédiction choisie. Ce premier test, mentionné par la suite de ce rapport comme test de reconstruction, permettra de juger les capacités du réseau à extraire les paramètres statistiques du visage.

Notre deuxième test vient examiner la pose estimée par le réseau. Pour celui-ci, nous utiliserons les modèles 3D provenant des données de référence auxquels nous appliquerons, d'un côté une transformation utilisant les vraies informations de pose et de l'autre, celle prédites par le réseau. On calculera une nouvelle fois une MAE qui nous permettra d'apprécier la qualité des données de pose retrouvée par le réseau. Ce test sera mentionné comme test d'estimation de pose

Enfin, le dernier test est plus global. Lors de celui-ci, il sera possible de comparer le modèle totalement prédit par le réseau, construit avec les paramètres que ce dernier a trouvé dans l'image et transformé avec les informations de pose et les données de référence pour obtenir une appréciation globale du réseau. Il se rapproche donc plus de ce que l'on retrouve dans la

littérature. On calculera une dernière MAE à ce moment pour juger de la qualité du réseau dans son ensemble. Ce test est mentionné par la suite comme test global.

CHAPITRE 3

RÉSULTATS

3.1 Validation des modèles statistiques

Cette section présente les résultats obtenus lors de la validation du modèle PCA entrainé sur les données d'AFLW2000-3D. Du fait que les modèles 3D de visages n'aient pas d'unité spécifiée, les MSE calculées ici sont exprimées sans unité.

3.1.1 Test de compacité

La Figure 13 présente les résultats obtenus lors du test de compacité réalisé sur le modèle PCA de visage établis avec les modèles 3D de visages de AFLW2000-3D. Les points du graphe représentent la moyenne d'erreur quand les barre d'erreur représente l'écart-type. On commence par remarquer sur la Figure 13 que le modèle retourne une erreur convergeant vers 0 lorsque l'on augmente le nombre de vecteurs propres conservés.

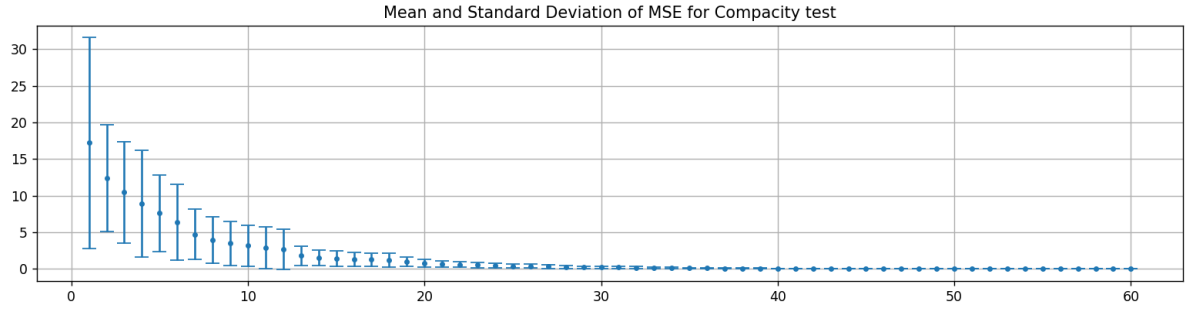

Figure 13 - Résultats du test de compacité sur les deux modèles PCA (seuls les 60 premiers vecteurs sont présentés pour une meilleure lisibilité du graphe)

3.1.2 Test de généralisation

La Figure 14 présente les résultats obtenus lors du test de généralisation réalisé sur le modèle PCA de visage établis avec les modèles 3D de visages de AFLW2000-3D. Sur cette figure, on

remarque également une erreur décroissante et convergeant vers 0 quand le nombre de VP augmente.

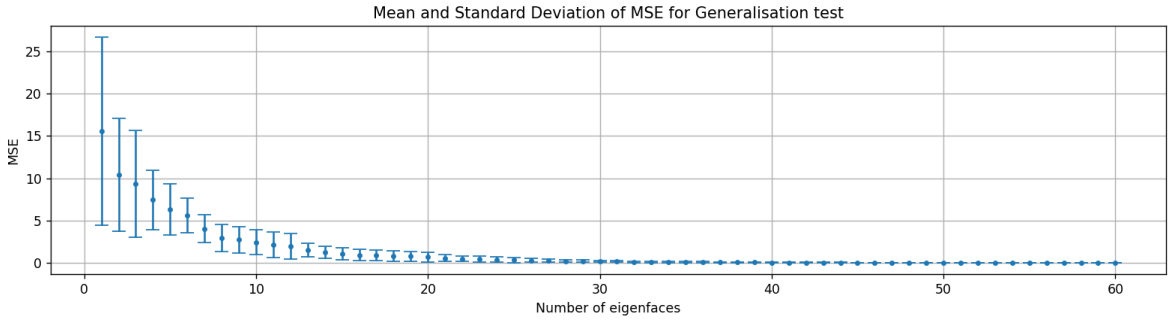

Figure 14 - Résultats du test de généralisation sur les deux modèles PCA (seuls les 60 premiers vecteurs sont présentés pour une meilleure lisibilité du graphe)

3.1.3 Test de spécificité

3.1.3.1 Génération de visage

En suivant la méthode présentée dans la section 1.1.5.3, notre modèle est capable de générer différents visages. La Figure 15 présente les résultats de la génération de 10 modèles 3D de visages réalisés par le modèle PCA.

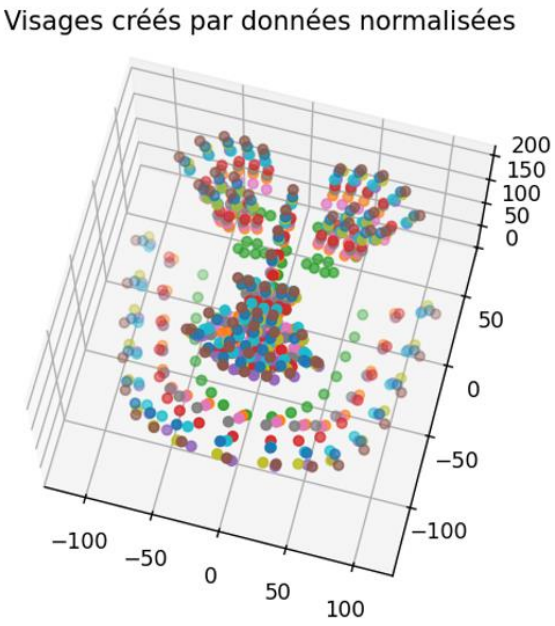

Figure 15 - Création de visage aléatoire à partir des modèles statistiques normalisé (à gauche) et non normalisé (à droite)

Comme on peut le voir de manière qualitative, les modèles 3D générés par notre modèle PCA ressemblent bien à des visages tous différents. Certains, comme le vert, sont plus petits que d'autres, ou encore ont une mâchoire plus carrée comme pour le rose.

3.1.3.2 Résultats du test de spécificité

La Figure 16 présente les résultats obtenus lors du test de spécificité réalisé sur le modèle PCA de visage établi avec les modèles 3D de visages de AFLW2000-3D

L'erreur semble rester, comme attendu, relativement stable avec une MSE oscillant majoritairement entre 2.5 et 5.

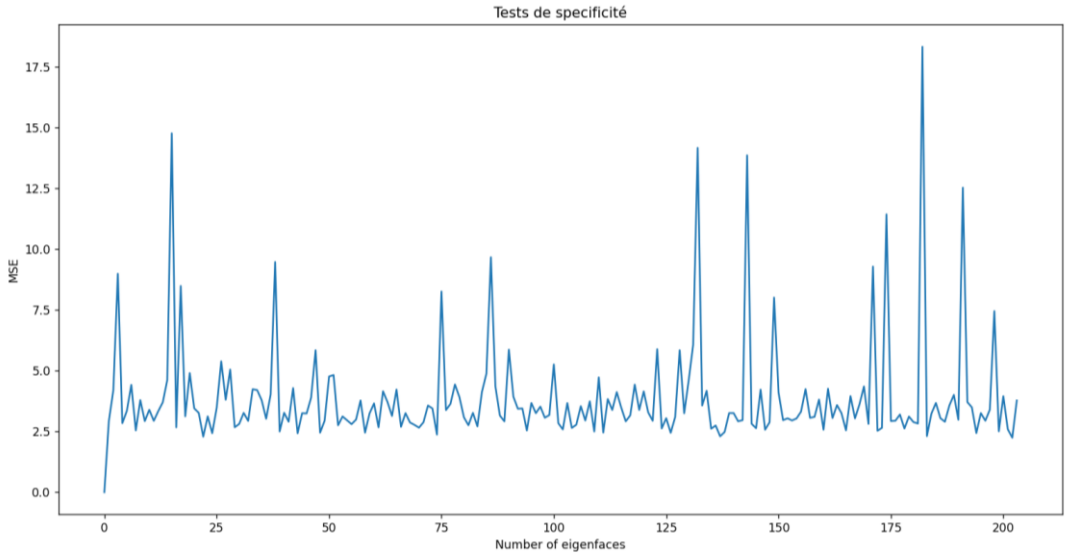

Figure 16 - Résultats du test de spécificité sur les modèles PCA normalisé et non normalisé

3.1.4 Réduction de paramètres

En effet, lors de l'entraînement de ce dernier, les vecteurs propres sont classés par ordre de pertinence. On regarde donc ici l'évolution de la variance expliquée en fonction du nombre de vecteurs propres conservés.

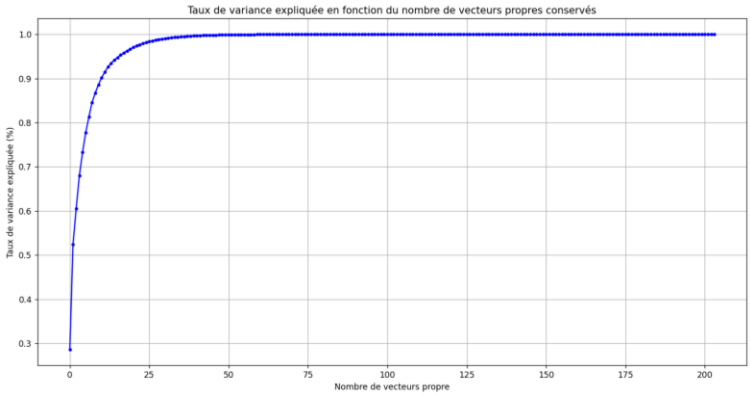

Figure 17 - Taux de variance expliquée en fonction du nombre de vecteurs propres

Comme prévu, l'évolution du taux de variance expliquée augmente de manière exponentielle en convergeant vers 1. On peut ressortir plusieurs valeurs de cette courbe telles que :

Tableau 1 - Données de variances expliquées en fonction du nombre de vecteurs propres

Nombre de VP	2	7	17	30	50
Variance expliquée	52%	81%	95%	99%	99.9%

On remarque sur le Tableau 1 que les 95% de la variance des données sont expliqués par les 17 premiers vecteurs propres sur les 204 totaux du modèle et qu'il en faut 14 de plus pour augmenter ce nombre de 4%. Ainsi, conserver uniquement les 17 premiers vecteurs propres du modèle semble un bon compromis entre précision et complexité de la reconstruction.

Quelques tests de reconstruction ont également pu être réalisés sur les données de test de la base de données afin de vérifier si la perte d'information entrainée par cette réduction est importante.

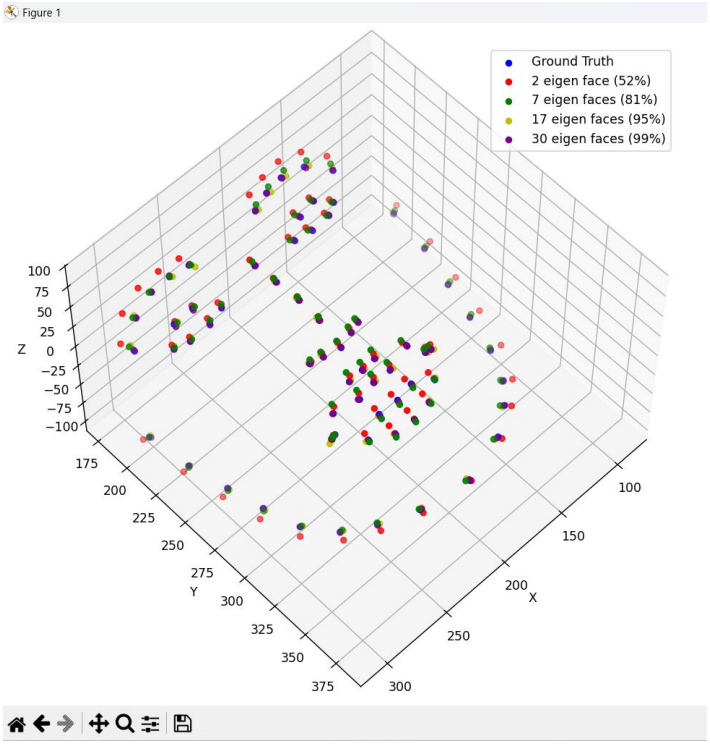

Figure 18 - Démonstration qualitative de la précision du placement des points en fonction du nombre de vecteurs propres conservés. Entre parenthèse dans la légende se trouve le taux de variance expliqué par le modèle PCA conservant le nombre de VP (eigen faces) indiqué

Sur la Figure 18, on a pris un modèle de visage de la base de données à partir duquel on a déterminé les paramètres statistiques grâce au modèle PCA en variant le nombre de vecteurs propres conservés. On a ensuite reconstruit le modèle du visage en utilisant ces différents sets de paramètres statistiques pour ainsi déterminer l'erreur de reconstruction induite par le modèle PCA lorsqu'il n'utilise pas l'ensemble des vecteurs propres. Le modèle 3D présenté en rouge sur la Figure 18, est reconstruit avec le modèle PCA conservant 2 VP. Respectivement en vert, jaune et violet, le modèle PCA conserve 7, 17 et 30 VP pour reconstruire le modèle 3D. Enfin, le modèle de référence est représenté en bleu. Ainsi, on peut voir que les points du modèle rouge sont les plus éloignés du modèle 3D de référence, les points verts semblent proches mais on peut toujours remarquer quelques erreurs et les points jaunes et violet semblent les plus proches de la réalité.

Voici les valeurs d'erreur quantitatives obtenues en calculant l'erreur absolue moyenne (MAE) point à point sur la base de test pour ces mêmes modèles.

Tableau 2 - Erreur moyenne absolue d'un point d'un modèle de visage en fonction du nombre de vecteurs propres utilisé pour générer ce modèle

Nombre de VP	2	7	17	30	50
Erreur moyenne	6.8	4.5	2.2	0.7	0.5

On remarque sur le Tableau 2 que la MAE diminue avec l'augmentation du nombre de VP conservés par le modèle PCA. Cette diminution est de plus en plus faible le plus on conserve de VP.

3.2 Estimation des paramètres de pose et statistiques par réseaux de neurones

Cette section présente les résultats des entrainements de réseaux de neurones pour l'estimation des paramètres de pose et des paramètres statistiques du visage à partir de son image sur une base de test composée de 15% d'AFLW2000-3D soit 300 ensembles de données. omme présenté dans la section 2.4, On entraîne un réseau ResNet50 dans deux configurations afin de comparer l'utilisation d'une matrice de transformation appelé RéseauA contre l'utilisation d'un vecteur de pose utilisant les angles d'Euler appelé RéseauB.

Pour chaque configuration on commencera par porter notre attention sur les courbes de perte issues de l'entrainement puis on présentera, les résultats quantitatifs des trois tests présentés dans la section 2.5.2, les tests de reconstruction, de recalage et un test global. Enfin des exemples qualitatifs tirés du test global seront également présentés.

3.2.1 Entrainement du RéseauA

Dans cette configuration le réseau entrainé est le ResNet50. Il prend en entrée une image et retourne un vecteur [p α] où p reprend la matrice de transformation telle que présentée à l'équation .

Ici, l'entraînement s'est arrêté après 67 epochs. La courbe de perte à la Figure 19 montre la convergence du réseau à une MSE de 0.02 sur la base d'entraînement contre 0.8 sur la base de validation. Pour ce qui est de la courbe de précision, on voit que celle de la base de validation suit celle de la base d'entrainement sur les premières epochs l'augmentation de sa précision s'arrête rapidement oscillant autour de 0.22 tandis que la précision du réseau sur la base d'entrainement converge autour de 0.8.

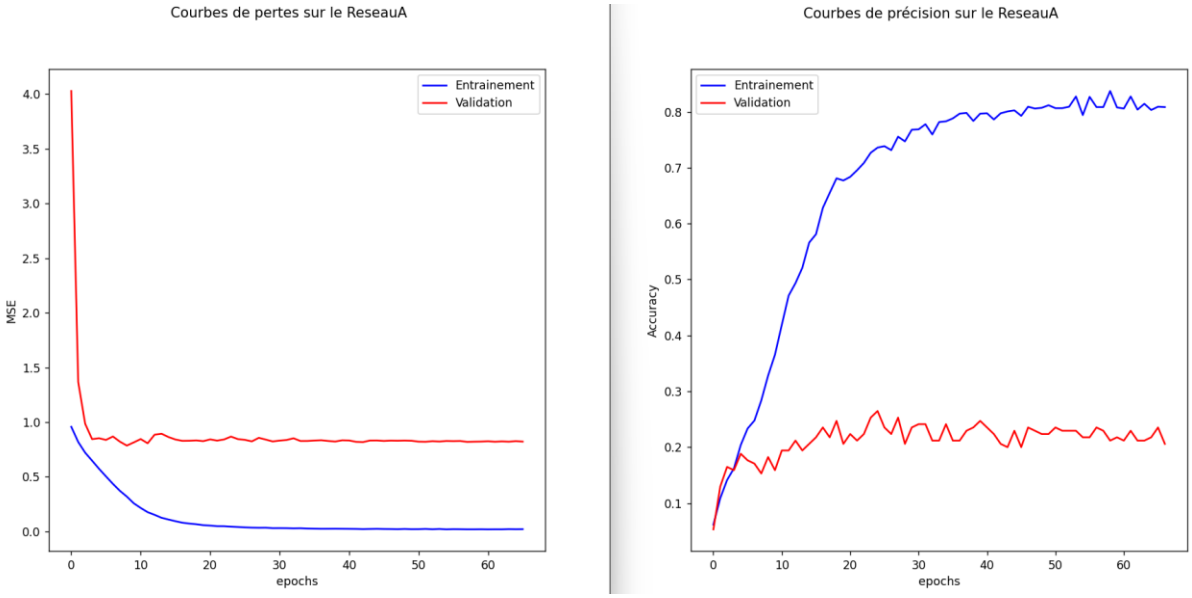

Figure 19 - Courbes de la fonction de perte (à gauche) et de précision (à droite) sur la base d'entrainement (en bleu) et de validation (en rouge)

3.2.1.1 Résultats quantitatifs

On a calculé, l'erreur en comparant la position des points 3D des modèles recréés à partir du réseau de neurones au modèle de référence. La MAE est réalisée point par point sur les coordonnées 3D. Comme expliqué dans la section 2.5.2 la MAE est sans unité car alors que les coordonnées x et y des modèles de visage sont exprimées en pixel, la coordonnée en z n'a pas d'unité précisée.

Ainsi, à la fin de l'entrainement, le réseau donne, sur la base de test les résultats que nous présentent le

Tableau 3. On remarque ici également que le test de reconstruction donne des erreur plus faibles (erreur moyenne de 5.7) que les deux autres tests qui nous présentent des résultats plus élevés. De plus, les résultats du test de recalage sont très proches de ceux du test global.

Tableau 3 - Résultats quantitatifs des tests de reconstruction, recalage et global de l'entrainement du ResNet50 en utilisant la matrice de transformation pour représenter la pose

Test	MAE	Min	Max
Reconstruction	5.7 ± 2.4	1.7 ± 1.1	15.6 ± 6.0
Estimation de pose	28.9 ± 16.4	6.1 ±3.6	139.6 ± 39.1
Global	29.4 ± 16.4	7.8 ± 3.6	142.8 ± 38.2

3.2.1.2 Résultats qualitatifs

Voici maintenant quelques exemples de résultats obtenus lors de cet entrainement. Les points bleus représenteront le modèle 3D de référence tandis que les points rouges montrent celui qui a été reconstruit à l'aide des paramètres statistique et les informations de pose estimées par le réseau de neurones.

La Figure 20 montre le meilleur résultat (à gauche) et le pire (à droite) rencontrés dans la base de test. On y voit que pour le sujet obtenant les meilleurs résultats, le modèle de visage généré grâce au réseau de neurones suit relativement bien le modèle de référence. On remarque un léger décalage de ce dernier vers la gauche comparé au modèle de référence. D'un autre côté, pour l'exemple avec les pires résultats, le réseau n'a pas l'air capable de générer un modèle suivant le modèle de référence. Le visage créé n'est pas placé dans la même orientation et semble légèrement déformé par rapport au modèle 3D de référence.

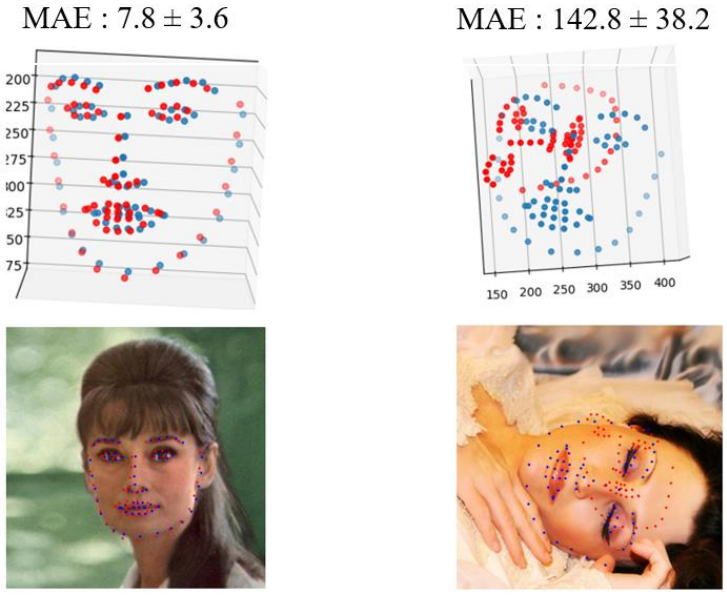

Figure 20 - Résultats minimum (à gauche) et maximum (à droite) sur la base de test représentation des modèles 3D sur un graphe (en haut) puis projection desdits modèles sur leur image de référence (en bas)

La Figure 21 montre des exemples de résultats obtenus lors des tests qui se situent plus proches de la moyenne présentée lors des résultats quantitatifs (voir

Tableau 3). En rouge sont représentés les points des modèles 3D générés par le RéseauA et en bleu, les modèles de références sur un graphe 3D puis la projection de ces points sur leurs images de référence.

Premièrement, On peut remarquer que les points des modèles générés des sujets à gauche de la Figure 21 suivent relativement bien le visage et que, pour les deux autres sujets les modèles générés sont placés dans une orientation plutôt proche de celles de référence. Cependant, on peut remarquer une déformation de certains modèles générés, principalement marquée sur le sujet en haut à droite de la figure. En effet, on aperçoit que le modèle obtenu est bien moins allongé que le modèle de référence. La distance entre le nez et l'arrière de la mâchoire est bien plus courte sur le modèle généré. Le modèle en bas à droite de la Figure 21 semble également légèrement déformé, la mâchoire du modèle 3D généré est plus étroite que celle du modèle de référence.

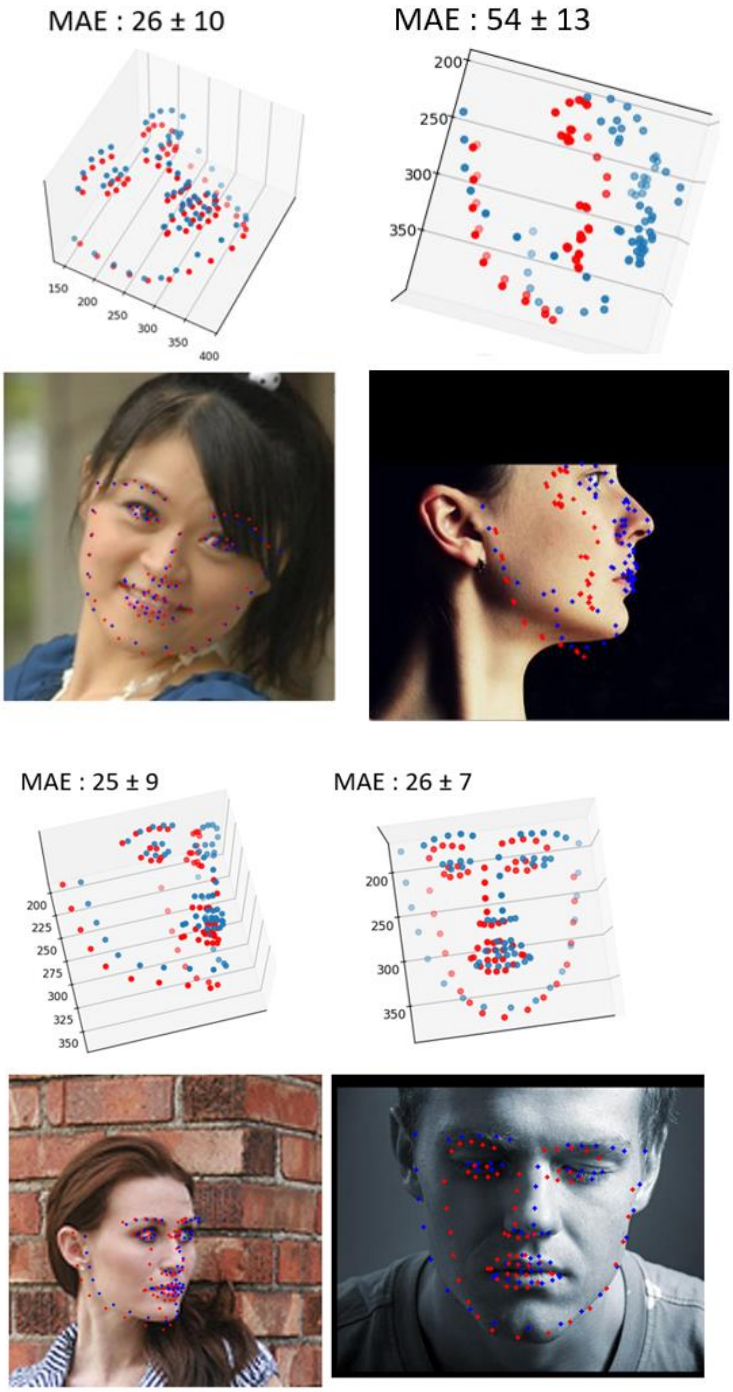

Figure 21 - Exemples de résultats proches de la moyenne obtenus sur la base de test

3.2.2 Entraînement du RéseauB

Dans cette configuration le réseau entrainé est, comme pour le test précédent, un ResNet50. Il prend en entrée une image et retourne un vecteur [p α] où p reprend le vecteur de pose utilisant les angles d'Euler tel que présentée à l'équation 1.1 plutôt que la matrice de transformation utilisée lors du test précédent.

Ici, l'entraînement s'est arrêté après 73 epochs. La courbe de perte à la Figure 22 montre la convergence du réseau à une MSE de 0.04 sur la base d'entraînement. La perte sur la base de validation suit celle sur la base d'entrainement avant de se remettre à augmenter convergeant autour de 1.1. Pour les courbes de précision, sur la base de validation on peut voir une augmentation légère avant de converger autour de 0.13 tandis que la précision sur la base d'entrainement converge vers 0.8.

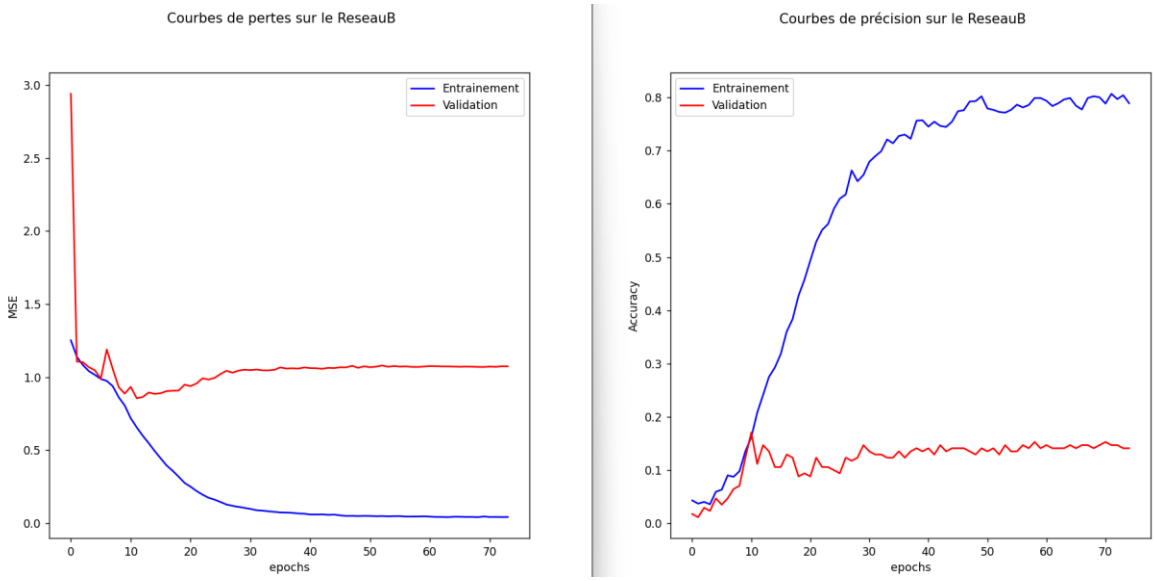

Figure 22 - Courbes de la fonction de perte (à gauche) et de précision (à droite) sur la base d'entrainement (en bleu) et de validation (en rouge) en fonction du nombre d'epochs

3.2.2.1 Résultats quantitatifs

Le Tableau 4 nous présente les résultats quantitatifs de l'entrainement réalisé sur le RéseauB. On remarque ici également que le test de reconstruction donne des erreur plus faibles (erreur moyenne de 6.7) que les deux autres tests qui nous présentent des résultats plus élevés. De plus, les résultats du test de recalage sont très proches de ceux du test global.

Tableau 4 - Résultats quantitatifs des tests de reconstruction, recalage et global de l'entrainement du ResNet50 en utilisant le vecteur de pose (Angles d'Euler)

Test	MAE	Min	Max
Reconstruction	6.7 ± 2.8	2.0 ± 1.2	19.9 ± 11.7
Estimation de pose	59.4 ± 32.3	15.6 ± 9.1	244.7 ± 47.3
Global	59.7 ± 32.3	15.6 ± 7.4	241.8 ± 67.5

3.2.2.2 Résultats qualitatifs

Voici quelques exemples de résultats obtenus lors de cet entrainement. Les points bleus représentent les modèles 3D de référence et les points rouges, ceux reconstruits à l'aide des paramètres statistique et les informations de pose estimées par le réseau de neurones.

La Figure 23 montre le meilleur résultat obtenu lors de cet entrainement (à gauche) et le pire (à droite). On remarque que pour le pire résultat, le modèle généré n'est pas du tout dans la même pose que le modèle de référence. On remarque également que la mâchoire du modèle généré pour le meilleur résultat est positionnée légèrement plus haut que le modèle de référence mais suit relativement bien le modèle.

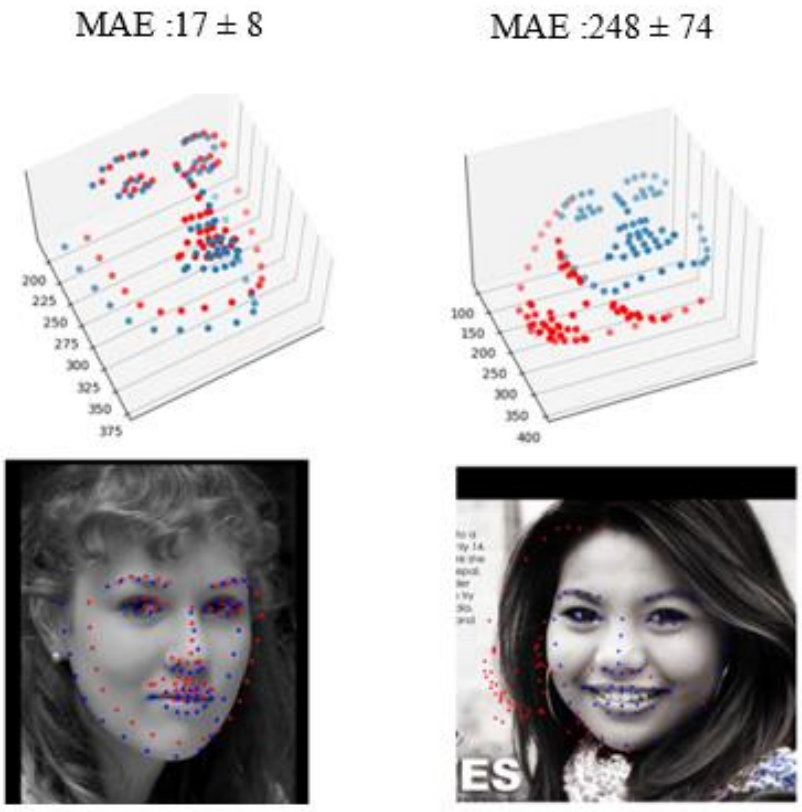

Figure 23 - Résultats minimum (à gauche) et maximum (à droite) sur la base de test représentation des modèles 3D sur un graphe (en haut) puis projection desdits modèles sur leur image de référence (en bas)

La Figure 24 montre des exemples de résultats obtenus lors des tests globaux qui se situent plus proches de la moyenne présentée lors des résultats quantitatifs (voir Tableau 4). On remarque que, pour certains exemples tels que le sujet en haut à gauche, on a un modèle généré orienté dans la même direction que le modèle de référence. Cependant, les exemples situés en bas de la figure montrent des modèles généré mal orientés par rapport à leurs références.

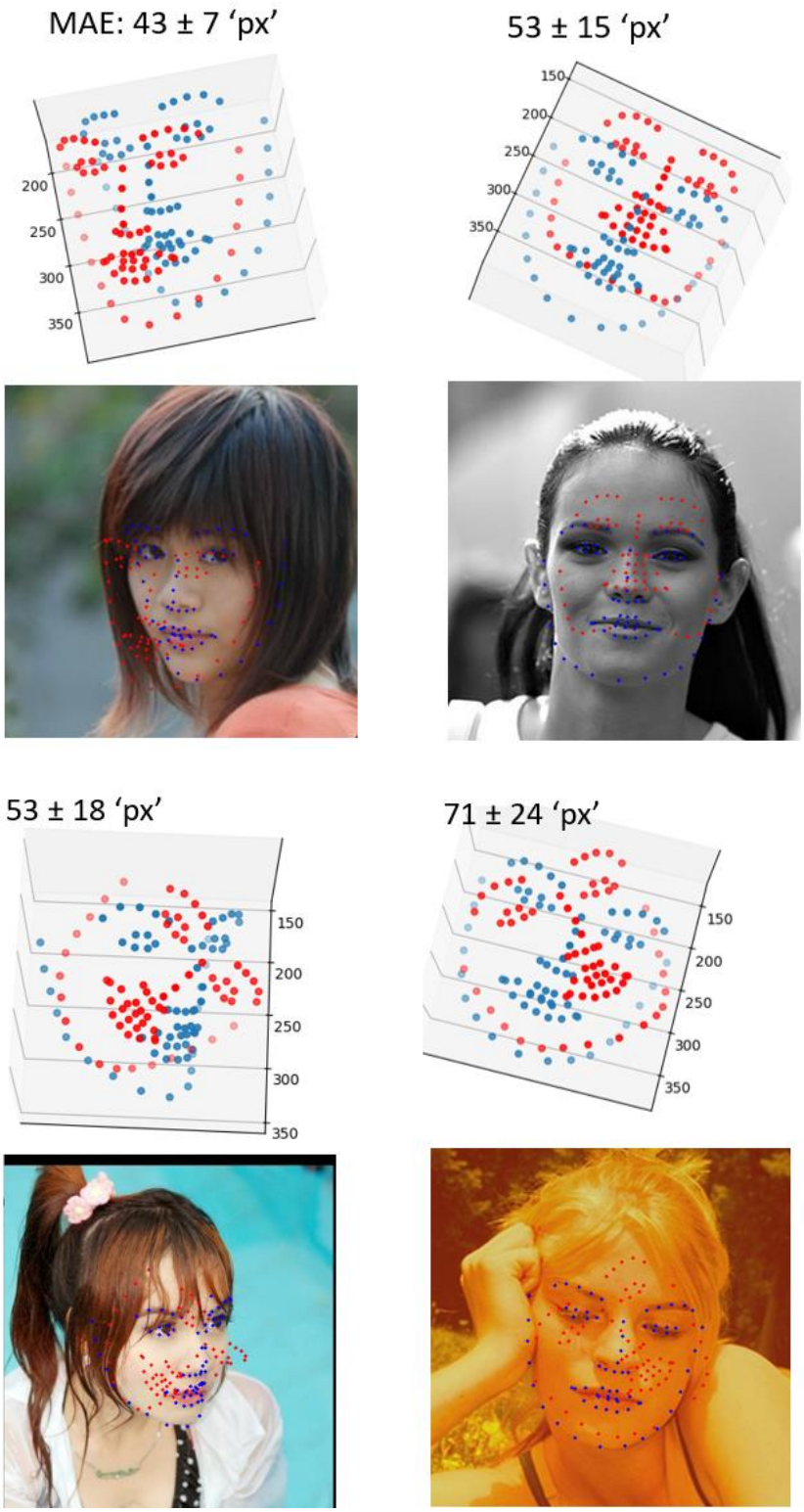

Figure 24 - Exemples de résultats proches de la moyenne obtenus sur la base de test

3.2.3 Comparaison des résultats

Le Tableau 5 reprend les résultats présentés précédemment sur les tests réalisés sur les entrainements du réseau de neurones utilisant la matrice de transformation en gris et le vecteur de pose en blanc afin de les comparer. Premièrement, on remarque que les résultats des deux entrainements suivent la même tendance : le teste de reconstruction donne une erreur plus faible que les deux autres tandis que le test de recalage obtient une erreur très proche de celle du test global. On peut également remarquer que les erreurs présentées ici sont plus importantes que lors de l'entrainement utilisant une matrice de transformation pour représenter la pose au lieu du vecteur de pose utilisant des angles d'Euler. En effet, sur les résultats globaux des deux entrainements celui utilisant la matrice de transformation obtient une MAE de 29.4 alors que le deuxième nous donne une MAE de 59.7, soit le double.

Tableau 5 - Tableau récapitulatif des résultats des entraînements utilisant la matrice de transformation (en gris) et utilisant le vecteur de pose (en blanc)

Test	**MAE**	**Min**	**Max**
Reconstruction	5.7 ± 2.4	1.7 ± 1.1	15.6 ± 6.0
	6.7 ± 2.8	2.0 ± 1.2	19.9 ± 11.7
Estimation de pose	28.9 ± 16.4	6.1 ±3.6	139.6 ± 39.1
	59.4 ± 32.3	15.6 ± 9.1	244.7 ± 47.3
Global	29.4 ± 16.4	7.8 ± 3.6	142.8 ± 38.2
	59.7 ± 32.3	15.6 ± 7.4	241.8 ± 67.5

Pour ce qui est des résultats qualitatifs, l'entrainement réalisé avec la matrice de transformation semble générer des modèles de visages qui suivent mieux le modèle de référence mais on peut remarquer certaines déformations que l'on ne remarque pas sur les résultats sortant de l'entrainement réalisé avec le vecteur de pose.

CHAPITRE 4

DISCUSSION

4.1 Implémentation du modèle PCA

Pour ce qui est des tests de généralisation et de compacité, notre modèle PCA se comporte de la manière prédite par (Heimann & Meinzer, 2009). L'erreur de reconstruction décroît bien vers 0 dans les deux cas. Notre modèle est donc validé pour ces tests. Pour la génération de visage, nous avons pu voir (Figure 15) que notre modèle est capable de créer des visages différents sans pour autant créer des modèles 3D trop éloignés de la base d'entraînement. Cela fait que lors du test de spécificité, notre erreur reste relativement stable.

Pour conclure cette comparaison, notre modèle est compact, généralise facilement, et est suffisamment spécifique pour générer des visages cohérents. Il est donc validé et pourra être utilisé pour la suite.

4.1.1 Réduction de paramètre

Le modèle PCA se comporte, pour ce qui est du taux de variance expliquée, comme on l'attendait (Figure 17). On peut y voir que les 17 premiers vecteurs propres sur les 204 au total sont capable d'expliquer 95% de la variance parmi les modèles 3D de visage et qu'il faut ajouter énormément de vecteur propre pour augmenter cette quantité.

La Figure 18 présente de manière plus explicite l'erreur impliquée par la réduction de paramètres. On a une erreur plus grande lorsque l'on a seulement 2 VP conservés que lorsque l'on en conserve plus. Cela dit, on remarque que plus on conserve de vecteurs propres, plus il faut en rajouter pour obtenir une augmentation significative de la précision du modèle de visage reconstruit par le modèle PCA. Le Tableau 2 nous montre que passer de 17 à 30 vecteurs propres conservés diminue l'erreur moyenne de 1.5 pixel. Cette amélioration semble trop peu significative face à l'augmentation du nombre de VP à conserver. En effet, plus on conserve

de vecteurs propres, plus on augmente le nombre de paramètres statistiques nécessaire pour décrire le visage (voir Figure 2) et donc le nombre de paramètres que notre réseau de neurones devra extraire de l'image du visage par la suite. Cela peut créer une source d'erreur plus grande que l'augmentation de la précision permise par un plus grand nombre de VP. Par conséquent, 17 vecteurs propres conservés par le modèle PCA semble bien être une solution pertinente.

Le modèle PCA est à la hauteur de nos attentes, on est capable de reconstruire relativement précisément des modèles de visages à partir d'uniquement 17 paramètres. Il est cependant dommage que les modèles 3D ne soient composés que de 68 points, beaucoup d'informations sur les régions plus secondaires du visage (joues, pommettes, …) sont perdues ainsi.

4.2 Estimation des paramètres statistiques et de pose

On peut commencer par analyser les courbes obtenues en sortie d'entrainement (Figure 19 et Figure 22). En effet, on peut voir que pour la base d'entraînement, les courbes de perte décroissent et les courbes de précision croissent. Cela signifie que le réseau apprend lors de l'entraînement. Cependant, les valeurs de perte et de précision sur la base de validation ne suivent pas correctement celles de la base d'entraînement ce qui signifie que le réseau n'arrive pas à généraliser les résultats de son apprentissage sur des données inconnues. Ce problème se manifeste dans les résultats des tests. En effet, on remarque sur les Figure 20 et Figure 23 que les meilleurs résultats de nos réseaux sont précis, les points du modèle 3D généré sont proches des points du modèle de référence mais que dans les pires résultats on a pas l'impression que le réseau de neurones ait reconnu un visage dans l'image. Le RéseauA nous renvoie un visage déformé et tourné à 90 degrés par rapport au visage de référence (à droite de la Figure 20) et le RéseauB renvoie un modèle qui n'est même pas placé sur le visage (à droite de la Figure 22).

Ce problème aurait pu être dû à un phénomène de sur-apprentissage où le réseau aurait tellement appris sur ses données d'entrainement qu'il deviendrait incapable de se servir de données différentes mais des précautions ont été prises pour éviter ce problème.

L’entrainement est arrêté dès que le réseau ne s’améliore, le taux d’apprentissage diminue au cour de l’entrainement et une technique de dropout ont été mis en place pour cela.

Une raison plus probable qui causerait ce problème serait directement liée aux données. Il est possible que le nombre de données (2000 visages au total) ne soit pas suffisant pour que le réseau soit capable de généraliser ses résultats. Ou encore, il est possible que le problème soit lié à l’uniformité de la répartition des données dans les bases d’entrainement, validation, et test. En effet, on peut remarquer que certains types de visages sont plus facile pour le modèle que d’autre. Sur la Figure 21 les résultats sur le visage de profil (en haut à droite) ou le visage à l’éclairage fortement contrasté (en bas à droite) sont bien moins précis que les autres. Il est probable que la base d’entrainement contient peu de visage de ces conditions alors que les bases de validation et de test en contiennent plus. En effet, la répartition des données dans ces bases été réalisée aléatoirement, les proportions des différents types de visages dans ces dernières n’ont pas été vérifiée.

Cependant, certains résultats sont tout de même satisfaisant ce qui laisse penser qu’une fois que le réseau de neurones sera capable de généraliser ces résultats, notre système pourra obtenir des résultats intéressants.

On peut tout de même réaliser une comparatison des résultats de nos tests entre les RéseauA et RéseauB. Un premier point que l’on peut remarquer avec les résultats de l’entrainement, c’est que notre système semble meilleur pour l’extraction des paramètres statistiques que pour celle des informations de pose. En effet, on retrouve une MAE bien plus grande sur le test de recalage d’un modèle de référence que sur le test de comparaison avec le modèle de visage généré après avoir subi la même transformation. Cependant, cela peut être dû au fait que les paramètres statistiques sont réutilisés par le modèle PCA avant de créer les points 3D et que notre modèle PCA est plus robuste à l’erreur. Le réseau pourrait être tout aussi précis sur les paramètres statistiques que les informations de pose mais le modèle PCA rattraperait cette erreur sur les paramètres PCA mais pas sur les informations de pose.

Ensuite, le RéseauA semblent avoir de meilleurs résultats. Cela est possiblement dû au phénomène de Gimbal Lock. En effet, les données décrivent un grand nombre de position de la tête dont des rotations très larges autour de l'axe vertical de la tête. Cela peut créer certaines configurations où les deux autres axes de rotations soient confondus. Il est possible que ce phénomène soit mal représenté dans les données, ce qui entrainerait la différence dans l'erreur. On pourra d'ailleurs remarquer sur les exemples que lors des essais utilisant le vecteur de pose, la rotation autour de l'axe vertical semble plus précise que les deux autres rotations.

D'un autre côté, lors les essais utilisant la matrice de transformation, il est possible de remarquer que certains modèles de visages, pourtant relativement bien reconstruits lors des premiers tests, se retrouvent déformés après la transformation. Par exemple, le visage du sujet en haut à droite de la Figure 21 se retrouve écrasé après l'application de la matrice de transformation. Cela est dû à des différences de précision lors des prédictions du réseau sur la partie rotation de la matrice. Dans une matrice de rotation, chaque paramètre est lié aux autres de manière à avoir une matrice orthogonale. Cependant, notre réseau de neurones prédit chacun de ces paramètres indépendamment des autres. Il est donc possible que, lors de l'entrainement, le réseau n'ait pas compris les relations entre ces paramètres. Certains paramètres prédits, peuvent être trop grands ou trop petit et concordent pas avec les autres ce qui va venir se traduire par une déformation du modèle lorsque la matrice de transformation lui est appliqué et non une simple rotation et translation. De plus, on remarquera également mauvaise prédiction de la translation sur quelques exemples de la base de test. Lorsque cela arrive, on obtient une lourde augmentation de la MAE comme tous les points vont voir leur erreur augmenter du fait que le modèle entier est mal positionné.

Pour résumer, on fait face à deux problèmes avec nos réseaux de neurones. Premièrement représenter la pose à l'aide d'une matrice de transformation démontre de meilleurs résultats sur la précision des points du modèle de visage généré mais introduit actuellement des déformations du modèle. Tandis que la représentation à l'aide du vecteur de pose et des angles d'Euler conserve la forme du modèle mais le phénomène de Gimbal Lock rend l'apprentissage de l'orientation plus difficile. Et deuxièmement, les données utilisées lors des entrainements

ne permettent pas aux réseaux de généraliser ses apprentissages ce qui fait que l'on se retrouve avec un type d'image sur lesquelles les résultats obtenus sont relativement précis tandis que d'autres type d'image sont très mal traités.

CONCLUSION

Pour conclure, ce rapport reprend l'essentiel de ce qui a été réalisé au cours de ce projet de reconstruction de modèles 3D de visage de patients atteints de cancer au niveau du visage, réalisé avec le Centre de Recherche du Centre Hospitalier Universitaire de Montréal dans le cadre ma maîtrise en technologie de la santé à l'ÉTS.

Y a été présenté l'état de l'art actuel démontrant les points importants de ce domaine tels que la détection de visage, la reconstruction faciale et le recalage d'objets numérique sur une image qui nous a permis d'établir une méthodologie qui a servi à implémenter notre solution retenue par la suite, un système capable d'extraire les informations de pose et le paramètres statistiques d'un visage présent dans une image pour ensuite venir utiliser ces données afin de reconstruire un modèle 3D correspondant à ce visage.

Une fois implémentée, nos tests nous ont montrés que notre système actuel est précis sur la reconstruction du visage trouvé dans l'image mais moins sur son positionnement soit dû à un phénomène de déformation du modèle 3D soit à une confusion dans la prédiction des angles. Cette imprécision pourra cependant être améliorée en augmentant le nombre de données utilisées et en s'assurant de leur uniformité au travers des bases d'entrainement, validation et de test afin de permettre au réseau une meilleure généralisation.

Ce projet ne s'arrête pas ici. En effet, Antoine continue de travailler dessus dans le but d'implémenter des méthodes plus adaptées à notre cas. Par exemple l'utilisation de méthodes dynamique profitant du fait que l'on utilise une vidéo et non une image unique afin d'améliorer le modèle de visage tout au long de l'utilisation du système.

Il sera également intéressant de recueillir des données de patients comme des scans3D de visage afin de pouvoir d'entraîner nos modèles sur des données plus spécifiques. En effet, la base de données utilisée lors de ce projet ne contient que des visages communs. Pour rendre notre système réellement efficace sur des patients, il serait préférable qu'il se soit familiarisé

avec le type de visage qu'il pourra rencontrer. En plus de cela, recueillir de nouvelles données nous permettrait d'avoir des modèles de visages composés de plus de points et donc de créer un modèle de visage plus réaliste.

Enfin, il serait également possible d'améliorer notre système actuel. On a pu remarquer que la détection de la pose du visage pouvait être améliorée. Peut-être est-ce un problème de représentation de cette dernière. Utiliser d'autres représentations telles que les quaternions pour exprimer les angles n'a pas été vu dans la littérature mais pourrait apporter de meilleurs résultats. Le problème de déformation de la matrice de transformation et celui de Gimbal Lock pour les angles d'Euler seraient ainsi tous deux évités.

BIBLIOGRAPHIE

Albahli, S. (2022). Figure 4: Architecture of ResNet50. *ResearchGate*. Repéré à https://www.researchgate.net/figure/Architecture-of-ResNet50_fig3_357668796

Arif, A. (2021, 7 janvier). How Principal Component Analysis, PCA Works - Dataaspirant. Repéré à https://dataaspirant.com/principal-component-analysis-pca/

Bouaziz, S., Wang, Y., & Pauly, M. (2013). Online modeling for realtime facial animation. *ACM Transactions on Graphics*, *32*(4), 40:1-40:10. https://doi.org/10.1145/2461912.2461976

Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M., Correia, L., & J. Tallón-Ballesteros, A. (2023). The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis. Dans P. García Bringas, H. Pérez García, F. J. Martínez de Pisón, F. Martínez Álvarez, A. Troncoso Lora, Á. Herrero, … E. Corchado (Éds), *18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023)* (pp. 344-353). Cham : Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42536-3_33

de Lucena, J. O., Lima, J. P., Thomas, D., & Teichrieb, V. (2019). Real-Time Facial Motion Capture Using RGB-D Images Under Complex Motion and Occlusions. Dans *2019 21st Symposium on Virtual and Augmented Reality (SVR)* (pp. 120-129). https://doi.org/10.1109/SVR.2019.00034

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus, *24*(6).

Grishchenko, I., Ablavatski, A., Kartynnik, Y., Raveendran, K., & Grundmann, M. (2020, 19 juin). Attention Mesh: High-fidelity Face Mesh Prediction in Real-time. arXiv. Repéré à http://arxiv.org/abs/2006.10962

Guo, J., Zhu, X., Yang, Y., Lei, Z., Fan, Y., & Li, S. Z. (2020). Towards Fast, Accurate and Stable 3D Dense Face Alignment. Repéré à https://doi.org/10.48550/arXiv.2009.09960

Guo, Y., Zhang, J., Cai, J., Jiang, B., & Zheng, J. (2018, 15 mai). CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images. arXiv. Repéré à http://arxiv.org/abs/1708.00980

Gupta, A., Thakkar, K., Gandhi, V., & Narayanan, P. J. (2018, 3 décembre). Nose, eyes and ears: Head pose estimation by locating facial keypoints. arXiv. https://doi.org/10.48550/arXiv.1812.00739

Hashemibakhtiar, P., Cresson, T., Nault, M.-L., de Guise, J., & Vázquez, C. (2024). 2D/3D RECONSTRUCTION OF THE DISTAL TIBIOFIBULAR JOINT FROM BIPLANAR RADIOGRAPHS USING DEEP LEARNING REGISTRATION AND STATISTICAL SHAPE AND INTENSITY MODEL.

He, K., Zhang, X., Ren, S., & Sun, J. (2015, 10 décembre). Deep Residual Learning for Image Recognition. arXiv. https://doi.org/10.48550/arXiv.1512.03385

Heimann, T., & Meinzer, H.-P. (2009). Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, *13*(4), 543-563. https://doi.org/10.1016/j.media.2009.05.004

Hemingway, E., & O'Reilly, O. (2018). Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics*, *44*. https://doi.org/10.1007/s11044-018-9620-0

Jeni, L. A., Tulyakov, S., Yin, L., Sebe, N., & Cohn, J. F. (2016). The First 3D Face Alignment in the Wild (3DFAW) Challenge. Dans G. Hua & H. Jégou (Éds), *Computer Vision – ECCV 2016 Workshops* (pp. 511-520). Cham : Springer International Publishing. https://doi.org/10.1007/978-3-319-48881-3_35

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, *81*(2), 155-166. https://doi.org/10.1007/s11263-008-0152-6

Minaee, S., Luo, P., Lin, Z., & Bowyer, K. (2021, 13 avril). Going Deeper Into Face Detection: A Survey. arXiv. https://doi.org/10.48550/arXiv.2103.14983

Nguyen, D.-P., Nguyen, T.-N., Dakpé, S., Ho Ba Tho, M.-C., & Dao, T.-T. (2022). Fast 3D Face Reconstruction from a Single Image Using Different Deep Learning Approaches

for Facial Palsy Patients. *Bioengineering*, *9*(11), 619. https://doi.org/10.3390/bioengineering9110619

OpenCV. (s.d.). OpenCV: Perspective-n-Point (PnP) pose computation. Repéré à https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

O'Shea, K., & Nash, R. (2015, 2 décembre). An Introduction to Convolutional Neural Networks. arXiv. https://doi.org/10.48550/arXiv.1511.08458

Paysan, P., Knothe, R., Amberg, B., Romdhani, S., & Vetter, T. (2009). A 3D Face Model for Pose and Illumination Invariant Face Recognition. Dans *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance* (pp. 296-301). https://doi.org/10.1109/AVSS.2009.58

Pillai, R. K., Jeni, L. A., Yang, H., Zhang, Z., Yin, L., & Cohn, J. F. (2019). The 2nd 3D Face Alignment in the Wild Challenge (3DFAW-Video): Dense Reconstruction From Video. Dans *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (pp. 3082-3089). Seoul, Korea (South) : IEEE. https://doi.org/10.1109/ICCVW.2019.00371

Prados-Torreblanca, A., Buenaposada, J. M., & Baumela, L. (2022, 13 octobre). Shape Preserving Facial Landmarks with Graph Attention Networks. arXiv. Repéré à http://arxiv.org/abs/2210.07233

Salem, N., & Hussein, S. (2019). Data dimensional reduction and principal components analysis. *Procedia Computer Science*, *163*, 292-299. https://doi.org/10.1016/j.procs.2019.12.111

Tensorflow. (s.d.). tf.keras.callbacks.LearningRateScheduler | TensorFlow v2.16.1. *TensorFlow*. [Documentation]. Repéré à https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/LearningRateScheduler

Terzakis, G., & Lourakis, M. (2020). A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. Dans A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Éds), *Computer Vision – ECCV 2020* (pp. 478-494). Cham : Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_28

Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., & Zhou, Q. (2018, 26 mai). Look at Boundary: A Boundary-Aware Face Alignment Algorithm. arXiv. https://doi.org/10.48550/arXiv.1805.10483

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, *23*(10), 1499-1503. https://doi.org/10.1109/LSP.2016.2603342

Zhang, Z. (1993). Le Probleme de la mise en correspondance : l'etat de l'art.

Zhu, X., Lei, Z., Liu, X., Shi, H., & Li, S. Z. (2019). Face Alignment Across Large Poses: A 3D Solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(1), 78-92. https://doi.org/10.1109/TPAMI.2017.2778152

Zielonka, W., Bolkart, T., & Thies, J. (2022, 19 octobre). Towards Metrical Reconstruction of Human Faces. arXiv. https://doi.org/10.48550/arXiv.2204.06607