**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR**
**ET DE LA RECHERCHE SCIENTIFIQUE**
**UNIVERSITE ABBES LAGHROUR DE KHENCHELA**
**FACULTE DES SCIENCES ET DE LA TECHNOLOGIE**

**Département Mathématique et Informatique**

**N° de série : ……**

# Mémoire de fin d'études

*Pour l'obtention du diplôme de Master (L.M.D)*

## Spécialité : Sécurité et Technologie Web

# *Apprentissage Profond pour la reconstruction d'un visage 3D à partir d'une image 2D.*

*Présenté et soutenu publiquement par :*
*Mehdi Malah & Ramzi Agaba*

Le : 14/09/2020

*Membres de jury :*

*Président du jury : Mohamed Mahdi Malik*

*Examinateur : Mohamed Boussalem*

*Encadreur : Abbas Fayçal.*

*Année universitaire : 2019/2020.*

## Résumé

Ces dernières années, la reconstruction 3d du visage a marqué sa présence dans plusieurs domaines telle que la sécurité biométrique, vision par ordinateur, synthèse d'image. De nombreuses applications de reconnaissance faciale nécessitent une reconstruction 3D précise, cependant cette tâche est complexe et nécessite beaucoup de calcules, Ce mémoire présente une nouvelle méthode basée sur l'apprentissage profond afin d'offrir une solution automatique pour la reconstruction 3D de visage à partir d'une seule image 2D, notre méthode opère en trois étape : la première consiste à entrainer un réseau de neurone convolutifs sur notre base de donnée afin de prédire la détection des points de repères du visage et estimer leurs positions avec précision à partir d'une seule image de visage dans le repère de texture (l'espace image). La seconde étape consiste à produire une forme géométrique (mesh) du visage, La troisième étape consiste à effectuer une transformation (translation) entre l'espace 3D de l'objet et l'espace 2D de l'image afin de déterminer les cordonnées de texture qui correspondent à chaque polygone de visage.

Notre méthode produit de bons résultats en termes de précision et de qualité visuel, et cela en prenant en compte tous les paramètres du modèle (forme, expression, réflectance et éclairage) en entrée, notre méthode simple à mette en œuvre et offre une reconstruction 3D du visage en temps réel.

**Mots clés** :

Apprentissage automatique, Apprentissage profond, Reconstruction 3D du visage, Réseau de neurones convolutifs.

**ملخص:**

في السنوات الأخيرة، تركت إعادة بناء وجه ثلاثي الأبعاد بصمتها في العديد من المجالات مثل الأمن البيومتري، ورؤية الكمبيوتر، وتركيب الصور. تتطلب العديد من تطبيقات التعرف على الوجه إعادة بناء دقيقة ثلاثية الأبعاد، ولكن هذه المهمة معقدة وتتطلب الكثير من الحسابات، تقدم هذه الأطروحة طريقة جديدة تعتمد على التعلم العميق من أجل تقديم حل ألي لإعادة بناء الوجه ثلاثي الأبعاد انطلاقا من صورة واحدة ثنائية الأبعاد، تعمل طريقتنا في ثلاث خطوات: الخطوة الأولى تتكون من تدريب شبكة الخلايا العصبية الملتوية الاصطناعية على قاعدة البيانات الخاصة بنا من أجل التنبؤ باكتشاف معالم الوجه وتقدير موقعها بدقة انطلاقا من صورة واحدة للوجه ، ممثلة في معلم الصورة، ثم في الخطوة الثانية التي تتمثل في إنتاج شكل هندسي للوجه. الخطوة الثالثة تتمثل في التحويل النقطي بين الفضاء ثلاثي الأبعاد للكائن والفضاء ثنائي الأبعاد للصورة لتحديد إحداثيات النقط التي تتوافق مع كل مضلع وجه.

نتج عن طريقتنا نتائج جيدة من حيث الدقة والجودة المرئية، وذلك من خلال مراعاة جميع معالم النموذج (الشكل والتعبير والانعكاس والإضاءة) كمدخلات، وطريقتنا سهلة الاستعمال.. وتوفر إعادة بناء ثلاثية الأبعاد للوجه في الوقت الحقيقي.

**الكلمات المفتاحية:**

تعلم الآلة، التعلم العميق، إعادة بناء الوجه ثلاثي الأبعاد، شبكة من الخلايا العصبية الملتوية.

## Abstract

In recent years, 3D facial reconstruction marked its presence in several areas such as biometric security, computer vision, image synthesis. Many facial recognition applications require accurate 3D reconstruction, however this task is complex and requires a lot of calculations, this thesis presents a new method based on deep learning in order to offer an automatic solution for 3D reconstruction of the face depending on a single 2D image, our method operates in three steps : the first step is to train a convolutional neural network with our dataset in order to predict and detect the landmarks of the face and estimating its positions accurately from a single facial image in the image space then in a second step that consists of producing a geometric shape (mesh) of the face, finally the third step is to make a translation between the 3D space of the object and the 2D image space in order to determine the texture referrals that correspond to each face polygon.

Our method produces good results in terms of accuracy and visual quality, taking into account all the parameters of the model (shape, expression, reflectance and lighting) as inputs, our simple method is easy to implement and offers a 3D reconstruction of the face in real time.

**Keywords**:

Machine learning, Deep learning, 3D facial reconstruction, Convolutional neural network.

# *Dédicaces*

*La dédicace primordiale va à ceux qui m'ont amené à l'existence et m'a fait la personne que je suis aujourd'hui, à l'affection, l'amour et le soutien que ma mère « **Assia** » incarne, et à la personne que je respecte et chéris, mon père et mon modèle « **Sebti** »*

*A mes piliers de force et de ténacité, mes frères « **Mohamed** » et « **Ayoub El Mahdi** »*

*À ceux qui sont toujours préoccupés par ma santé et mon bien-être, à ma tendre grand-mère « **Fatma** » et ma belle tante « **Nadjiba** »*

*Je suis également redevable de dédier cet effort aux compagnons de voyage, le généreux à qui on peut compter sur « **Mehdi Malah** », le sage et raisonnable « **Aymen Chekhab** », le jeune joyeux et talentueux « **Islem Djebabra** », l'amie d'enfance « **Ines Bouziane** », Enfin, je souhaite exprimer ma gratitude à toute l'équipe et aux camarades de combat : Alla Eddine, Badr Eddine, Hamza, Mohamed, Oussama, Soulaimen et aux nombreux autres qui méritent d'être félicités et honorés.*

*« La valeur d'un homme tient dans sa capacité à donner et non dans sa capacité à recevoir. » Albert Einstein*

***Ramzi Agaba***

# *Dédicaces*

Mes Parents, mon frère, ma sœur,

Mes Amis et Mes collègues

**Malah Mehdi**

# *Remerciements*

*En premier, nous aimerions remercier le bon Dieu le tout puissant de nous avoir donné le courage et la volonté de réaliser ce mémoire.*

*Nous tenons tout d'abord à manifester notre profonde gratitude envers notre encadreur de ce mémoire Dr. Fayçal ABBAS pour nous avoir fait profiter de son expérience dans la recherche et de la qualité de son encadrement pendant toute la durée du mémoire, sa disponibilité et ses conseils ont été très constructifs.*

*Nous désirons remercier nos chers parents qui nous ont soutenus et encouragé durant toute notre vie et pendant notre cursus d'étude.*

*Nos remerciements les plus chaleureux vont à tous nos amis pour leurs disponibilités et leurs très précieux conseils ainsi que leurs remarques qui nous ont permis d'améliorer la qualité de ce travail.*

*Nous tenons à exprimer toute notre grande gratitude aux membres de jury d'avoir accepté de juger ce travail.*

# Table de Matières

**Introduction générale** ..... **1**
**Chapitre I :** ..... **3**
**1. Introduction** ..... **4**
**2. L'intelligence artificielle** ..... **4**
**2.1.** Définition D'IA ..... **4**
**2.2.** Une courte histoire de l'Intelligence artificielle ..... **5**
**2.3.** Les Techniques de l'intelligence artificielle ..... **5**
**2.3.1.** Système Expert **5**
**2.3.2.** Système Multi-Agent **6**
**2.3.3.** Réseau de neurones **7**
**2.4.** Les Application de l'intelligence artificielle ..... **7**
**2.4.1.** Vision **7**
**2.4.2.** Langage **8**
**2.4.3.** Robots **9**
**3. Apprentissage Automatique** ..... **9**
**3.1.** Diverses définitions ..... **10**
**3.2.** Les Différents Types d'apprentissage automatiques ..... **10**
**.3.2.1** Apprentissage supervisé **10**
**3.2.2.** Apprentissage non supervise **11**
**3.2.3.** Apprentissage semi-supervise **12**
**4. Apprentissage profond** ..... **12**
**4.1.** Qu'est-ce que l'apprentissage profond ..... **13**
**4.2.** Les algorithmes de l'apprentissage profond ..... **13**
**4.3.** Les réseaux de neurones convolutifs (CNN) ..... **13**
**4.3.1.** La couche convolution **14**
**4.3.2.** La couche max-pooling **15**
**5. Conclusion** ..... **15**
**Chapitre II :** ..... **16**
**1. Introduction** ..... **17**
**2. Techniques pour la détection des visages** ..... **17**
**2.1.** Approches basées sur les connaissances acquises ..... **17**
**2.2.** Approches basées sur le « Template-matching » ..... **17**
**2.3.** Approches basées sur l'apparence ..... **19**

**2.4.** Approches basées sur des caractéristiques invariantes ..... **20**
**3. Techniques pour la reconnaissance des visages** ..... **22**
**3.1.** Principales difficultés de la reconnaissance de visage ..... **22**
**3.2.** Systèmes d'acquisition 3D ..... **22**
**3.3.** Approches modèle ..... **23**
**4. Techniques de Reconstruction 3D du visage** ..... **24**
**4.1.** Les Méthodes Monoculaires ..... **24**
**4.1.1.** Shape From Shading **25**
**4.1.2.** Shape From Texture **25**
**4.1.3.** Shape from Focus/Defocus **27**
**4.1.4** Caméras 2,5D à temps de vol **27**
**4.2.** Les Méthodes multi-vues ..... **28**
**4.2.1.** Stéréovision **28**
**4.2.2.** Lumière Structurée **29**
**4.2.3.** Shape-From-Silhouette **30**
**5. Discussion et Présentation des Approches** ..... **31**
**5.1.** Accurate 3D Face Reconstruction with Weakly-Supervised Learning : From Single Image to Image Set ..... **31**
**5.2.** Face Alignment in Full Pose Range : A 3D Total Solution ..... **32**
**5.3.** Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network ..... **33**
**5.4.** Learning Detailed Face Reconstruction from a Single Image ..... **34**
**5.5.** Self-Supervised Monocular 3D Face Reconstruction by Occlusion-Aware Multi-view Geometry Consistency ..... **35**
**5.6.** Learning Robust 3D Face Reconstruction and Discriminative Identity Representation ..... **36**
**6. Conclusion** ..... **37**
**Chapitre III :** ..... **38**
**1. Introduction** ..... **39**
**2. Description des Activités de Notre Modèle** ..... **39**
**2.1.** La Création du Jeu de Données ..... **39**
**2.1.1.** La Collection des données **39**
**2.1.2.** Détection Du Visage 2D **40**
**2.1.3.** Profondeur des Points **41**
**2.1.4.** La Sélection des Objets 3D **41**
**2.2.** Architecture de CNN ..... **42**

**2.2.1.** Prétraitement du jeu de données **42**
**2.2.2.** Apprentissage **42**
**2.3.** La Reconstruction 3D ..... **45**
**2.3.1.** La Création de la géométrie 3D **45**
**2.3.2.** Le Plaquage de texture **46**
**3. Conclusion** ..... **46**
**Chapitre IV :** ..... **47**
**1. Introduction** ..... **48**
**2. La Configuration du Matériel Utilisé** ..... **48**
**3. L'environnement de Travail** ..... **49**
**3.1.** Python ..... **49**
**3.2.** JavaScript ..... **49**
**3.3.** NodeJS ..... **49**
**3.4.** Tensorflow ..... **49**
**3.5.** Keras ..... **49**
**3.6.** THREE JS ..... **49**
**4. Résultats** ..... **50**
**4.1.** Le Jeu de Données ..... **50**
**4.2.** La Phase de L'apprentissage ..... **52**
**4.3.** Résultat de reconstruction 3D ..... **53**
**5. Validation** ..... **55**
**5.1.** La Comparaison Quantitative ..... **55**
**5.2.** La Comparaison Qualitative ..... **56**
**6. Conclusion** ..... **57**
**Conclusion générale et perspectives** ..... **59**
**Bibliographie** ..... **60**

# Liste des Figures

**Figure I.1** Structure d'un système Expert ..... **6**
**Figure I.2** Interaction de l'agent avec l'environnement ..... **6**
**Figure I.3** Structure d'un Réseau de neurones. ..... **7**
**Figure I.4** Exemple de détection d'objets ..... **8**
**Figure I.5** Exemples des applications de langage. ..... **8**
**Figure I.6** Capteurs et effecteurs d'un robot. ..... **9**
**Figure I.7** Exemple d'un jeu de données pour la reconnaissance manuscrite. ..... **10**
**Figure I.8** Exemple de la couche convolution. ..... **14**
**Figure I.9** Exemple de la couche max-pooling. ..... **15**
**Figure II.1** Différentes régions utilisées pour la phase de template matching. [16] ..... **18**
**Figure II.2** Modèle de visage composé de 16 régions (les rectangles) associées à 23 relations (flèches). [17] ..... **18**
**Figure II.3** Eigenfaces calculés à partir de l'image d'entrée. [18] ..... **19**
**Figure II.4** Une image de visage original et ça projection sur l'espace des eigenfaces défini dans la figure II.3. [18] ..... **19**
**Figure II.5** Sortie de Haar-Cascade sur un certain nombre d'images de test obtenus à partir les jeux de données MIT+CMU. [20] ..... **20**
**Figure II.6** Exemples d'entrées et leurs résultats avec une méthode explicite. [21] ..... **21**
**Figure II.7** A gauche l'image texture, transformée en 2.5D (au centre) et l'image 3D. ..... **23**
**Figure II.8** Résultat de l'approche Shape From Shading proposée par Meyer et al. dans [30]. A gauche l'image source, transformée en niveaux de gris (au centre) et la reconstruction correspondante par Shape From Shading. ..... **25**
**Figure II.9** Reconstruction de forme à partir de l'approche Shape From Texture. L'image de gauche représente l'image source, la géométrie estimée est représentée au centre, enfin la texture générée est présentée sur l'image de droite ..... **26**
**Figure II.10** Reconstruction de forme à partir de l'approche Shape From Texture proposée par Verbin et Zickler dans [31] ..... **26**
**Figure II.11** Estimation de carte de profondeur utilisant l'approche Shape From Defocus proposée par Jin et Favaro dans [32]. ..... **27**
**Figure II.12** Exemple de caméra temps de vol ..... **28**
**Figure II.13** Estimation de la carte de profondeur de la scène. L'image de droite a été calculée en utilisant l'approche de Klaus et al. [33] ..... **29**

**Figure II.14** Exemple de scène de la lumière structurée. ..... **29**
**Figure II.15** Résultat de l'approche Lumière structurée proposée par Dipanda et Woo dans [34]. A gauche l'image source en gray scale, les points acquises (au centre) et la représentation de résultat de la reconstruction 3D ..... **30**
**Figure II.16** Exemple d'une reconstruction en utilisant la méthode Shape From Silhouette. **30**
**Figure II.17** Résultat proposée par Goldluecke et Magnor dans [35] (a) des images source obtenu à partir des caméras (b) Coque visuelle raffinée, représentant la surface initiale. (c) Résultat final après l'exécution de l'algorithme complet..... **31**
**Figure II.18** Présentation de l'approche utilisée dans [36]. ..... **32**
**Figure II.19** Présentation du réseau utilisée dans [37]. ..... **32**
**Figure II.20** Présentation de la carte de position UV. Gauche : 3D représentation de l'image d'entrée et son nuage de points 3D aligné correspondant (comme vérité au sol). ..... **33**
**Figure II.21** Le réseau de bout en bout, composé de GrosseNet, FineNet et de la couche intermédiaire..... **34**
**Figure II.22** Le flux de formation de l'architecture MGCNet. ..... **35**
**Figure II.23** L'architecture du réseau utilisé. ..... **36**
**Figure III.1** Exemple des étapes pour la détection et la sélection du jeu de de données. ..... **40**
**Figure III.2** Résultat de la détection. ..... **40**
**Figure III.3** Exemple de la phase de sélection des géométries 3D ..... **41**
**Figure III.4** Les cordonnées 'landmarks'..... **41**
**Figure III.5** code source du CNN utilisé. ..... **43**
**Figure III.6** L'architecture CNN utilisée. ..... **43**
**Figure III.7** Exemple de la reconstruction 3D à partir d'une image 2D (sans Texture). ..... **46**
**Figure III.8** Exemple de la reconstruction 3D à partir d'une image 2D (Avec Texture)..... **46**
**Figure IV.1** L'instance utilisé du LambdaLabs GPU cloud. ..... **48**
**Figure IV.2** échantillon d'image du Kaggle. ..... **50**
**Figure IV.3** Exemples d'image du ThisPersondoesnotexist. ..... **50**
**Figure IV.4** échantillon du jeu des images du HELEN. ..... **51**
**Figure IV.5** échantillon du jeu de données généré. ..... **51**
**Figure IV.6** (a), (b) l'étape de l'apprentissage de notre CNN. ..... **52**
**Figure IV.7** précision et architecture du CNN ..... **53**
**Figure IV.8** Comparaison qualitative des formes géométriques générées par notre méthode et [36][37]..... **56**

**Figure IV.9** Comparaison qualitative des formes géométriques avec textures générées par notre méthode et [36][37]. ..... **56**

# Liste des Tableaux

**Tableau. III.1** Description de CNN utilisé**……..……46**

**Tableau. IV.1** Exemples de notre Modèle **...…..……..……55**

**Tableau. IV.1** Comparaison quantitative**…...……..……..……..56**

# Introduction générale

La reconstruction tridimensionnelle des visages repoussa plus loin les limites de la vision par ordinateur, en particulier pour plusieurs domaines tels que la réalité augmentée, la sécurité informatique et l'informatique graphique. Les améliorations matérielles apportées aux GPU suggèrent une reconstruction 3d automatique des visages en temps réel.

Les techniques traditionnelles utilisées pour l'extraction des caractéristiques de visage sont moins stable en terme de localisation des objets dans une image, La complexité imposée par la reconstruction tridimensionnelle des visages réside dans la variation d'illumination et de position ainsi les problèmes d'occultations et d'expression, récentes approches basées sur les modèles d'apprentissage profond sont apparus, en particulier les réseaux de neurones convolutifs qui ont marqué leur présence dans plusieurs domaines de la vision par ordinateur.

Dans notre travail de recherche nous présentons une méthode capable de reconstruire un visage depuis une image 2D. Notre méthode opère en trois parties la première consiste à entrainer un réseau de neurones convolutifs sur notre propre base de données afin de produire une approximation des points de repère (Landmark) dans l'espace image ensuite, une mesh de visage est reconstruite. Finalement nous effectuons une translation entre l'espace 3D de l'objet et l'espace 2D image (texture) afin de déterminer les cordonnées des textures qui correspondent à chaque polygone de visage.

Notre mémoire est organisé comme suite :

- Dans le premier chapitre, nous présenterons le principe de l'apprentissage en IA, pour cela nous avons défini la base associées à l'apprentissage, ainsi que les caractéristiques d'apprentissage et aussi les différentes classifications d'apprentissage automatique, pour les approches d'apprentissage en IA, il existe plusieurs types, nous montrons quelques approches, qui sont les plus connues dans le domaine de l'IA, et les réseaux de neurones convolutionnels qui seront utilisés tout au long de nos travaux.
- Dans le deuxième chapitre, nous présenterons les différentes techniques pour la détection faciale, il en existe plusieurs, nous montrons quelques approches qui sont les plus connus, et nous présenterons les techniques de la reconnaissance faciale. Enfin, nous mettront l'accent sur les différentes façons pour construire des modèles 3D.
- Dans le troisième chapitre, nous allons montrer la partie implémentation de notre travail.

- Dans le quatrième chapitre, nous discuterons les différents résultats obtenus et allons effectuer une comparaison avec d'autres techniques récentes (validation).

Enfin, nous terminerons ce travail par une conclusion générale et nous proposerons quelques perspectives pour des futurs travaux.

# Chapitre I :
# L’intelligence Artificielle

# 1. Introduction

L'intelligence artificielle (IA) est l'un des domaines qui est toujours en plein essor, cet essor d'IA a créé une explosion dans les stratégies potentielles d'utilisation des données pour faire des prédictions scientifiques, L'IA comprend actuellement une grande variété de sous-domaines, tels que la reconnaissance faciale humaine, la reconnaissance tridimensionnelle du visage, la reconnaissance vocale, le diagnostic de maladies ...

Malgré ses contributions substantielles à la recherche scientifique, l'Intelligence Artificielle est concentrée sur les approches mathématiques, visant à résoudre des problèmes formels avec un ensemble d'objectifs bien définis [1].

L'apprentissage automatique (dit aussi ML Machine Learning en anglais) est l'une des extensions de l'intelligence artificielle, ML est un concept qui fait référence à la capacité d'un système d'acquérir et d'intégrer des connaissances d'une façon indépendante. Il englobe toute méthode permettant de construire un modèle de réalité à partir des données prédéfinies, soit en créant un nouveau modèle complètement, soit en améliorant un modèle partiel ou moins général. [2].

L'apprentissage profond (dit aussi DL Deep Learning en anglais) est dérivé de l'apprentissage automatique, il permet de construire des modèles qui ont montré des performances supérieures pour un large éventail d'applications, en particulier sur la vision par ordinateur et le traitement du langage naturel.

Dans ce chapitre nous allons présenter le principe de l'apprentissage en IA, en effet nous allons présenter quelques définitions de bases associées aux approches d'apprentissage en IA, nous présentons quelques approches d'apprentissages qui sont les plus connues dans le domaine de l'IA, et nous allons présenter les réseaux de neurones convolutif.

# 2. L'intelligence artificielle

## 2.1. Définition D'IA

L'IA est un ensemble de techniques qui sont utilisées pour résoudre des problèmes complexes, IA a été inspiré par les mécanismes cognitifs humains, agissant rationnellement sur la base de faits, de données et d'expériences, et capable d'atteindre de manière optimale un ou plusieurs objectifs donnés. [3]

On peut identifier plusieurs manières d'expliquer l'IA, nous allons en développer trois :

L'intelligence artificielle est un domaine de l'informatique qui combine un certain nombre de modèles technologiques qui sont complètement différents les uns des autres, y compris les algorithmes. [4]

L'intelligence artificielle est une interface adaptée à l'homme qui permet une communication facile avec la machine, une technologie qui élimine la technologie [4].

L'intelligence artificielle est l'une des parties les plus importantes de la science cognitive. Les offres liées à Aisi, Microsoft et IBM incluent l'intelligence artificielle "Cognitive Services », ou « Cognitive Computing », avec des niveaux de systèmes d'apprentissage qui visent à enseigner aux ordinateurs à voir, entendre, lire, la mémoire, la structure, l'esprit et la primauté du droit. [4]

### 2.2. Une courte histoire de l'Intelligence artificielle

L'intelligence artificielle est un domaine qui a une longue histoire et qui est toujours en constante évolution, voici quelques étapes importantes dans son histoire :

- 1956-1973 Epoque des pionniers de l'IA symbolique. Période de grand optimisme : « Machines will be capable, within twenty years, of doing any work a man can do »
- (H. Simon, 1965) « Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved » (M. Minsky, 1967).
- 1974→1980 « Hiver de l'IA »
- 1980→1989 Systèmes experts. En 1985, marché d'un milliard de dollars. Projets ambitieux (ordinateur de 5e génération au Japon).
- 1985→1995 Réseaux de neurones.
- 2000→2010 Explosion des applications (augmentation de la puissance des ordinateurs, disponibilité de grandes masses de données et progrès réalisés en apprentissage automatique).

### 2.3. Les Techniques de l'intelligence artificielle

L'intelligence artificielle est un ensemble de techniques visant à tenter d'approcher le raisonnement humain en va expliquer quelque technique importante :

#### 2.3.1. Système Expert

Système Expert est l'expression d'un savoir sous forme de connaissances heuristiques, résultant généralement d'une expérience cumulée dans un domaine précis. Ces connaissances traduisent un mode de raisonnement et peuvent s'exprimer par des règles telles que :«SI telle condition

est vérifiée ALORS effectuer telle action » (ces règles sont appelées plus précisément « règles de production »). [5]

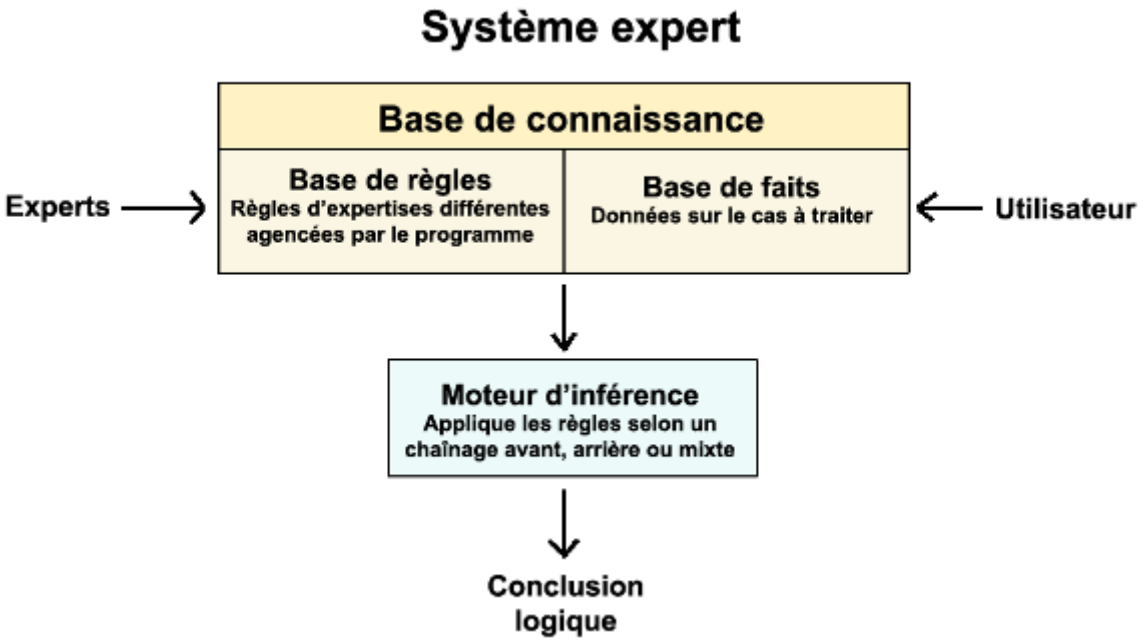



*Figure I.1 Structure d'un système Expert*

### 2.3.2. Système Multi-Agent

Un système multi-agents (ADM) est un système composé de plusieurs agents qui interagissent les uns avec les autres dans un environnement commun. Certains de ces agents peuvent être des personnes ou leurs représentants (avatars), ou même des machines mécaniques. S'il y a moins de trois agents, nous parlons davantage d'interaction homme/machine, ou de machine/machine que de systèmes multi-agents. La Figure I.2 donne une représentation d'un SMA. [6]

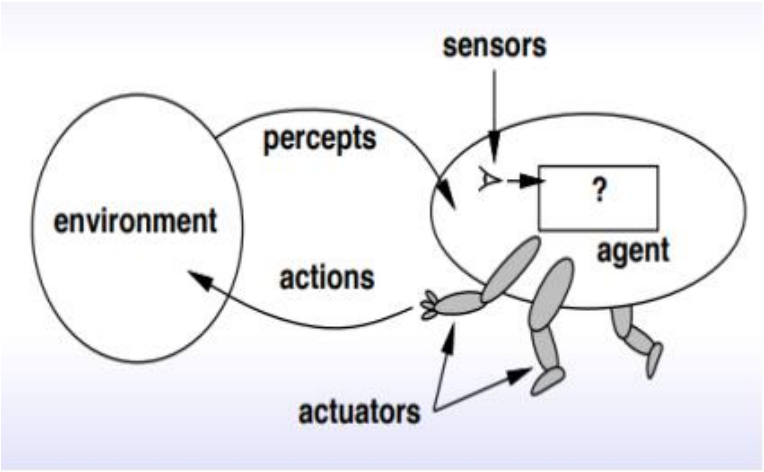



*Figure I.2 Interaction de l'agent avec l'environnement*

### 2.3.3. Réseau de neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. [7]

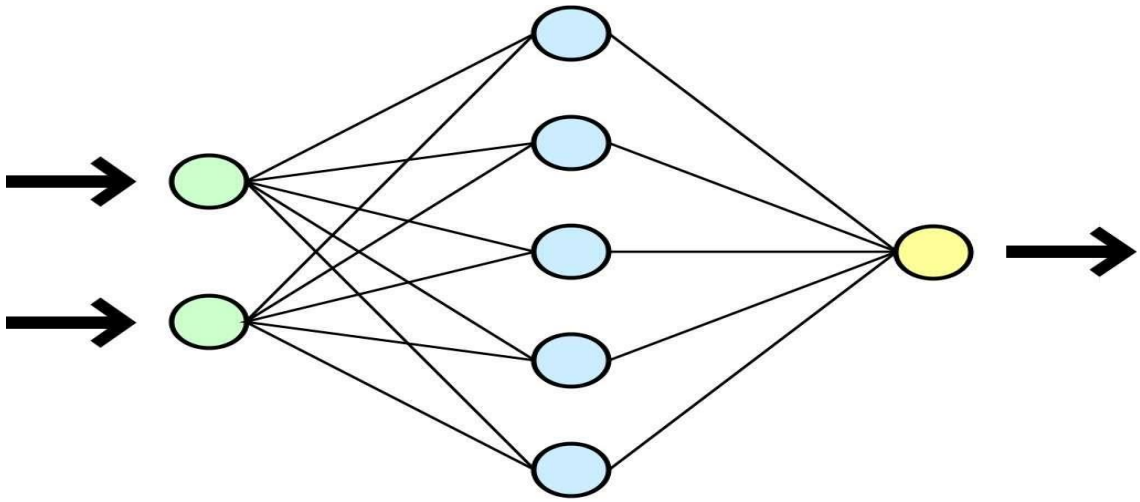


*Figure I.3 Structure d'un Réseau de neurones.*

## 2.4. Les Application de l'intelligence artificielle

L'intelligence artificielle est un domaine généralement complet pour les entreprises d'industrie. Il est organisé en deux catégories principales :

- La vision, le langage et la robotique qui s'appuient sur les briques fondamentales de l'IA.
- Le marketing : les ressources humaines, la comptabilité et la cybersécurité qui sont des fonctions horizontales dans les entreprises et font appel aux briques technologiques de l'IA ainsi qu'aux trois domaines précédents, en fonction des besoins.

### 2.4.1. Vision

La vision artificielle est l'application la plus courante de l'intelligence artificielle. C'est l'une des principales applications de l'apprentissage profond. Dans ce domaine les progrès de l'intelligence artificielle ont été plus performant au cours des 10 dernières années.

La recherche continue de progresser dans ce domaine, surtout dans les techniques de reconnaissance d'images afin d'augmenter au maximum le niveau sémantique d'identification des personnes. [3]


*Figure I.4 Exemple de détection d'objets*

### 2.4.2. Langage

Le traitement du langage est la deuxième application d'importance de l'intelligence artificielle, en plus du traitement d'image. Il comprend de nombreuses fonctions, y compris la reconnaissance vocale, Conversation Robots, la traduction automatique, l'exploration de données, la création de résumés et la génération de texte. [3]


*Figure I.5 Exemples des applications de langage.*

### 2.4.3. Robots

La robotique est une extension de l'IA, qui comprenne : capteurs, mécanique, batteries et logiciels. [3]

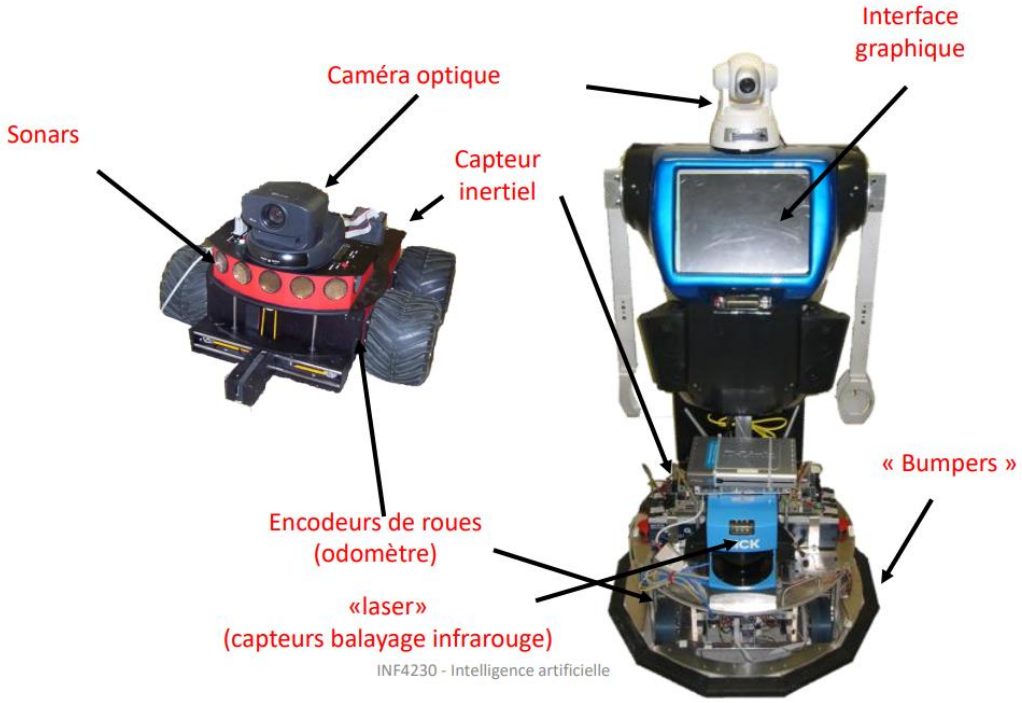



*Figure I.6 Capteurs et effecteurs d'un robot.*

## 3. Apprentissage Automatique

L'apprentissage automatique consiste à extraire les connaissances à partir des données. Il s'agit d'un domaine de recherche à l'intersection des statistiques, de l'intelligence artificielle et de l'informatique et il est également connu sous le nom d'analyse prédictive ou d'apprentissage statistique. L'application des méthodes d'apprentissage automatique est devenue côte à côte avec notre vie quotidienne au cours des dernières années. Des recommandations automatiques des films à regarder, à ce que la nourriture à commander ou quels produits à acheter, à la radio en ligne personnalisée et la reconnaissance de vos amis dans vos photos, de nombreux sites Web et appareils modernes ont des algorithmes d'association de machines à leur cœur. Lorsque vous regardez un site web complexe comme Facebook, Amazon ou Netflix, il est très probable que chaque partie du site contient plusieurs modèles d'équipe de machines. En dehors des applications commerciales, l'apprentissage automatique a eu une influence énorme sur la façon dont la recherche sur les données est effectuée aujourd'hui. [8]

### 3.1. Diverses définitions

La branche de l'intelligence artificielle qui s'occupe de développer des algorithmes pour rendre une machine capable d'exécuter des tâches complexes sans être explicitement programmée pour cela [9]

Exemple : comment écrire un programme qui reconnaisse les caractères manuscrits ?

- Entrer des règles manuellement (difficile et peu fiable).
- Meilleure méthode : écrire un algorithme (générique) qui génère automatiquement un programme de reconnaissance de caractères à partir d'un grand nombre d'exemples. [9]

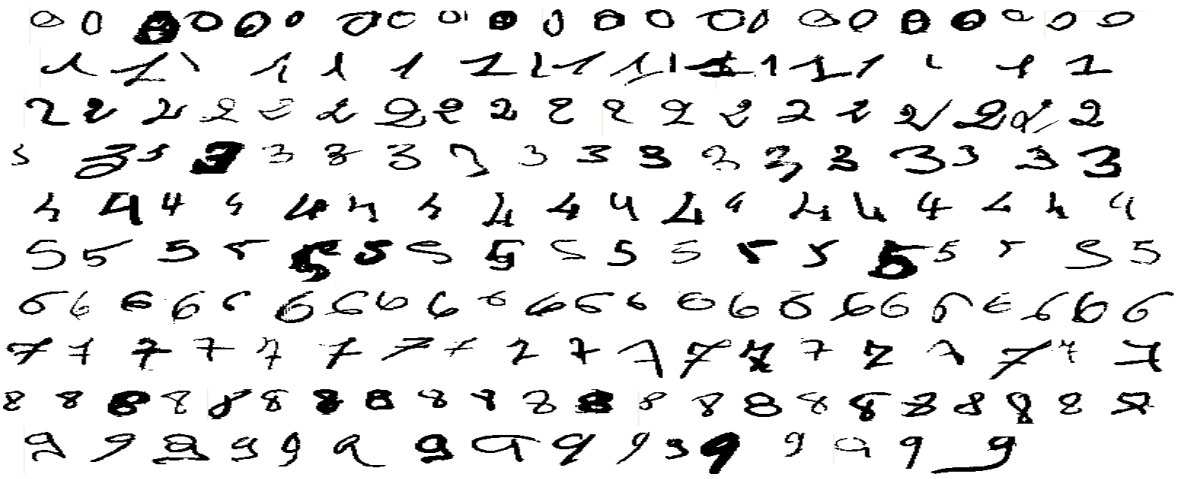


*Figure I.7 Exemple d'un jeu de données pour la reconnaissance manuscrite.*

### 3.2. Les Différents Types d'apprentissage automatiques

Nous présenterons différents types d'apprentissage utilisés dans l'apprentissage automatique. Nous verrons un apprentissage supervisé, non supervisé, semi-supervisé et renforcé. Chaque type d'apprentissage a sa propre spécificité et il est utilisé en fonction du problème auquel nous sommes confrontés.

#### 3.2.1. Apprentissage supervisé

L'apprentissage supervisé est un programme d'apprentissage où les données sont connues à l'avance, Par exemple, dans le cas d'un classificateur d'images, nous aurons un jeu de données intitulé. Ainsi, nous connaissons le nom de chaque image. Par ces autocollants, Nous pourrons enseigner à notre modèle les différentes classes représentées. Ainsi, chaque fois que notre modèle prédit, nous pouvons dire s'il a donné une bonne prédiction ou si elle était fausse. En conséquence, notre système apprend de ses erreurs. Il essaiera de réduire l'erreur en fonction des données que nous lui fournirons. [10]

Généralement, dans un apprentissage supervisé, Nous avons trois étapes principales. Une étape d'apprentissage, où nous allons donner à notre modèle un ensemble de données, ou il pourra s'entraîner. Puis une phase de validation, où nous allons chercher à vérifier l'apprentissage de notre système. Pour ce faire, nous allons prendre un ensemble de données que nous n'avons jamais montrer à notre modèle. Puis, nous allons chercher à vérifier si notre modèle arrive à bien différencier les différentes catégories. Enfin, nous avons une phase de production où nous allons pouvoir utiliser notre système. [10]

Représentation mathématique d'un Apprentissage supervisé

On reçoit des données d'exemple annotées : $(x^1, y^1)$, $(x^2, y^2)$, $(x^3, y^3)$, et on espère prédire la sortie sur de nouvelles observations : $x^* \rightarrow y^*$

Quelques algorithmes d'apprentissage supervisé

- Régression linéaire
- Régression logistique
- Arbres de classification et de régression
- K-NN
- Naïve Bayes Classifier

### 3.2.2. Apprentissage non supervise

Dans l'apprentissage non supervisé, nous ne connaissons pas la catégorie de nos données. Contrairement à l'apprentissage supervisé, ici nous n'avons pas d'autocollant sur nos données. Ainsi, l'objectif de notre modèle sera d'identifier les similitudes et, en conséquence, d'identifier des données spécifiques appartenant à la même classe.

Pour Clarifier ce type d'apprentissage, nous disposons d'un ensemble de données provenant d'un groupe d'animaux. Nous avons rassemblé un ensemble de caractéristiques telles que l'âge, le poids, la taille ... le but sera de déterminer si certains animaux peuvent appartenir à la même espèce et ainsi les regrouper [10]

Représentation mathématique d'un Apprentissage non supervisé

On reçoit uniquement des observations brutes de variables aléatoires : $x^1$, $x^2$, $x^3$, $x^4$, ... et on espère découvrir la relation avec des variables latentes structurelles : $x^i \rightarrow y^i$

Quelques algorithmes d'apprentissage supervisé

- K-MEANS
- Classification hiérarchique
- Le clustering par décalage moyen
- Apriori

### 3.2.3. Apprentissage semi-supervise

L'apprentissage semi-supervisé est un paradigme éducatif qui consiste à étudier comment les ordinateurs et les systèmes naturels comme les humains apprennent en présence de données à la fois classifiées et non classifiées. Traditionnellement, l'apprentissage a été étudié soit dans le modèle non supervisé (par exemple, agrégation, divulgation externe) où toutes les données ne sont pas classifiées, soit dans le modèle supervisé (par exemple, classification, régression) où toutes les données sont classées. L'objectif de l'apprentissage semi-supervisé est de comprendre comment la combinaison de données classifiées et non classifiées peut modifier le comportement d'apprentissage, et de concevoir les algorithmes qui tirent parti de cette combinaison. L'apprentissage semi-supervisé est d'une grande importance dans l'apprentissage automatique et l'exploration de données, car il peut utiliser des données non étiquetées facilement disponibles pour améliorer les tâches d'apprentissage supervisé lorsque les données désagrégées sont rares ou coûteuses. L'apprentissage semi-supervisé montre également qu'il peut être utilisé comme un outil quantitatif pour comprendre l'apprentissage humain en classe, car la plupart des intrants ne sont clairement pas classés. Dans ce livre d'introduction, nous présentons quelques modèles d'apprentissage semi-supervisé courants, y compris l'auto-coaching, les modèles mixtes, l'apprentissage croisé et multi-présentation, les méthodes basées sur des graphiques et les machines de support semi-supervisées. Pour chaque modèle, nous discutons de sa formulation mathématique de base. Le succès d'un apprentissage semi-supervisé dépend essentiellement de certaines hypothèses de base. Nous confirmons les hypothèses faites par chaque modèle et fournissons des contre-exemples le cas échéant pour démontrer les limites des différents modèles. De plus, nous discutons de l'apprentissage semi-supervisé pour la psychologie cognitive. Enfin, nous proposons une perspective théorique de l'apprentissage informatique sur l'apprentissage semi-supervisé et concluons le livre par une brève discussion des questions ouvertes sur le terrain. [11]

## 4. Apprentissage profond

L'apprentissage profond (en anglais, traduction : deep learning) est une forme d'intelligence artificielle, dérivée de l'apprentissage automatique. Pour comprendre ce qu'est le deep learning, il est essentiel de comprendre ce qu'est le machine learning.

### 4.1. Qu'est-ce que l'apprentissage profond

L'apprentissage profond (deep-learning) est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années.

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données.

L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

Par exemple, pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles… des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes…

Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données (big data). [12]

### 4.2. Les algorithmes de l'apprentissage profond

Il existe différents algorithmes pour l'apprentissage en profondeur. Donc on peut citer :

- Les réseaux de neurones convolutionnels (CNN ou Convolutional Neural Networks).
- Réseau de neurones récurrents (RNN ou *recurrent neural network*).
- Les auto encodeur (autoencoder).
- Réseaux antagonistes génératifs (GAN ou Generative adversarial networks).

En va expliquer le réseau de neurone **convolutifs,** vue son grande importance en ce mémoire.

### 4.3. Les réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs sont de loin le modèle le plus efficace pour classer les images, Celui-ci compare les images en fragment. Les fragments qu'il recherche sont appelés caractère. En trouvant des caractères approximatifs qui sont presque similaires dans 2 images différentes. CNN est beaucoup mieux à détecter les similitudes en comparant l'image à l'image. Chaque fonctionnalité est comme une miniature c'est à dire un petit tableau de valeurs en 2 dimensions. [13]

Contrairement à nous les êtres humains qui voyons les formes et les couleurs ; l'ordinateur ne voit que des nombres organisés comme suit :

- 224×224 pixels pour l'image, où le premier 224 est la largeur de l'image et le second est la hauteur
- 3 couches de ces 224×224 pixels, chacune correspondante à une coloration :
    - Rouge (r pour Red)
    - Vert (g pour green)
    - Bleu (b pour Blue)

### 4.3.1. La couche convolution

La couche convolutive est le composant clé des réseaux de neurones convolutifs, et elle forme toujours au moins la première couche.

La couche de convolution reçoit donc plusieurs images à titre d'entrée, et calcule la convolution de chacun d'eux avec chaque filtre. Les filtres correspondent exactement aux fonctionnalités que vous souhaitez trouver dans les images. [14]

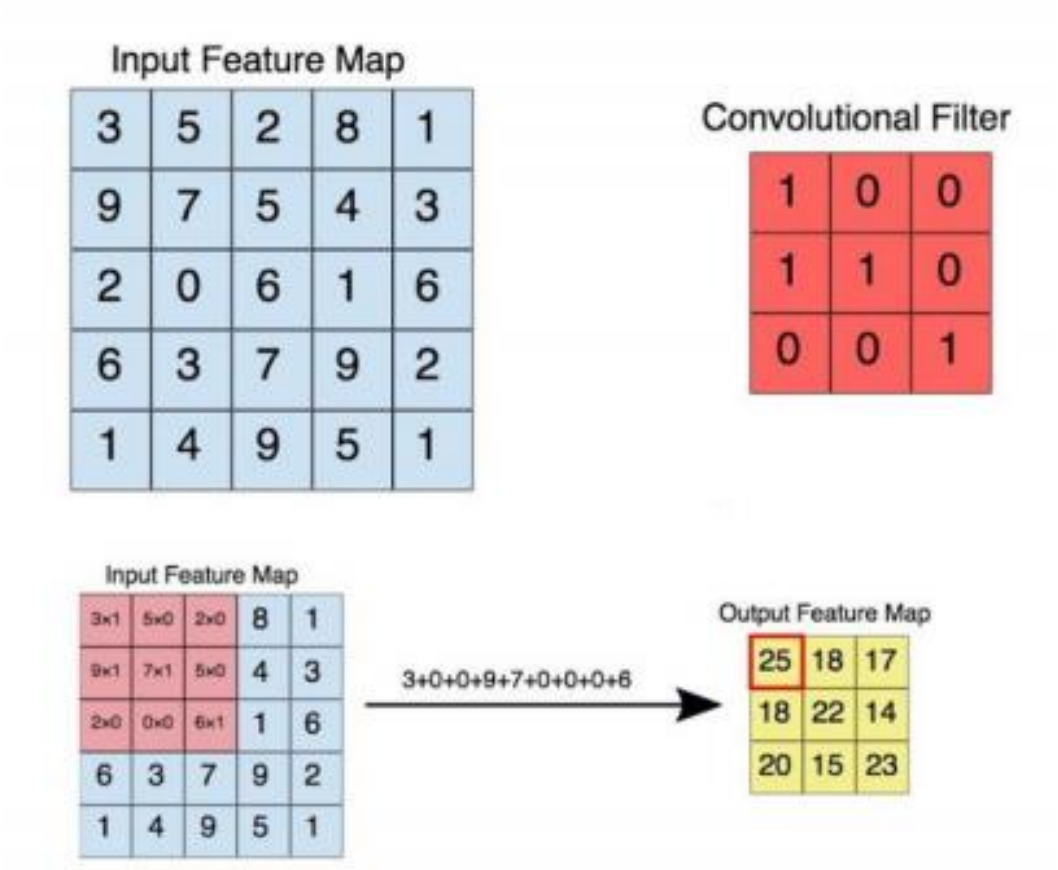



*Figure I.8 Exemple de la couche convolution.*

### 4.3.2. La couche max-pooling

L'opération de *pooling* consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour Cela, l'image est coupée en cellules normales, puis la valeur maximale est conservée dans chaque cellule. Dans la pratique, les petites cellules carrées sont souvent utilisées pour ne pas perdre trop d'informations. [14]

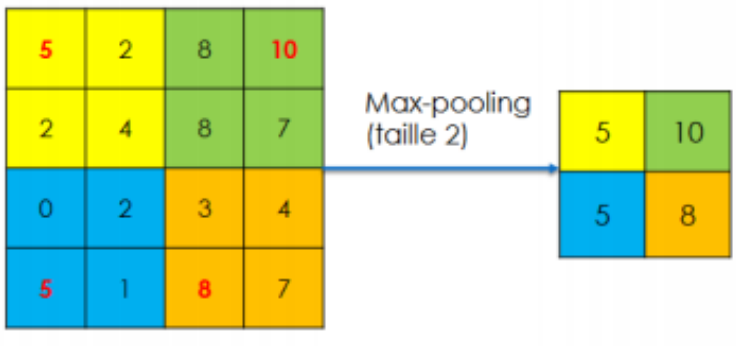



*Figure I.9 Exemple de la couche max-pooling*

## 5. Conclusion

L'intelligence artificielle est un domaine très vaste. Dans ce chapitre, nous avons vu les concepts importants en apprentissage automatique ; évidemment, il reste d'autres concepts et d'autres techniques d'apprentissage, nous avons essayé de faire un balayage sur tout ce qui concerne l'apprentissage automatique, nous avons mis l'accent sur ce qui est essentiel pour notre travail. Nous avons aussi présenté une étude sur les réseaux de neurones convolutionnels.

Dans le chapitre suivant, nous effectuons une étude des approches les plus connu de la détection, la reconnaissance et la reconstruction 3D.

# Chapitre II :

# La Détection et la Reconstruction 3D

# 1. Introduction

La recherche sur l'apprentissage automatique peut générer un modèle 3D d'un visage humain à partir d'une image utilisant des réseaux neuronaux :

La reconstruction 3D du visage est un problème fondamental de la vision artificielle, d'une difficulté extraordinaire. Les systèmes actuels supposent souvent la disponibilité d'images faciales multiples (parfois du même sujet) comme intrants et doivent aborder un certain nombre de défis méthodologiques comme l'établissement de correspondances denses entre les grandes poses faciales, les expressions et la lumière non uniforme.

En général, ces méthodes requièrent des canalisations complexes et inefficaces pour la construction et l'installation de maquettes.

Dans ce chapitre nous allons présenter les différentes techniques disponibles pour détecter les visages humains, il existe plusieurs, nous allons présenter quelques approches, qui sont les plus connues, et nous allons présenter les techniques de reconnaissance faciale, après nous présentons les différentes façons pour construire des modèles 3D.

# 2. Techniques pour la détection des visages

La détection est la première étape dans le processus de la reconstruction faciale. Son efficacité a une influence directe sur les performances du système de reconstruction de visages.

Il existe plusieurs méthodes pour la détection de visages, certaines utilisent la couleur de la peau, la forme de la tête, l'apparence faciale, alors que d'autres combinent plusieurs de ces caractéristiques.

Les méthodes de détection de visages peuvent être subdivisées en quatre catégories [15] :

## 2.1.Approches basées sur les connaissances acquises

Cette méthodologie s'intéresse aux parties caractéristiques du visage par la détection des contours du visage comme le nez, la bouche et les yeux. Elle est basée sur la définition de règles strictes à partir des rapports entre les caractéristiques faciales. Ces méthodes sont conçues principalement pour la localisation de visage.

L'inconvénient de cette méthode est qu'elle n'arrive pas à détecter le visage lorsque le visage est sur un arrière-plan complexe de sorte que cette technique provoque de nombreuses fausses détections et un faible taux de détection.

## 2.2.Approches basées sur le « Template-matching »

Les templates dit aussi « moules » peuvent être définis soit "manuellement", soit paramétrés à l'aide des fonctions.

Cette approche consiste à évaluer et calculer la corrélation entre l'image candidate et le template (moule). Ces méthodes rencontrent encore quelques problèmes de robustesse liés aux variations de lumière, d'échelle, etc.

*Figure II.1 Différentes régions utilisées pour la phase de template matching. [16]*

La figure. II.2 montre un modèle prédéfini correspondant à 23 relations. Ces relations prédéfinies sont classifiées en 11 relations essentielles (flèches) et 12 relations confirmations (gris). Chaque flèche représente une relation entre deux régions. Une relation est vérifiée si le rapport entre les deux régions qui lui correspond dépasse un seuil. Le visage est localisé si le nombre de relations essentielles et de confirmation dépasse lui aussi un seuil.

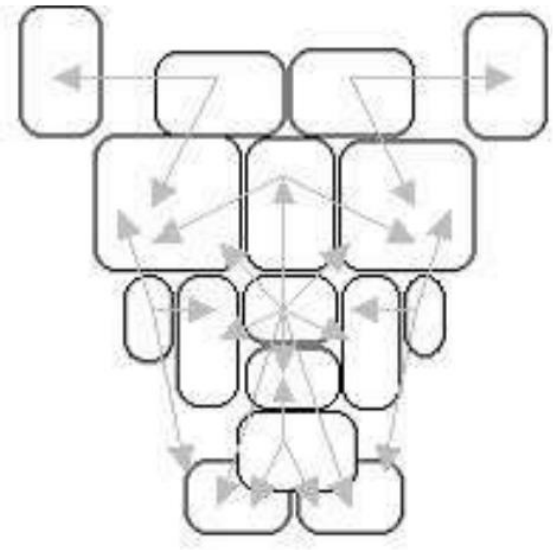


*Figure II.2 Modèle de visage composé de 16 régions (les rectangles) associées à 23 relations (flèches). [17]*

### 2.3.Approches basées sur l'apparence

Ces approches appliquent généralement des techniques d'apprentissage automatique. Ainsi, les modèles sont appris à partir d'un ensemble d'images représentatives de la variabilité de l'aspect facial. Ces modèles sont alors employés pour la détection.

L'idée principale de ces méthodes est de considérer que le problème de la détection de visage est un problème de classification (visage, non-visage). Une des approches les plus connues de détection de visage est l'Eigenface [18], elle consiste à projeter l'image dans un espace et à calculer la distance euclidienne entre l'image et sa projection. En effet, en codant l'image dans un espace, on dégrade l'information contenue dans l'image, puis on calcule la perte d'information entre l'image et sa projection. Si cette perte d'information est grande (évaluée à partir de la distance, que l'on compare à un seuil fixé a priori), l'image n'est pas correctement représentée dans l'espace : elle ne contient pas de visage. Cette méthode donne des résultats assez encourageants, mais le temps de calcul est très important.

*Figure II.3 Eigenfaces calculés à partir de l'image d'entrée. [18]*

*Figure II.4 Une image de visage original et ça projection sur l'espace des eigenfaces défini dans la figure II.3. [18]*

Une méthode bien connue de détection d'objets complexes tels que les visages est l'utilisation de « classifieurs de Haar » montés en cascade (boostés) au moyen d'un algorithme AdaBoost. Cette méthode est implémentée nativement dans la bibliothèque OpenCV [19] et a été présenté initialement dans Viola et Jones [20]. Le principe de cette méthode est un algorithme complexe de classification, composé de classifieurs élémentaires qui éliminent au fur et à mesure les zones de l'image qui ne sont pas compatibles avec l'objet recherché. Ces classifieurs binaires reposent sur des primitives visuelles qui dérivent des fonctions de Haar (Haar- like features).

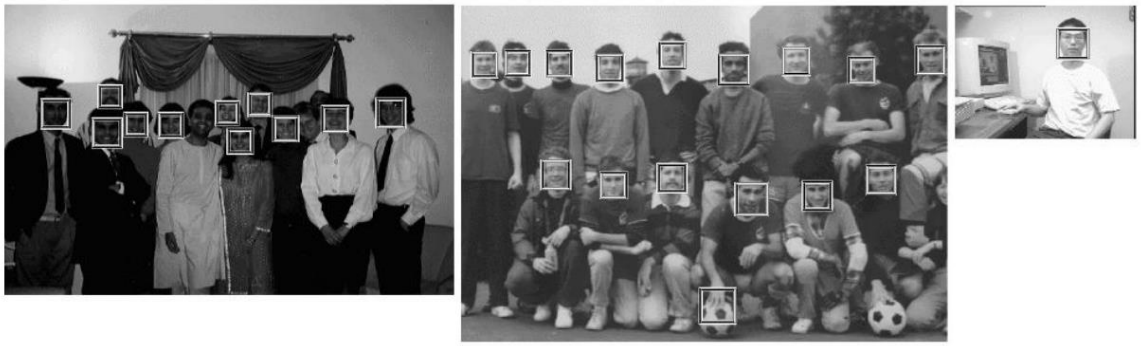


*Figure II.5 Sortie de Haar-Cascade sur un certain nombre d'images de test obtenus à partir les jeux de données MIT+CMU. [20]*

### 2.4.Approches basées sur des caractéristiques invariantes

Ces approches sont utilisées principalement pour la localisation de visage. Les algorithmes développés visent à trouver les caractéristiques structurales existantes même si la pose, le point de vue, ou la condition d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages. Nous pouvons citer deux familles de méthodes appartenant à cette approche : Les méthodes basées sur la couleur de la peau et les méthodes basées sur les caractéristiques de visage, elles consistent à localiser les cinq caractéristiques (deux yeux, deux narines, et la jonction nez/lèvre) pour décrire un visage typique.

#### Détection de visages basée sur l'analyse de la couleur de la peau

Les méthodes de détection basées sur l'analyse de la couleur de la peau sont des méthodes efficaces et rapides. Elles réduisent l'espace de recherche de la région visage dans l'image. De plus, la couleur de la peau est une information robuste face aux rotations, aux changements d'échelle, et aux occultations partielles. Plusieurs espaces couleur peuvent être utilisés pour détecter, dans l'image, les pixels qui ont la couleur de la peau. L'efficacité de la détection dépend essentiellement de l'espace couleur choisi. Les espaces couleur les plus utilisés sont :

- L'espace RVB, mis au point en 1931 par la Commission Internationale de l'Eclairage (CIE). Il consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs : Rouge-Vert-Bleu. Cet espace correspond à la façon dont

les couleurs sont généralement codées informatiquement, ou plus exactement à la manière dont les écrans à tubes cathodiques (ordinateurs, TV) représentent ces couleurs.

- L'espace HSL (Hue, Saturation, Luminance), appelé aussi TSL (Teinte, Saturation, Luminance) en Français, s'appuie sur les travaux du peintre Albert H. Munsell. C'est un modèle de représentation dit "naturel", car il est proche de la perception physiologique de la couleur par l'œil humain. En effet, le modèle RGB aussi adapté soit-il pour la représentation informatique de la couleur ou bien l'affichage sur les périphériques de sortie, ne permet pas de sélectionner facilement une couleur. Le modèle HSL consiste à décomposer la couleur selon des critères physiologiques :
    - La teinte (en Anglais Hue), correspondant à la perception de la couleur,
    - La saturation, décrivant la pureté de la couleur, c'est-à-dire son caractère vif ou terne,
    - La luminance, indiquant la quantité de lumière de la couleur, c'est-à-dire son aspect clair ou sombre.

Il existe d'autres modèles naturels de représentation proches du modèle HSL comme (HSB, HSV, HSI, HCI, YCrCb …).

Les techniques de détection du visage basées sur la couleur de la peau peuvent être classifiées en quatre catégories suivantes : les méthodes explicites, les méthodes non paramétriques, les méthodes paramétriques, et les méthodes semi paramétriques. Toutes ces approches pratiquent une phase d'apprentissage sur un nombre d'images représentatives pour calculer une densité de probabilité de la couleur peau.

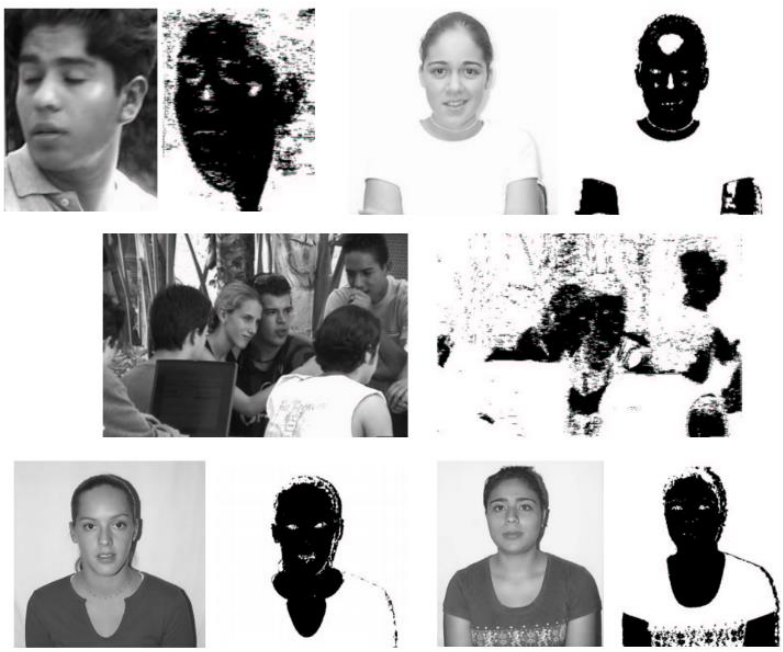


*Figure II.6 Exemples d'entrées et leurs résultats avec une méthode explicite. [21]*

## 3. Techniques pour la reconnaissance des visages

La reconnaissance 3D de visages constitue une alternative prometteuse pour surmonter les problèmes de la reconnaissance 2D de visages, surtout depuis l'apparition de dispositifs d'acquisition 3D performant. L'avantage principal des approches basées modèle 3D réside dans le fait que le modèle 3D conserve toutes les informations sur la géométrie de visage, ce qui permet d'avoir une représentation réelle de ce dernier. Dans cette section, après avoir rapidement évoqué les systèmes d'acquisition 3D, nous présenterons les travaux récents sur la reconnaissance 3D de visages.

### 3.1.Principales difficultés de la reconnaissance de visage

Pour le cerveau humain, le processus de la reconnaissance de visages est une tâche visuelle de haut niveau. Bien que les êtres humains puissent détecter et identifier des visages dans une scène sans beaucoup de peine, construire un système automatique qui accomplit de telles tâches représente un sérieux défi. Ce défi est d'autant plus grand lorsque les conditions d'acquisition des images sont très variables. Il existe deux types de variations associées aux images de visages: inter et intra sujet. La variation inter-sujet est limitée à cause de la ressemblance physique entre les individus. Par contre la variation intra-sujet est plus vaste. Elle peut être attribuée à plusieurs facteurs comme : le changement d'illumination, la variation de pose ou l'expressions faciales…

### 3.2.Systèmes d'acquisition 3D

Généralement le modèle du visage est représenté par des images 2.5D et 3D (voir la Figure. II.7). L'image 2.5D (image de profondeur) correspond à une représentation bidimensionnelle d'un ensemble de points 3D (x,y,z) où chaque pixel dans le plan X-Y stocke la valeur de sa profondeur z. On peut assimiler une image 2.5D à une image en niveau de gris où les pixels noirs correspondent au fond tandis que les pixels blancs représentent les points de surface les plus proches de la caméra. Par ailleurs, la méthode la plus simple pour représenter un visage 3D est le maillage polygonal 3D, ce dernier correspond à une liste de points connectés par des arêtes (polygones). Il existe plusieurs techniques pour construire un maillage 3D, les plus utilisées combinent des images 2.5D ou bien exploitent des systèmes d'acquisition 3D tel que le scanner 3D.

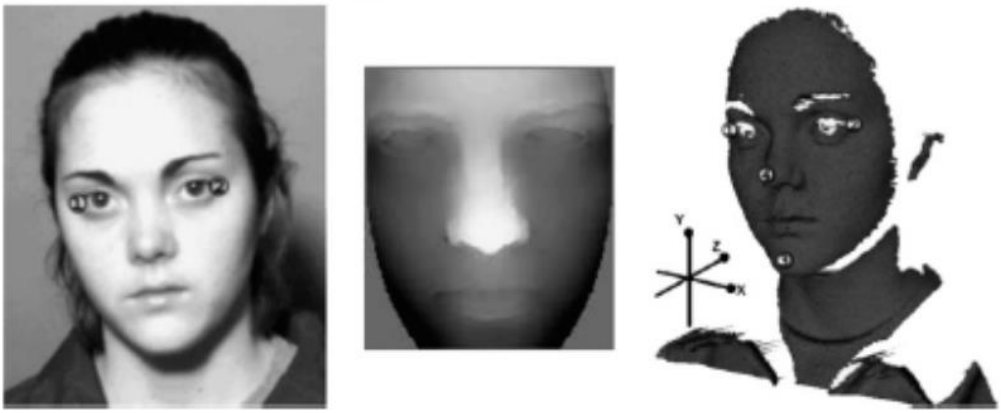


*Figure II.7 A gauche l'image texture, transformée en 2.5D (au centre) et l'image 3D.*

Les techniques de reconnaissance 3D de visages peuvent être regroupées en trois catégories principales : approches basées modèle, approches 3D, et approches multimodales 2D + 3D qui sont des techniques qui combinent les données 2D et 3D sur le visage pour améliorer les performances et la robustesse de la reconnaissance.

### 3.3.Approches modèle

Ces approches construisent à partir des points 3D, des modèles de visages qu'elles utilisent par la suite pour la reconnaissance.

Blanz et al.[22] [23] ont proposé une méthode basée sur un modèle 3D « morphable » du visage. L'ensemble des visages est représenté par un espace vectoriel [24]. La base de données contient 100 visages d'hommes et 100 visages de femmes acquis avec un scanner laser Cyberware™ 3030PS. Les points 3D des modèles de visages générés sont représentés par leurs coordonnées cylindriques définies par rapport à un axe vertical. Pour chaque visage de référence, les coordonnées et les valeurs de texture de tous les sommets (n = 75972) sont regroupées pour former deux vecteurs : un vecteur de forme et un vecteur de texture. Une fois le modèle générique créé, l'étape suivante consiste à l'ajuster sur l'image 2D à partir des paramètres de forme et de texture. La synthèse d'image permet de rendre les nouvelles positions projetées des sommets du modèle 3D, à l'aide l'illumination et la couleur extraites. Enfin, l'étape de reconnaissance est réalisée en mesurant la distance de Mahalanobis [25] entre la forme et les paramètres de texture des modèles dans la galerie et le modèle d'ajustement.

L'identification a été évaluée sur deux bases de données d'images à accès libre, Un taux de reconnaissance de 95% sur l'ensemble de données CMU-PIE et 95.9% sur l'ensemble de données FERET a été obtenu.

### 1.1.1 Approches 3D

Elles sont subdivisées en deux catégories : les approches basées surface qui utilisent la géométrie de la surface du visage et les approches holistiques 3D.

#### a. Approches surface

Dans ce cas, le problème de la reconnaissance 3D de visages est celui de l'alignement de deux surfaces 3D qui modélisent les deux visages à apparier. L'algorithme généralement utilisé est l'algorithme du plus proche voisin itéré, ou ICP (Iterative Closest Point), qui a été introduit par [26].

#### b. Approches holistiques 3D

Les techniques holistiques comme l'ACP ont été largement utilisées dans la reconnaissance faciale 2D. Plus récemment, ces techniques ont été aussi étendues aux données 3D de visage.
Les techniques basées ACP ont également été combinées avec d'autres méthodes de classification, comme le modèle caché de Markov (EHMM) puis appliquées à la reconnaissance 3D de visages [27]. Enfin, d'autres approches basées sur l'Analyse Discriminante Linéaire [28] ou l'Analyse des Composantes Indépendantes [29] ont aussi été développées pour l'analyse des données 3D de visages.

**L'analyse en Composantes Principales (ACP)** est une technique particulièrement prisée par les chercheurs de la communauté de la biométrie. Elle est utilisée soit de façon globale sur toute l'image du visage, soit de façon modulaire sur les différentes régions faciales. De plus, plusieurs extensions de l'ACP ont aussi été proposées et utilisées pour la reconnaissance faciale.

## 4. Techniques de Reconstruction 3D du visage

La problématique de la reconstruction géométrique 3D est un domaine de recherche très vaste. Depuis le début des années 70, il a suscité beaucoup d'intérêt dans les domaines de la recherche. Plusieurs approches sont apparues : les approches monoculaires qui estiment la géométrie de la scène à partir d'une seule vue, les approches multi-vues construites sur l'utilisation de plusieurs vues prises simultanément ou encore les approches construites sur l'analyse d'une ou plusieurs vues prises à instants différents. Selon les applications visées, ces techniques œuvrent en lumière ambiante, nécessitent une lumière contrôlée qui peut parfois projeter un motif structuré.

### 4.1.Les Méthodes Monoculaires

Ces approches dépendent d'une seule caméra, s'appuient sur l'extraction de certaines caractéristiques de l'image afin de déterminer l'information de profondeur, cette information

caractéristique peut être l'éclairement (Shape From Shading), la déformation de texture (Shape From Texture), la variation des paramètres de mise au point de l'objectif utilisé (Shape from Focus/Defocus), ou encore des approches par temps de vol de la lumière (Caméras à temps de vol).

L'ensemble des approches monoculaires utilisent certaines hypothèses fortes qui permettent de lever l'ambigüité de profondeur.

### 4.1.1. Shape From Shading

L'approche « Shape From Shading » estime la forme d'objets éclairés dans une image, à partir des variations graduelles de l'éclairement observé (voir figure II.8). Ayant une image en niveau de gris, le but est de déterminer la position des sources de lumière, ainsi que la surface de l'objet pour chaque pixel de l'image. Même si le modèle d'éclairement est supposé lambertien, que la direction des sources de lumière est connue et que l'information de luminance peut être décrite en fonction des sources de lumière et de la surface, le problème n'en est pas moins difficile. En dehors des problèmes de leur robustesse de ce type d'approche, les contraintes algorithmiques nécessaires à la résolution de Shape From Shading sont encore loin des contraintes temps réel.

*Figure II.8 Résultat de l'approche Shape From Shading proposée par Meyer et al. dans [30]. A gauche l'image source, transformée en niveaux de gris (au centre) et la reconstruction correspondante par Shape From Shading.*

### 4.1.2. Shape From Texture

Les approches Shape from texture estiment la forme des objets en fonction des variations observées dans la texture. Ainsi les objets à construire doivent être texturés que ce soit naturellement ou artificiellement (voir les figures II.9 et II.10).

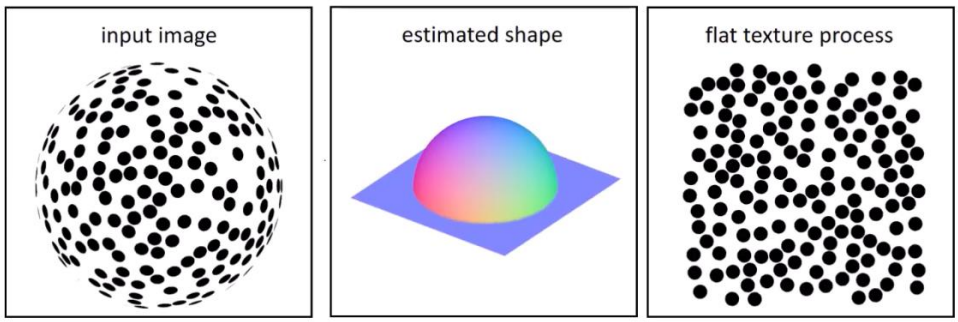



*Figure II.9 Reconstruction de forme à partir de l'approche Shape From Texture. L'image de gauche représente l'image source, la géométrie estimée est représentée au centre, enfin la texture générée est présentée sur l'image de droite.*

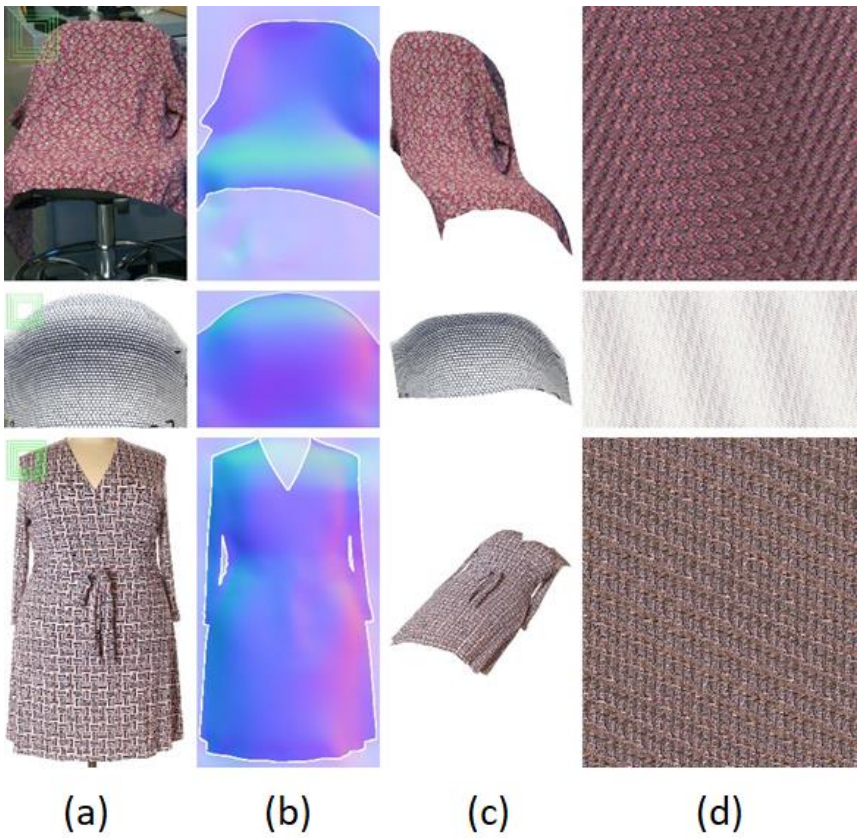



*Figure II.10 Reconstruction de forme à partir de l'approche Shape From Texture proposée par Verbin et Zickler dans [31]*

La figure II.10 montre une reconstruction d'une forme en utilisant l'approche Shape From Texture dans [31], (a) représente l'image source (l'entrée), la géométrie estimée sans texture est représentée dans (b), la géométrie texturée est représentée dans (c), enfin (d) représente la texture générée.

### 4.1.3. Shape from Focus/Defocus

L'estimation de la profondeur « Depth From Defocus » ou de la forme « Shape From Defocus » consiste à reconstruire la forme 3D à partir de plusieurs images prises depuis une caméra fixe, avec différentes valeurs de mise au point. Avec certaines hypothèses de simplification sur la caméra, ce problème peut être posé comme étant l'inversion d'équations d'intégration qui décrivent le processus de création de l'image. Cette approche utilise le fait que, l'image de la scène sur le plan image, dépend de la radiance de la région, autant que de la forme de cette région. Cependant la valeur de chaque pixel provient de l'intégration d'une radiance inconnue par un noyau de convolution inconnu qui dépend aussi de la forme de la scène. Connaissant la valeur de l'intégrale en chaque pixel il faut alors estimer la valeur du noyau et de la radiance correspondante.

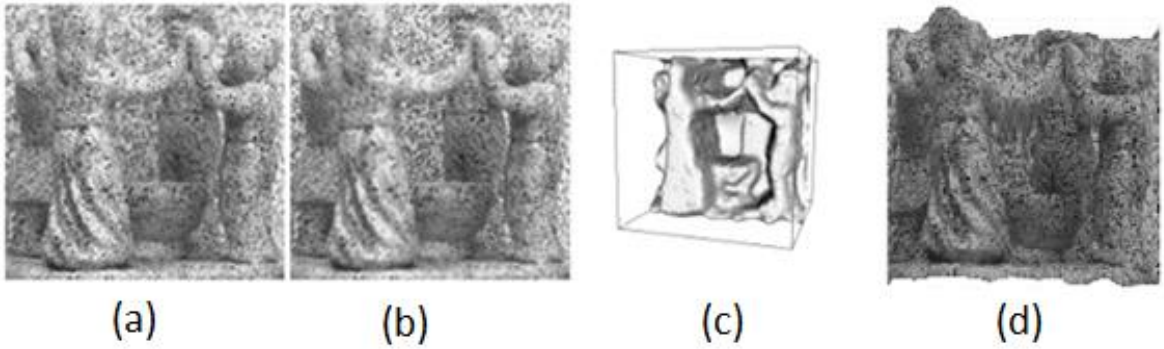



*Figure II.11 Estimation de carte de profondeur utilisant l'approche Shape From Defocus proposée par Jin et Favaro dans [32].*

Dans la Figure II.11 Les images (a) et (b) représentent respectivement deux images de la même scène avec des paramètres de mise au point différents. L'image de gauche (a) est produite avec une mise au point au loin, alors que l'image de droite (b) est acquise avec une mise au point courte. L'image (c) représente l'estimation de profondeur correspondante. Notons que la scène est très texturée, Enfin la forme finale texturée est représentée à la droite dans (d).

### *4.1.4 Caméras 2,5D à temps de vol*

Les caméras temps de vol, représenté sur la figure II.12 sont des capteurs actifs qui fournissent des images de profondeur en temps réel, ils envoient un signal optique proche infrarouge, le signal réfléchi par les objets de la scène est ensuite détecté par le capteur qui calcul la profondeur. Il existe un premier type de temps de vol qui se base sur des impulsions, il consiste à mesurer le temps que met le signal pour effectuer le trajet entre l'objet et la caméra. Un autre type qui se base sur des ondes de modulation continues, il consiste à mesurer le déphasage entre

le signal émis et celui réfléchi par démodulation synchrone du signal réfléchi. Le principe de ce dernier type est basé sur la détection et la sauvegarde des charges photo-électriques par synchronisation des électrodes des pixels. Les caméras à temps de vol sont beaucoup utilisées, elles ont l'avantage d'acquérir les images 1à haute fréquence, de consommer peu d'énergie et d'avoir un plus faible poids.


*Figure II.12 Exemple de caméra temps de vol*

### 4.2.Les Méthodes multi-vues

Les approches monoculaires permettent l'estimation de la géométrie de la scène à partir de l'extraction de primitives depuis un seul point de vue. La présence de conditions environnementales défavorables influe directement sur les résultats de ces approches. Afin de les rendre plus robustes et précises, une solution est de travailler sur l'analyse de plusieurs images prises simultanément depuis plusieurs points de vue.

#### 4.2.1. Stéréovision

Le processus de la Stéréoscopie est souvent décrit en trois étapes :

- ✓ Prétraitement (Calibration, Acquisition, Rectification)
- ✓ Appariement (Mise en correspondance)
- ✓ Reconstruction (Triangulation).

La difficulté principale de la technique stéréovision provient du problème de la mise en correspondance, elle consiste à estimer la position 3D par appariement, c'est à dire trouver les points dans chaque image qui correspondent aux mêmes points dans la scène 3D, puis faire la triangulation pour pouvoir déterminer la position 3D de la scène correspondante. Il existe des méthodes pour estimer la mise en correspondance, comme la méthode qui recherche les correspondances pour chaque pixel par des approches d'autocorrélation normalisées, c'est une méthode sensible au bruit et aux conditions d'éclairage dans chaque vue.

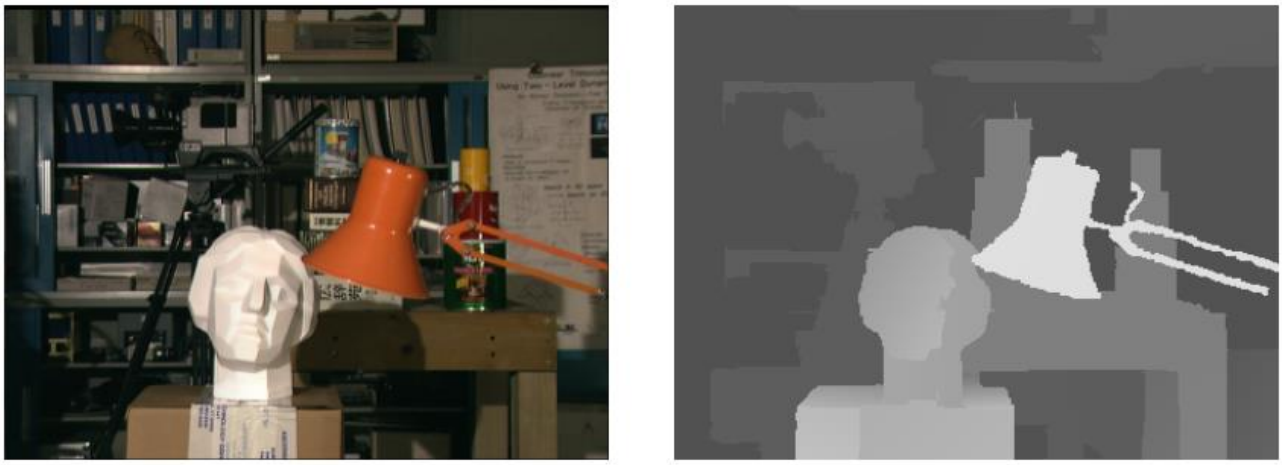


*Figure II.13 Estimation de la carte de profondeur de la scène. L'image de droite a été calculée en utilisant l'approche de Klaus et al. [33]*

### 4.2.2. Lumière Structurée

Cette approche utilise une caméra et un projecteur, comme montré sur la Figure. II.14, elle utilise le même principe de la triangulation optique que pour la stéréovision et la triangulation laser, un motif connu est projecté sur la scène, une grille souvent ou barres horizontales, la déformation de ce motif lorsqu'il heurte la surface permet le calcul de la profondeur et d'extraire des informations sur les objets présents dans la scène.

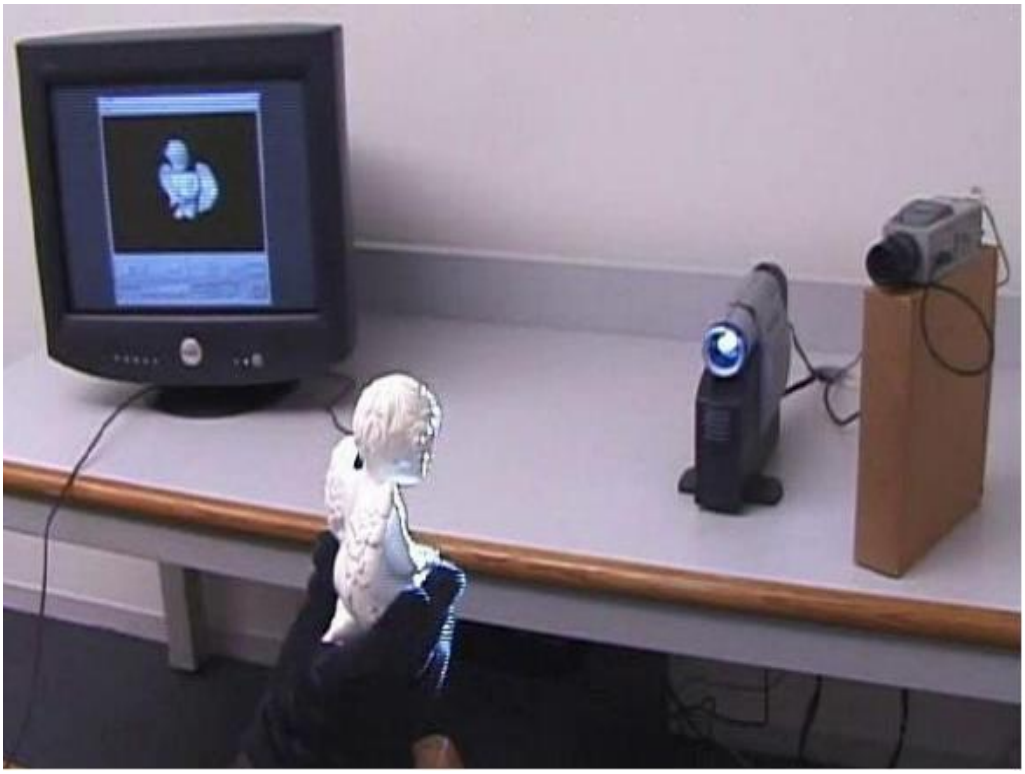


*Figure II.14 Exemple de scène de la lumière structurée.*

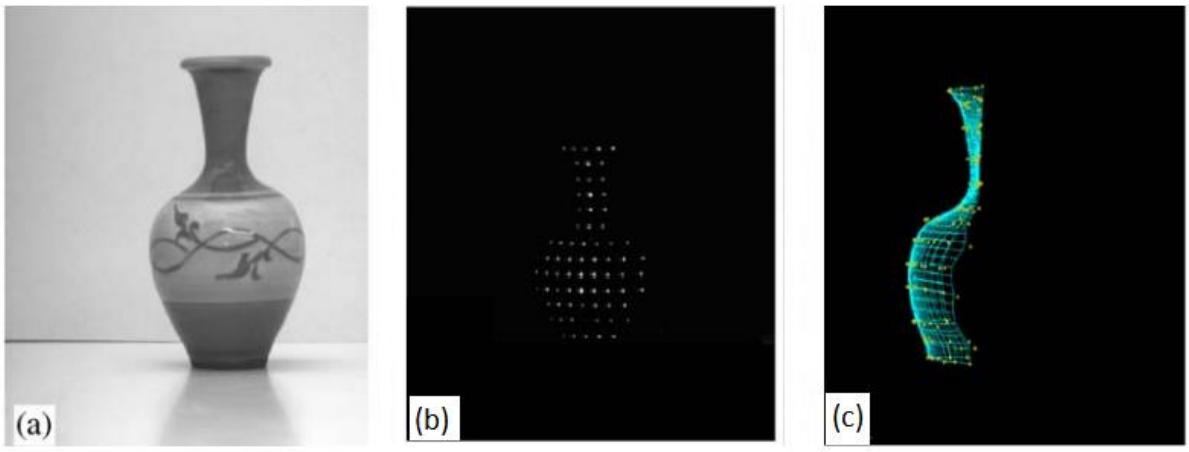



*Figure II.15 Résultat de l'approche Lumière structurée proposée par Dipanda et Woo dans [34]. A gauche l'image source en gray scale, les points acquises (au centre) et la représentation de résultat de la reconstruction 3D*

### 4.2.3. Shape-From-Silhouette

Shape-From-Silhouette est une méthode qui estime, à partir de plusieurs vues, l'enveloppe visuelle des objets d'intérêts, qui représente une estimation englobante de leur forme 3D comme montré dans la Figure. II.16. Baumgart a été le premier en 1974 à utiliser cette méthode. A partir d'un ensemble de n vues, l'information de silhouette, est extraite des images capturées par une approche dite d'extraction de silhouette. La forme produite fournit un volume englobant des objets d'intérêts.

Les approches Shape-From-Silhouette ont été utilisées pour diverses applications, telles que la surveillance de foule, la modélisation 3D ainsi que l'acquisition de mouvements sans marqueurs.

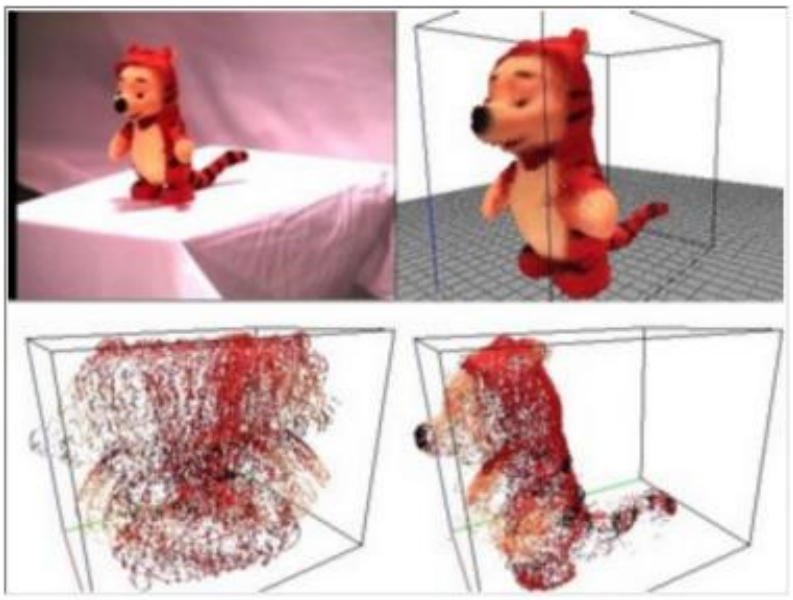


*Figure II.16 Exemple d'une reconstruction en utilisant la méthode Shape From Silhouette.*


*Figure II.17 Résultat proposée par Goldluecke et Magnor dans [35] (a) des images source obtenu à partir des caméras (b) Coque visuelle raffinée, représentant la surface initiale. (c) Résultat final après l'exécution de l'algorithme complet.*

## 5. Discussion et Présentation des Approches

La reconstruction de la structure géométrique détaillée d'un visage à partir d'une image donnée est la clé de nombreuses applications de vision et de graphisme par ordinateur, telles que la capture de mouvement et la reconstitution. La tâche de reconstruction est difficile car les visages humains varient considérablement lorsqu'on considère les expressions, les poses, les textures et les géométries intrinsèques. Bien que de nombreuses approches s'attaquent à cette complexité en utilisant des données supplémentaires pour reconstruire le visage d'un seul sujet, l'extraction de la surface faciale d'une seule image reste un problème difficile. En conséquent, Plusieurs approches apparut pour résoudre ces problèmes :

### 5.1. Accurate 3D Face Reconstruction with Weakly-Supervised Learning : From Single Image to Image Set

[36] Propose une nouvelle approche profonde de reconstruction du visage 3D, son principe consiste à tirer parti d'une fonction de perte robuste et hybride pour un apprentissage faiblement supervisé qui prend en compte à la fois l'information de faible niveau et de perception pour la supervision, et également effectue une reconstruction du visage en exploitant des informations complémentaires à partir de différentes images pour l'agrégation de forme. Leurs est rapide, précise, et robuste à l'occlusion et grande Poser.

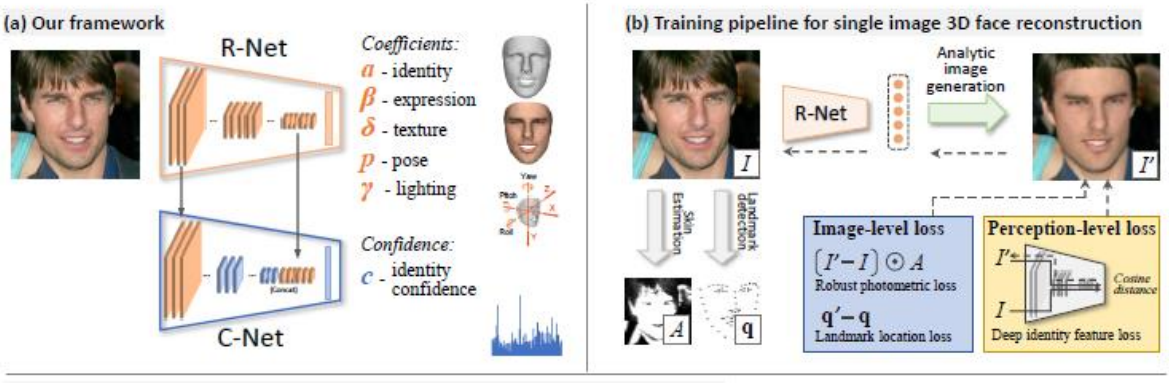



*Figure II.18 Présentation de l'approche utilisée dans [36].*

Des expériences complètes ont montré que leur méthode surpasse les méthodes précédentes par une grande marge en termes de précision et de robustesse. Ils ont également proposé une nouvelle méthode d'agrégation de reconstruction de visage multi-images à l'aide de CNN. Sans étiquette explicite, leur méthode peut apprendre à mesurer la qualité de l'image et à exploiter l'information complémentaire dans différentes images pour reconstruire les visages 3D avec plus de précision.

### 5.2. Face Alignment in Full Pose Range : A 3D Total Solution

[37] Propose l'utilisation d'un nouveau cadre d'alignement appelé 3D Dense Face Alignment (3DDFA), dans lequel le modèle morphable 3D dense (3DMM) est monté sur l'image via Cascaded Convolutional Neural Networks. Ils ont utilisé également des informations 3D pour synthétiser des images faciales dans des vues de profil afin de fournir des échantillons abondants pour la formation. Les expériences menées dans la base de données difficile de l'AFLW montrent que l'approche proposée permet d'apporter des améliorations importantes par le nombre de méthodes de pointe.

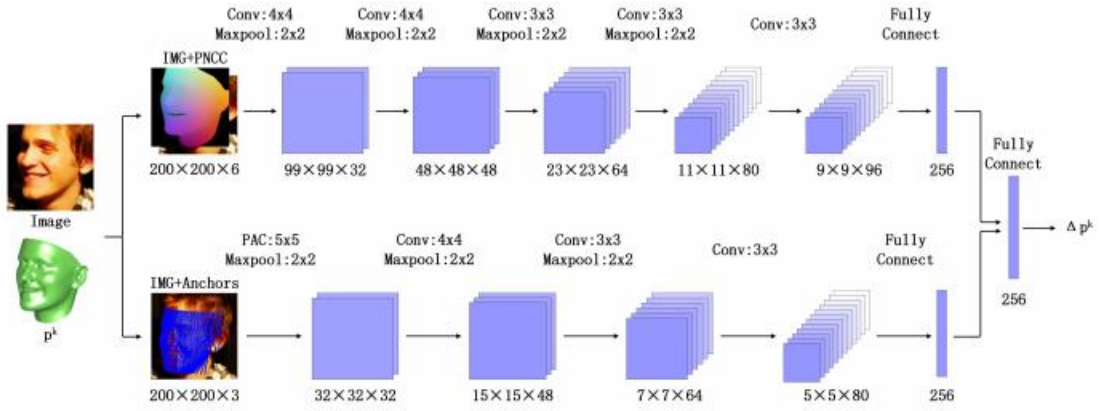



*Figure II.19 Présentation du réseau utilisée dans [37].*

La plupart des méthodes d'alignement du visage ont tendance à échouer dans la vue de profil puisque les repères auto-occlus ne peuvent pas être détectés. Au lieu du cadre de détection traditionnel, Ils ont également proposé une méthode qui s'adapte à un modèle morphable 3D dense pour atteindre l'alignement de visage sans pose.

### 5.3. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network

[38] Propose une méthode simple qui simultanément reconstruit la structure faciale 3D et fournit un alignement dense. À Pour ce faire, Ils ont également conçu une représentation 2D appelée carte de position UV qui enregistre la forme 3D d'un visage complet dans l'espace UV, puis entraîne un simple réseau neuronal à convolution pour le faire régresser à partir d'une seule image 2D. Ils ont intégré également un masque de poids dans la fonction perte pendant l'entraînement pour améliorer les performances du réseau. Leur méthode ne repose pas sur n'importe quel modèle de visage antérieur, et peut reconstruire la géométrie faciale complète le long avec une signification sémantique. Pendant ce temps, leur réseau est très léger et ne passe que 9,8 ms pour traiter une image, ce qui est extrêmement plus rapide que les travaux précédents. Les expériences sur plusieurs ensembles de données difficiles montrent que leur méthode surpasse les autres méthodes de pointe pour les tâches de reconstruction et d'alignement de loin.

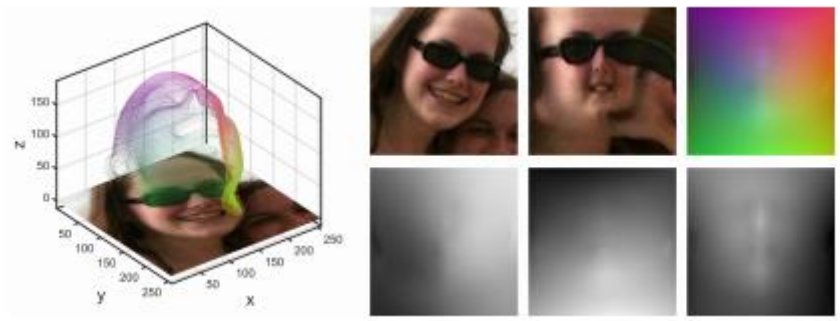



*Figure II.20 Présentation de la carte de position UV. Gauche : 3D représentation de l'image d'entrée et son nuage de points 3D aligné correspondant (comme vérité au sol).*

[38] Ils ont proposé une méthode de bout en bout, qui résout bien les problèmes de l'alignement du visage 3D et de la reconstruction du visage 3D simultanément. En apprenant carte de position, Il régresse directement la structure 3D complète avec la sémantique signification à partir d'une seule image. Les résultats quantitatifs et qualitatifs démontrent que leur méthode est robuste pour les poses, les illuminations et les occlusions. Expériences sur trois jeux de

données de test montrent que cette méthode permet d'obtenir des améliorations par rapport à d'autres. Ils montrent en outre que leur méthode fonctionne plus vite que d'autres méthodes et convient à l'utilisation en temps réel.

### 5.4. Learning Detailed Face Reconstruction from a Single Image

[39] Propose d'augmenter de la puissance des réseaux neuronaux convolutionnels pour produire une reconstruction faciale très détaillée à partir d'une seule image. À cette fin, Ils ont introduit un cadre CNN de bout en bout qui tire la forme d'une manière grossière à fine. L'architecture proposée est composée de deux blocs principaux, un réseau qui récupère la géométrie faciale grossière (GrosseNet), suivie d'un CNN qui affine les traits du visage de cette géométrie (FineNet). Les réseaux proposés sont reliés par une nouvelle couche qui rend une image de profondeur donnée un maillage en 3D. Contrairement aux problèmes de reconnaissance et de détection d'objets, il n'existe pas de jeux de données appropriés pour la formation des CNN afin d'effectuer la reconstruction de la géométrie du visage. Par conséquent, leur régime d'entraînement commence par une phase supervisée, basée sur des images synthétiques, suivie d'une phase non supervisée qui n'utilise que des images faciales non contraintes.

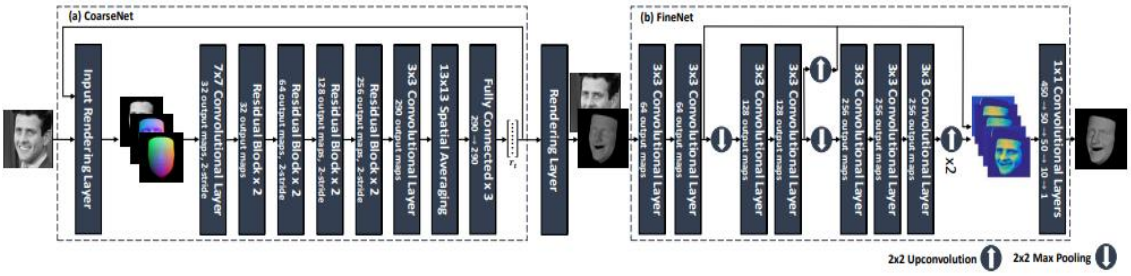



*Figure II.21 Le réseau de bout en bout, composé de GrosseNet, FineNet et de la couche intermédiaire.*

[39] Ils ont proposé une approche de bout en bout pour la reconstruction détaillée du visage à partir d'une seule image. Leur méthode est composée de deux blocs principaux, un réseau pour récupérer une estimation approximative de la géométrie du visage suivie d'un réseau de reconstruction de détails fins. Alors que le premier est formé avec des images synthétiques, le second est formé avec de vraies images faciales dans un programme de formation de bout en bout non supervisée. Pour connecter les deux réseaux, une couche différente est introduite. Par leur démonstration et comparaisons, cette méthode surpasse les approches récentes à la fine pointe de la technologie.

### 5.5. Self-Supervised Monocular 3D Face Reconstruction by Occlusion-Aware Multi-view Geometry Consistency

[40] Propose une architecture de formation auto-supervisée en tirant parti de la cohérence de géométrie multi-vue, qui fournit des contraintes fiables sur la pose du visage et l'estimation de la profondeur. Ils ont également proposé une méthode de synthèse de vue d'occlusion-consciente pour appliquer la cohérence multi-vue de géométrie à l'apprentissage auto-supervisé. Ensuite, Ils ont conçu trois nouvelles fonctions de perte pour la cohérence multi-vue, y compris la perte de cohérence des pixels, la perte de cohérence de profondeur, et la perte épipolaire des points de repère du visage. Leur méthode est précise et robuste, en particulier dans de grandes variations d'expressions, poses, et les conditions d'éclairage. Des expériences complètes sur l'alignement du visage et les repères de reconstruction du visage 3D ont démontrés une supériorité sur les méthodes de pointe.

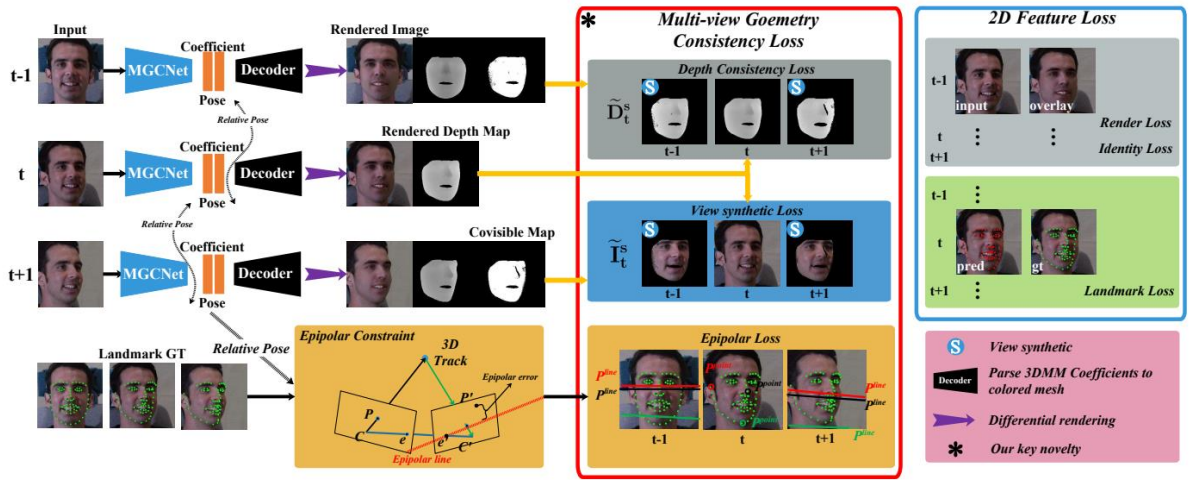



*Figure II.0 Le flux de formation de l'architecture MGCNet.*

Ils ont également présenté un pipeline auto-supervisé MGCNet pour la reconstruction monoculaire 3D Face et ont démontré les avantages de l'exploitation de la cohérence de géométrie multi-vue pour fournir une contrainte plus fiable sur la pose du visage et l'estimation de la profondeur. Ils mettent l'accent sur la synthèse de vue occlusion-consciente et les pertes multi-vues pour rendre le résultat plus robuste et fiable. Leur MGCNet révèle profondément la capacité de la cohérence de géométrie multi-vue avec un apprentissage auto-supervisé dans la capture à la fois des indices de haut niveau et des correspondances de fonctionnalités avec le raisonnement de géométrie. Les résultats par rapport à d'autres méthodes indiquent que leur MGCNet peut atteindre le résultat exceptionnel sans données étiquetées coûteuses.

### 5.6. Learning Robust 3D Face Reconstruction and Discriminative Identity Representation

[41] Propose une solution robuste pour résoudre le problème de la reconstruction 3D en concevant soigneusement le nouveau réseau neuronal convolutionnel siamois « Siamise » (SCNN).

Plus précisément, en ce qui concerne les paramètres du 3D Morphable face Model (3DMM) du même individu dans la même classe, ils ont utilisé la perte contrastée pour agrandir la distance interclasse et ont réduit la distance intra-classe pour les paramètres 3DMM de sortie. Ils ont également proposé une perte d'identité pour préserver les informations d'identité pour le même individu dans l'espace de fonctionnalité. Train avec ces deux pertes, Leur SCNN pourrait apprendre des représentations qui sont plus discriminatoires pour l'identité du visage et généralisable pour les variantes de pose. Des expériences sur la base de données 300W-LP et AFLW2000-3D ont montré l'efficacité de leur méthode en comparant avec l'état des arts.

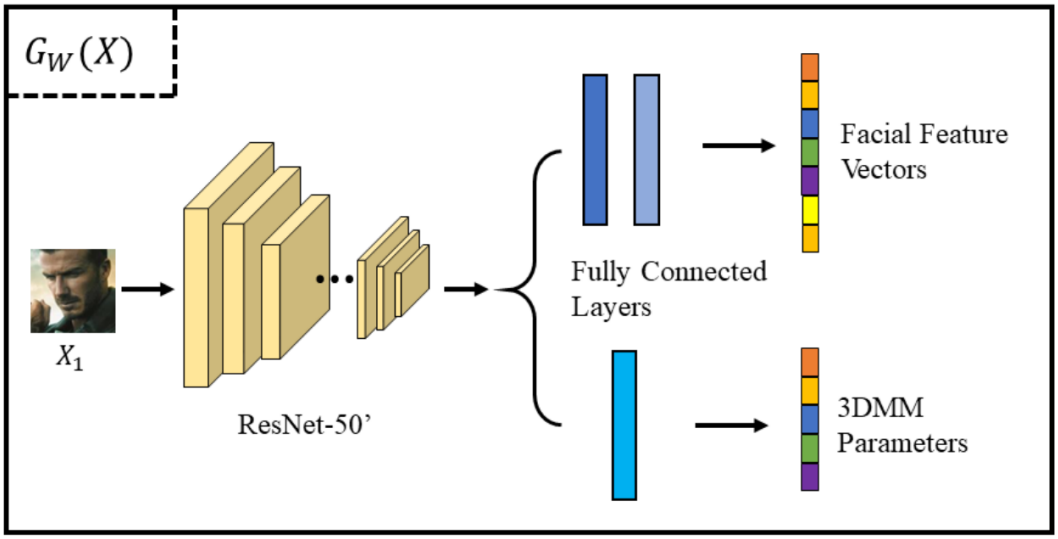



*Figure II.23 L'architecture du réseau utilisé.*

Ils ont utilisé la nouvelle méthode siamois (SCNN) pour la reconstruction robuste du visage 3D et l'identité discriminative. À cette fin, ils ont introduit trois pertes pour assurer l'exactitude et la robustesse de la reconstruction du visage, tout en préservant l'information d'identité de l'image d'entrée. Les expériences sur les deux jeux de données faciales illustrent l'efficacité du SCNN sur la reconstruction faciale 3D et la reconnaissance faciale 2D en comparant avec d'autres état des arts.

## 6. Conclusion

Il existe plusieurs autres méthodes basées sur l'apprentissage automatique qui sont capables d'effectuer une reconstruction du visage 3D et qui sont fondées sur une seule entrée (l'image). Chaque méthode a ses propres avantages, Ces dernières sont différentes les unes des autres en termes de : temps nécessaire pour générer le résultat visé, la plate-forme sur laquelle la reconstruction est exécutée…

Dans le chapitre suivant, nous représenterons notre travail et nous allons l'expliquer étape par étape.

# Chapitre III :
# Implémentation

# 1. Introduction

Dans ce chapitre, nous allons montrer comment produire une reconstruction 3D du visage à partir d'une seule image 2D. En effet l'architecture de notre méthode est composée de trois étapes, La première étape consiste à effectuer un apprentissage sur une base de données que nous avons créé à l'aide d'un réseau de neurone convolutif dont l'entrée est une image 2D et la sortie est un vecteur comportant les Landmarks dite aussi les points de repères, la seconde étape consiste à générer la géométrie et le mesh correspondant à nos points de repères, ensuite la dernière étape dont la tâche est le plaquage de texture.

# 2. Description des Activités de Notre Modèle

Nous allons voir comment nous avons créé le jeu de données utilisé plus tard dans le réseau, ensuite nous allons voir le réseau CNN utilisé.

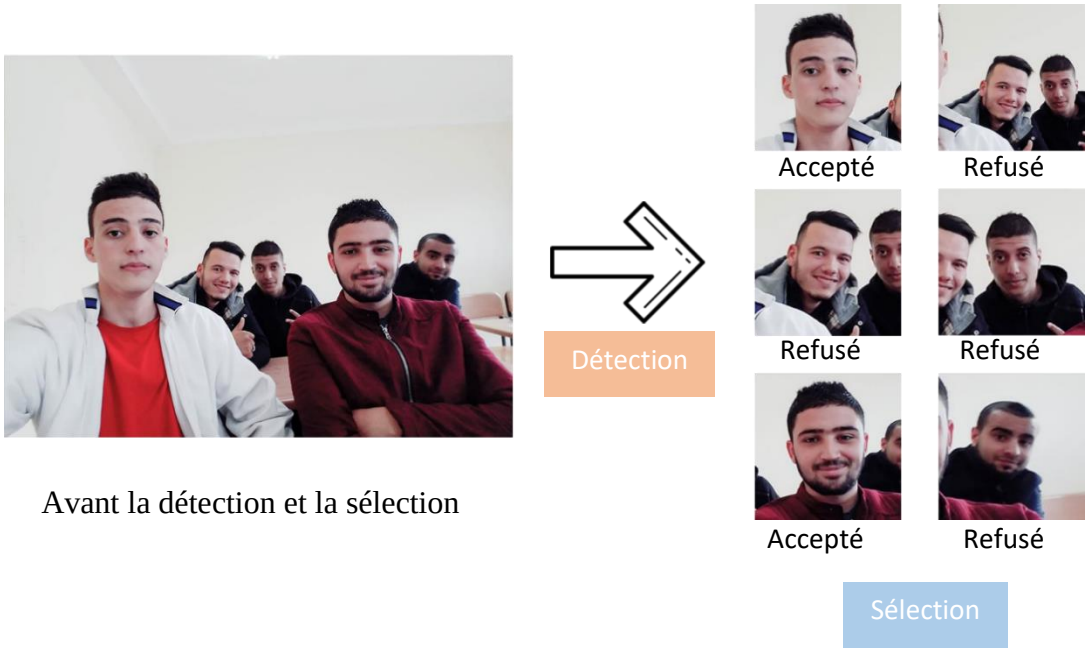
## 2.1.La Création du Jeu de Données

La mise en place d'un ensemble ou jeu de données d'images n'est pas facile, le jeu de données utilisée affectera le résultat d'une manière directe, de toute façon nous devons collecter et sélectionner les données basées sur de nombreuses caractéristiques.

### 2.1.1. La Collection des données

Tout d'abord, nous avons recueilli des images de différentes sources « Kaggle [42], ThisPersondoesnotexist [43], HELEN DATASET [44] » qui contient différentes positions du visage humains, environ 30 mille images ont été recueillies à partir de ces sources, Puis, nous avons utilisé le classificateur de Haar Cascade pour trouver et détecter les visages humains puis les redimensionner en (224x224), après cela, nous sommes passés dans une autre phase, nous avons sélectionné manuellement les images avec les meilleurs résultats basés sur lesquelles sera mieux pour notre reconstruction 3D. Enfin, le total des données recueillies est d'environ 15 000 image.

La Figure III.1 montre les différentes étapes du détection et sélection.




Avant la détection et la sélection

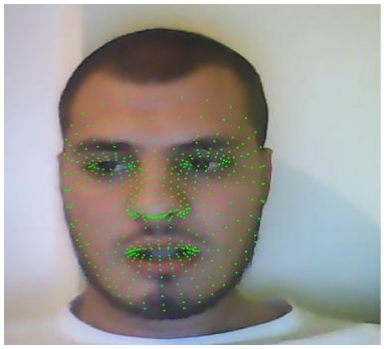
*Figure III.1 Exemple des étapes pour la détection et la sélection du jeu de de données.*

### 2.1.2. Détection Du Visage 2D

Dans cette étape, Nous avons utilisé la bibliothèque Facemesh [45] pour détecter les points de repère du visage humain, Les points utilisés ont une grande variabilité et une grande importance dans la perception du visage humain, Il permet de construire une représentation de surface lisse plausible selon la subdivision Catmull Clark [46], nous avons utilisé les cordonnées (x,y) et la triangulation de cette bibliothèque, après nous avons faire une sélection et une filtration pour les résultats obtenu parce qu'ils n'étaient pas précis.

La topologie utilisée dans notre travail est composée de 468 points (x,y,z) et de 880 triangulation. Dans l'étape suivante, nous allons expliquer comment nous avons calculé la coordonnées « z » dit aussi la profondeur pour chacun des points obtenus.

Les Points Sans Triangulation



Les Points Avec Triangulation

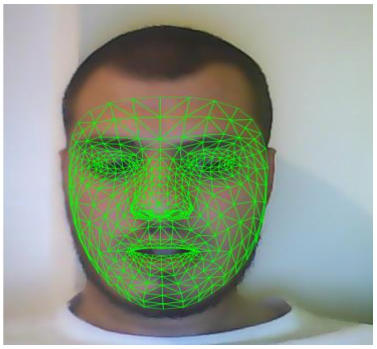


*Figure III.2 Résultat de la détection.*

### 2.1.3. Profondeur des Points

Dans cette étape nous allons calculer la profondeur (depth) ou la cordonnée « z » pour chacun des points obtenus à partir l'étape précédente, jusqu'à 87% des images avaient des prédictions appropriées pour l'étape suivante en effet 13K images de bons résultats ont été sélectionnées.

### 2.1.4. La Sélection des Objets 3D

A partir les résultats des étapes précédentes, et à l'aide d'un algorithme que nous avons écrit, nous avons créé pour chaque image une géométrie 3D (objet 3D) correspondante. En utilisant les coordonnées x, y et z des points de repère, la triangulation et le plaquage de texture est présenté dans un espace vectoriel UV, ensuite nous effectuons une autre phase de sélection manuelle afin de garder seulement les objets (mesh) en bonne état en termes de précision et sans défaut. Les données englobées étaient d'environ 8K images qui nous donne un bon résultat.


*Figure III.3 Exemple de la phase de sélection des géométries 3D*

Chaque coordonnée a été enregistrée dans une format qui peut être correctement représenté sur L'échelle du texture (espace image) où chaque coordonnée aura une valeur entre 0 et 1, cependant pour passer à une représentation 3D nous devons effectuer une translation entre l'espace 2D et l'espace 3D.

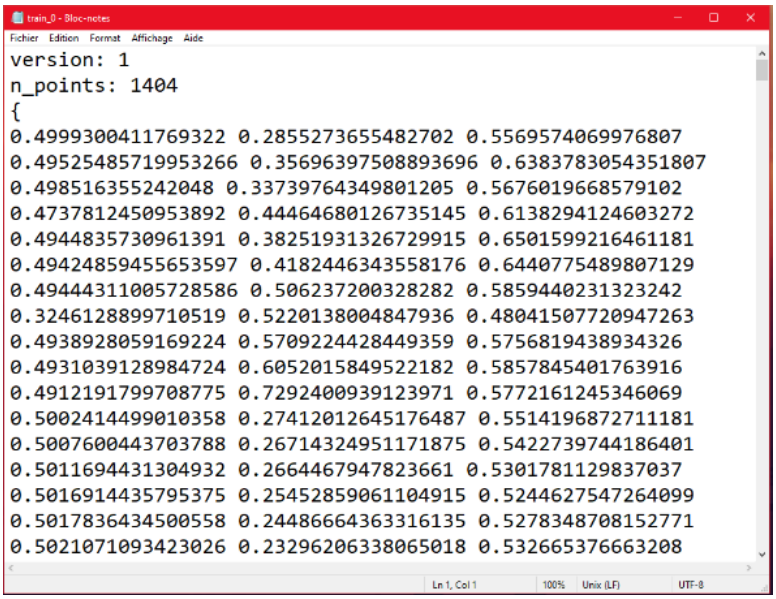



*Figure III.4 Les cordonnées 'landmarks'*

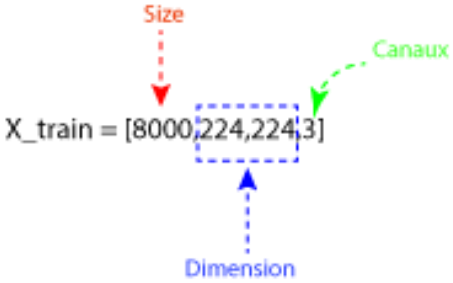
## 2.2.Architecture de CNN

Les réseaux de neurones convolutifs ou CNN repose sur une architecture à base de couches, celles-ci vont subir une succession de différents traitements afin d'extraire les caractéristiques les plus importantes de l'image. Au fur et à mesure que l'on traverse le réseau, les couches seront compressées, chaque couche de convolution est suivie d'une couche d'activation (RELU) ensuite ces caractéristiques seront transmises à un réseau de neurone intégralement connecté afin d'effectuer une phase de reconnaissance.

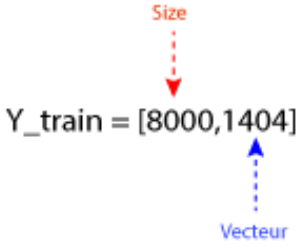
### 2.2.1. Prétraitement du jeu de données

Cette étape est primordiale pour améliorer les performances de notre modèle, elle est effectuée avant l'étape d'apprentissage. Dans notre modèle nous avons tous d'abord appliqué deux opérations de prétraitement :

- On a transformer le dossier d'images (jeu de données) en tableau de 4 dimensions :




- On a transformer le dossier des Landmarks (les points de repère) en tableau de 2 dimensions




Les vecteurs 1404 : ce sont présenter comme suite :

$$[x_0,x_1,x_2...x_{468},y_0,y_1,y_2...y_{468},z_0,z_1,z_2...z_{468}]$$

Après ces prétraitements on a destiné 75% des images (6000 instances) pour l'entrainement (train) et 25% (2000 instances) pour la phase d'essai (test).

### 2.2.2. Apprentissage

L'architecture de réseau de neurones convolutifs utilisée dans notre modèle est composée de six couches de convolution, et cinq couches de maxpooling. La Fig. IV.5 ci-dessous représente l'architecture CNN.

```python
def get_my_CNN_model_architecture():

    model = Sequential()
    model.add(Conv2D(16, (3, 3), input_shape=(224, 224, 3), kernel_initializer='random_uniform', activation='relu'))

    model.add(Conv2D(32, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.3))

    model.add(Conv2D(256, (5, 5), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(512, (5, 5), activation='relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.5))

    model.add(Flatten())

    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.7))

    model.add(Dense(1404))

    return model
```

*Figure III.5 code source du CNN utilisé.*

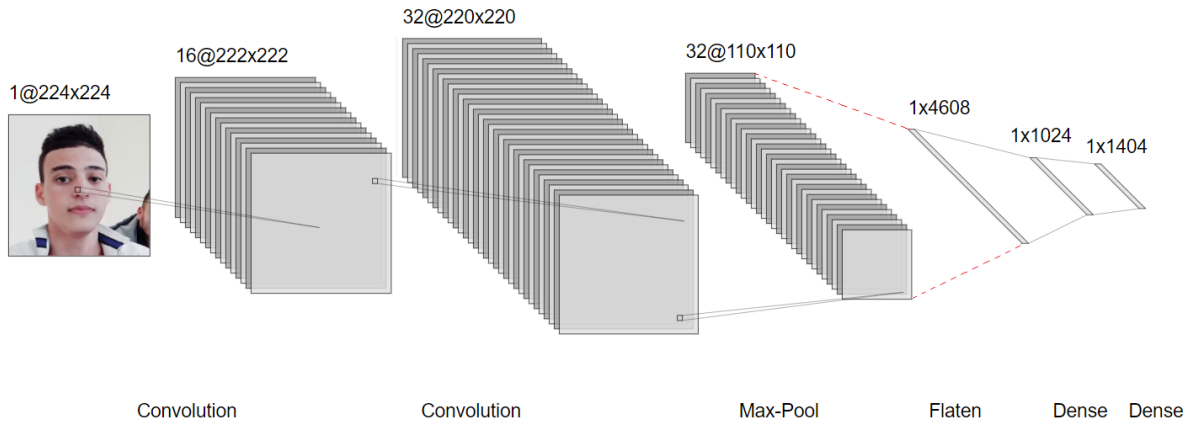



*Figure III.6 L'architecture CNN utilisée.*

***La couche de convolution C1 :*** après le prétraitement de l'image, la première couche de convolution C1 est paramétrée par la taille, le nombre de cartes, les tailles de noyau et la table de connexion.

- ❖ La couche de convolution adopte 16 noyaux de convolution (kernels).
- ❖ La taille de chaque noyau de convolution est 3x3.
- ❖ Il peut produire six cartes de caractéristiques (feature map), chaque carte de caractéristiques contient 49284 neurones c'est-à-dire (224-3+1)*(224-3+1)=222*222.

***La couche de convolution C2 :*** la deuxième couche de convolution 2 est paramétrée par la taille, le nombre de cartes, les tailles de noyau et la table de connexion.

- La couche de convolution adopte 32 noyaux de convolution (kernels).
- La taille de chaque noyau de convolution est 3x3.

- Il peut produire six cartes de caractéristiques (feature map), chaque carte de caractéristiques contient 48400 neurones c'est-à-dire (222-3+1)*(222-3+1)=220*220.

***La couche de Max Pooling P1 :*** Une couche de Max Pooling est utilisée pour réduire la taille des images et pour obtenir une convergence plus rapide pendant l'entraînement.

- Elle contient 32 cartes de caractéristiques (feature map).
- Chaque carte de caractéristiques contient 12100 neurones c'est-à-dire (110*110).
- La fenêtre de maxpooling est une matrice de 2*2 dimensions.

Les couches de convolution C3, C4, C5, C6 et les couches de maxpooling P2, P3, P4, P5 sont exécutées de la même façon.

***La couche Dropout :*** cette couche définit de manière aléatoire les unités d'entrée sur 0 avec une fréquence d'occurrence à chaque étape pendant le temps d'entraînement, ce qui permet d'éviter le surajustement. Les entrées non réglées sur 0 sont mises à l'échelle de 1 / (taux 1) de sorte que la somme de toutes les entrées reste inchangée.

- ***La fonction BatchNormalization :*** Cette fonction est utilisée pour Normaliser et mettre à l'échelle les entrées de la couche précédente, c'est-à-dire appliquer une transformation qui maintient l'activation moyenne proche de 0 et l'écart type d'activation proche de 1.
- ***La fonction Flatten :*** Le principe de cette étape consiste à aplatir toutes les cartes de caractéristiques de l'étape précédentes, en d'autres termes cela consiste à aplatir la matrice reçue en entrée en un vecteur colonne de dimension 1 afin de reconstruire la couche d'entrée du réseau de neurones complètement connecté.

***La couche complètement connectée (full connected) :*** Cette couche est composée d'un réseau de neurones complètement connectée, dont l'entrée est un vecteur colonne constituée des caractéristique pertinentes, ainsi un nombre de couches cachées et une fonction d'activation pour prédire la valeur finale.

Le tableau ci-dessous résume les étapes de construction du réseau convolutifs

| **Layer** | **Description** | **Input** | **Output** |
|---|---|---|---|
| 1 | Conv (3x3) RELU activation | 224x224x3 | 222x222x16 |
| 2 | Conv (3x3) RELU activation | 222x222x16 | 220x220x32 |
| 3 | Max_Pool(2x2, stride 2) | 220x220x32 | 110x110x32 |
| 4 | Conv (3x3) RELU activation | 110x110x32 | 108x108x64 |
| 5 | Max_Pool(2x2, stride 2) | 108x108x64 | 54x54x64 |
| 6 | Conv (3x3) RELU activation | 54x54x64 | 52x52x128 |

| | | | |
|---|---|---|---|
| 7 | BatchNormalization() | 52x52x128 | 52x52x128 |
| 8 | Max_Pool(2x2, stride 2) | 52x52x128 | 26x26x128 |
| 9 | DropOut() | 26x26x128 | 26x26x128 |
| 10 | Conv (3x3) RELU activation | 26x26x128 | 22x22x256 |
| 11 | Max_Pool(2x2, stride 2) | 22x22x256 | 11x11x256 |
| 12 | Conv (3x3) RELU activation | 11x11x256 | 7x7x512 |
| 13 | BatchNormalization() | 7x7x512 | 7x7x512 |
| 14 | Max_Pool(2x2, stride 2) | 7x7x512 | 3x3x512 |
| 15 | DropOut () | 3x3x512 | 3x3x512 |
| 16 | Flatten() | 3x3x512 | 4608 |
| 17 | Dense() | 4608 | 1024 |
| 18 | BatchNormalization() | 1024 | 1024 |
| 19 | DropOut() | 1024 | 1024 |
| 20 | Dense() | 1024 | 1404 |

*Tableau III.1 Description de CNN utilisé*

### 2.3.La Reconstruction 3D

Après avoir obtenu le résultat de l'apprentissage profond, et après avoir effectué certaines opérations sur le modèle pour une meilleure performance, nous allons utiliser ce modèle pour nous donner les landmarks (points de repères), puis on va créer la géométrie 3D.

#### 2.3.1. La Création de la géométrie 3D

Les points 3D du visages générés à partir d'un modèle sont représentés par leurs coordonnées définies dans l'espace image 2D (texture). Puis nous calculons les coordonnées de la forme 3D à l'aide des fonctions mathématiques :

$$F(x) = (x - centreX) * 224$$

$$G(y) = (y - centreY) * 224$$

$$H(z) = (z * 200) - 100$$

- x,y,z sont les cordonnée de prédiction.
- centreX, centreY sont les point prediction [195] , prediction [663].

Les coordonnées des 468 points sont regroupés et représentés sur la scène, puis avec l'utilisation de la triangulation, nous représentons la forme géométrique 3D du visage sans le plaquage de la texture.


Entrée (Image 2D)

La Prédiction

La Reconstruction 3D

*Figure III.7 Exemple de la reconstruction 3D à partir d'une image 2D (sans Texture).*

### 2.3.2. Le Plaquage de texture

Dans cette étape, chaque face de l'objet obtenu sera texturée avec le mappage UV correspondant que nous avons déjà obtenu avec notre modèle.

*Figure III.8 Exemple de la reconstruction 3D à partir d'une image 2D (Avec Texture).*

## 3. Conclusion

Dans ce chapitre, nous avons expliqué notre travail, le modèle que nous avons obtenu aprés ces étapes est facile à utiliser et simple de le mettre en œuvre.

Dans le chapitre suivant il est temps d'évaluer les résultats obtenus et de mesurer les performances de notre modèle prédictif avec la démonstration de quelques exemples.

# Chapitre IV :
# Résultat et Validation

## 1. Introduction

Une fois notre modèle construit, il est temps d'évaluer les résultats obtenus et de mesurer les performances de notre modèle prédictif. C'est à ce moment que le jeu de test est crucial. Les prédictions se font sur cette dernière, c'est une nouvelle donnée pour laquelle la prédiction du visage est inconnue.

## 2. La Configuration du Matériel Utilisé

Afin d'exécuter l'apprentissage de ce projet, nous avons utilisé un GPU cloud (Lambda Labs) [47] dont les principales caractéristiques sont les suivantes :

- RAM : 32 GO.
- Système d'exploitation : Ubuntu Server 18.04 LTS.
- Carte Graphique (GPU) : 4x NVIDIA® Pascal Based GPUs (11 GB).

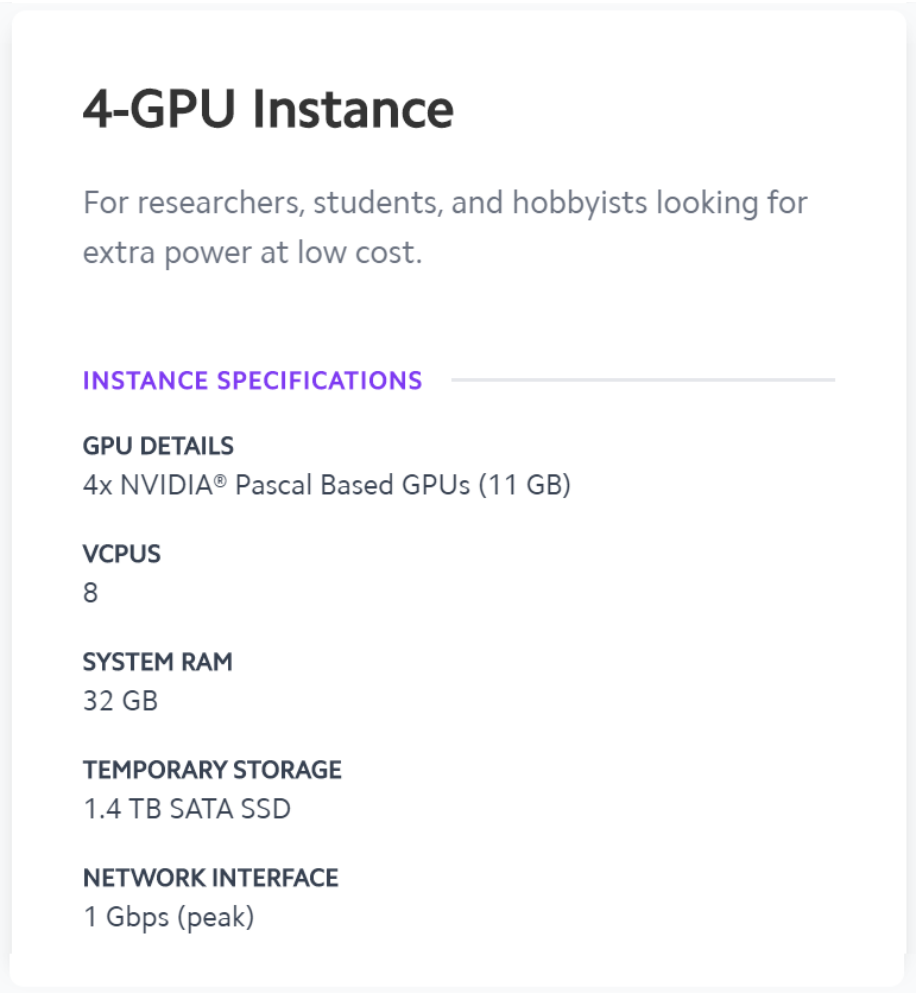



*Figure IV.1 L'instance utilisé du LambdaLabs GPU cloud.*

## 3. L'environnement de Travail

### 3.1.Python

Python est un langage de programmation qui est facile à comprendre et très utile, Il contient plusieurs structures de données et utilise aussi la programmation orientée objet, Python est l'une des meilleurs langages pour le script et le développement rapide des applications dans de nombreux domaines et sur la plupart des plates-formes. [48]

### 3.2.JavaScript

JavaScript est un langage de programmation qui est utilisé dans presque tous les coins du Web si ce n'est pas tous, il permet de nombreuses fonctionnalités d'être présentés sur une page Web (afficher le contenu mis à jour à des moments spécifiés, des cartes interactives, animations 2D/3D, défilement des menus vidéo, etc.). JS est également appelé La troisième couche de technologies web standard que nous utilisons encore aujourd'hui. [49]

### 3.3.NodeJS

Node.js est un environnement de serveur open-source basé sur le moteur JavaScript V8 de Chrome, il vous permet d'exécuter JavaScript sur le côté serveur. [50]

### 3.4.Tensorflow

TensorFlow est une bibliothèque open-source qui existe dans de nombreux langages de programmation tels que Python, Javascript et C++, il a été initialement lancé par l'équipe de recherche de Google artificielle intelligence pour mener des recherches sur l'apprentissage automatique et les réseaux neuronaux profonds. TF peut également être utilisé dans une grande variété d'autres domaines. [51]

### 3.5.Keras

Keras est une API d'apprentissage profond écrite en Python, fonctionnant sur la plate-forme d'apprentissage automatique TensorFlow. Il a été développé dans le but de permettre une expérimentation rapide. Être capable de passer d'une idée à un résultat aussi vite que possible.[52]

### 3.6.THREE JS

Three.js est une bibliothèque open-source en JavaScript, il est utilisé pour créer des scènes et des infographies 3D dans un navigateur web. Elle peut être utilisée à l'aide du balise canvas du HTML5, l'objectif de cette bibliothèque est de mettre tout le monde accessible pour avoir des infographie 3D dans le web, il utilise WebGL, CSS3D et SVG. [53]

# 4. Résultats

## 4.1.Le Jeu de Données

Le jeu de données a été créé à partir des images de différentes sources « Kaggle [54], ThisPersondoesnotexist [55], HELEN DATASET [56] ». Le total que nous avons collecté était d'environ 30 000 instances. La taille des images varie entre 1024×1024 et 224×224 pixels.

L'ensemble de données publié dans Kaggle se compose de 70 000 images PNG de haute qualité à la résolution 1024×1024 et contient des variations considérables en termes d'âge, d'origine ethnique et d'origine image. Les images ont été rampées à partir de Flickr, héritant ainsi de tous les biais de ce site, et automatiquement alignés et recadrés en utilisant dlib. Seules les images sous licence permissive ont été collectées. Divers filtres automatiques ont été utilisés pour mettre à l'échelle l'ensemble des images, et finalement Amazon Mechanical Turk a été utilisé pour enlever les statues occasionnelles.

La Figure IV.2 montre des exemples du jeu de données.

*Figure IV.2 échantillon d'image du Kaggle.*

*ThisPersondoesnotexist* est un site web qui génère des photos aléatoires pour des personnes qui n'éxiste pas, il utilisent le moteur GAN (generative adversarial network). La Figure IV.3. montre des exemples générés par le Moteur GAN.

*Figure IV.3 Exemples d'image du ThisPersondoesnotexist.*

L'ensemble de données de HELEN se compose de 2330 images au total, ce sont précises et détaillées. Un échantillonnage de l'ensemble de données est montré dans la Figure IV.4.

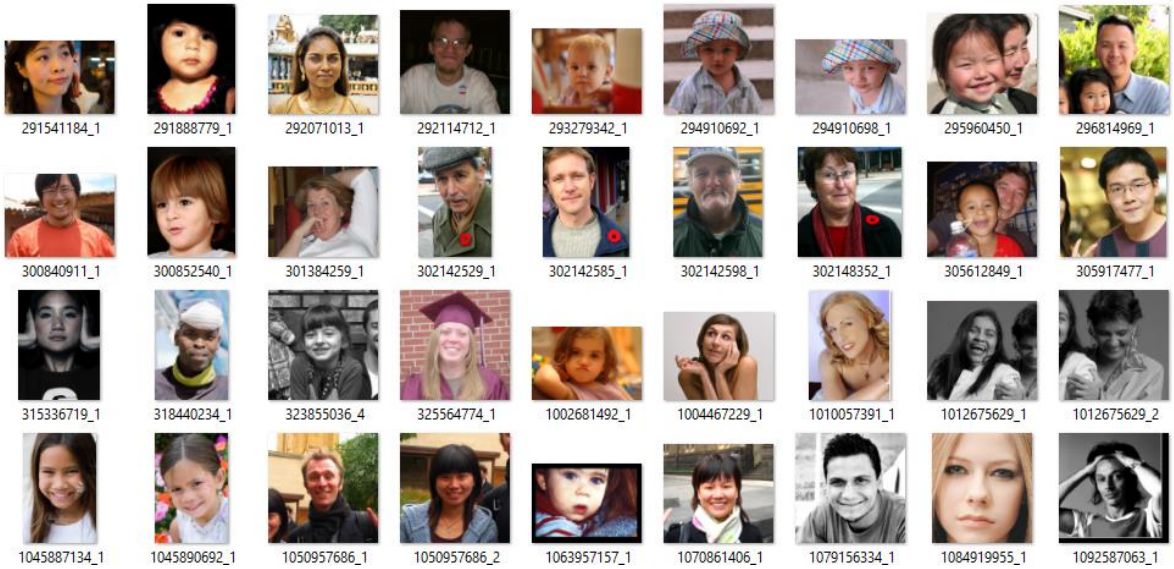



*Figure IV.4 échantillon du jeu des images du HELEN.*

Après avoir recueilli d'environ de 30 000 images à partir de ces sources et après des traitements sur cet ensemble de données (*voir Chapitre III*). Un total de 8000 instances d'image de taille 244x244 pixels en RGB avec un format JPG. La Figure IV.5 montre un échantillon du jeu de données finalisé.


*Figure IV.5 échantillon du jeu de données généré.*

### 4.2.La Phase de L'apprentissage

Dans cette étape en utilisant le jeu de données résultant, et à l'aide d'un réseau CNN. Dans Les figures V.6 (a) et V.6 (b) l'apprentissage de notre réseau de neurones sur le cloud à travers 600 epochs.

```
Epoch 242/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.6125e-04 - acc: 0.6009 - val_loss: 1.1296e-04 - val_acc: 0.6919
Epoch 243/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.5533e-04 - acc: 0.5930 - val_loss: 7.7221e-04 - val_acc: 0.6956
Epoch 244/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.5076e-04 - acc: 0.6047 - val_loss: 9.2137e-05 - val_acc: 0.7063
Epoch 245/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4509e-04 - acc: 0.6028 - val_loss: 9.2708e-05 - val_acc: 0.7106
Epoch 246/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4554e-04 - acc: 0.6052 - val_loss: 0.0030 - val_acc: 0.7250
Epoch 247/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4463e-04 - acc: 0.6164 - val_loss: 9.5949e-05 - val_acc: 0.7169
Epoch 248/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.5707e-04 - acc: 0.6073 - val_loss: 0.0024 - val_acc: 0.6475
Epoch 249/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4709e-04 - acc: 0.6059 - val_loss: 8.5471e-05 - val_acc: 0.7206
Epoch 250/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4595e-04 - acc: 0.6175 - val_loss: 9.5921e-05 - val_acc: 0.7100
Epoch 251/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4708e-04 - acc: 0.6116 - val_loss: 0.7666 - val_acc: 0.5469
Epoch 252/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.6462e-04 - acc: 0.5973 - val_loss: 0.0941 - val_acc: 0.6919
Epoch 253/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4483e-04 - acc: 0.6009 - val_loss: 1.0769e-04 - val_acc: 0.6863
Epoch 254/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4946e-04 - acc: 0.6162 - val_loss: 1.0168e-04 - val_acc: 0.7144
Epoch 255/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4844e-04 - acc: 0.6003 - val_loss: 0.4013 - val_acc: 0.6781
Epoch 256/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.7159e-04 - acc: 0.5853 - val_loss: 0.0012 - val_acc: 0.6600
Epoch 257/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4823e-04 - acc: 0.6077 - val_loss: 1.0529e-04 - val_acc: 0.6869
Epoch 258/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4573e-04 - acc: 0.6078 - val_loss: 0.0244 - val_acc: 0.7181
Epoch 259/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4529e-04 - acc: 0.6081 - val_loss: 7.6593e-04 - val_acc: 0.7100
Epoch 260/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4425e-04 - acc: 0.6155 - val_loss: 2.5216e-04 - val_acc: 0.7075
Epoch 261/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.4267e-04 - acc: 0.6142 - val_loss: 9.6763e-05 - val_acc: 0.7175
Epoch 262/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.3804e-04 - acc: 0.6133 - val_loss: 9.1399e-05 - val_acc: 0.7200
Epoch 263/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.3707e-04 - acc: 0.6152 - val_loss: 8.9815e-05 - val_acc: 0.7094
```

(a)

```
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0571e-04 - acc: 0.6664 - val_loss: 9.4769e-05 - val_acc: 0.7494
Epoch 563/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0112e-04 - acc: 0.6669 - val_loss: 8.7174e-05 - val_acc: 0.6869
Epoch 564/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0242e-04 - acc: 0.6666 - val_loss: 8.1077e-05 - val_acc: 0.7319
Epoch 565/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0025e-04 - acc: 0.6664 - val_loss: 7.3923e-05 - val_acc: 0.7425
Epoch 566/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0408e-04 - acc: 0.6762 - val_loss: 8.9476e-05 - val_acc: 0.7306
Epoch 567/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0037e-04 - acc: 0.6584 - val_loss: 8.3766e-05 - val_acc: 0.7575
Epoch 568/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0360e-04 - acc: 0.6661 - val_loss: 8.2916e-05 - val_acc: 0.7256
Epoch 569/600
6400/6400 [==============================] - 19s 3ms/step - loss: 9.8930e-05 - acc: 0.6719 - val_loss: 8.3574e-05 - val_acc: 0.7388
Epoch 570/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0194e-04 - acc: 0.6750 - val_loss: 8.5529e-05 - val_acc: 0.7137
Epoch 571/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0509e-04 - acc: 0.6622 - val_loss: 7.7564e-05 - val_acc: 0.7475
Epoch 572/600
6400/6400 [==============================] - 19s 3ms/step - loss: 9.9766e-05 - acc: 0.6663 - val_loss: 8.3150e-05 - val_acc: 0.7488
Epoch 573/600
6400/6400 [==============================] - 19s 3ms/step - loss: 9.9577e-05 - acc: 0.6680 - val_loss: 9.8814e-05 - val_acc: 0.7169
Epoch 574/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0955e-04 - acc: 0.6572 - val_loss: 9.7004e-05 - val_acc: 0.6663
Epoch 575/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0333e-04 - acc: 0.6602 - val_loss: 8.3991e-05 - val_acc: 0.7569
Epoch 576/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0922e-04 - acc: 0.6500 - val_loss: 9.6099e-05 - val_acc: 0.7281
Epoch 577/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0685e-04 - acc: 0.6617 - val_loss: 8.5156e-05 - val_acc: 0.7488
Epoch 578/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0318e-04 - acc: 0.6495 - val_loss: 7.7470e-05 - val_acc: 0.7469
Epoch 579/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0280e-04 - acc: 0.6677 - val_loss: 7.8665e-05 - val_acc: 0.7512
Epoch 580/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0295e-04 - acc: 0.6623 - val_loss: 7.8815e-05 - val_acc: 0.7569
Epoch 581/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0125e-04 - acc: 0.6680 - val_loss: 7.3221e-05 - val_acc: 0.7494
```

(b)

*Figure IV.6 (a), (b) l'étape de l'apprentissage de notre CNN.*

La Figure ci dessous montre la précision final, architecture ainsi l'ensemble des paramètres du CNN.

```
Epoch 600/600
6400/6400 [==============================] - 19s 3ms/step - loss: 1.0209e-04 - acc: 0.6739 - val_loss: 7.3169e-05 - val_acc: 0.7538

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 222, 222, 16)      448
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 220, 220, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 110, 110, 32)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 108, 108, 64)      18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
batch_normalization_1 (Batch (None, 52, 52, 128)       512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 22, 22, 256)       819456
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 11, 11, 256)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 7, 7, 512)         3277312
_________________________________________________________________
batch_normalization_2 (Batch (None, 7, 7, 512)         2048
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 3, 3, 512)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 3, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              4719616
_________________________________________________________________
batch_normalization_3 (Batch (None, 1024)              4096
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1404)              1439100
=================================================================
Total params: 10,359,580
Trainable params: 10,356,252
Non-trainable params: 3,328
_________________________________________________________________
ubuntu@96-76-203-56:~$
ubuntu@96-76-203-56:~$
```

*Figure IV.7 précision et architecture du CNN*

### 4.3.Résultat de reconstruction 3D

Après l'exécution de l'apprentissage, nous avons obtenu notre module d'apprentissage sous un format h5, nous l'avons transformé en format « JSON » avec quelques fichiers de format « bin » cela afin d'avoir un traitement plus rapide sur le web.

Le Tableaux IV.1 montre quelques résultats obtenus par notre méthode, la première colonne représente les images d'entrées de notre CNN, la deuxième colonne représente les sommets de notre mesh puis la troisième montre la forme géométrique, cette dernière est le résultat de l'étape de triangulation, et finalement la dernière colonne qui représente le visage texturé obtenu par notre méthode.

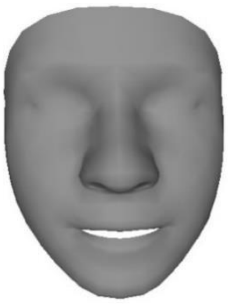
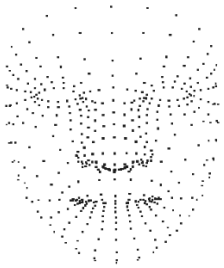
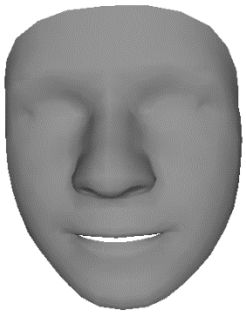
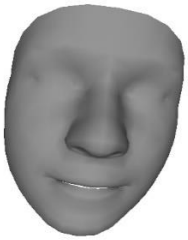
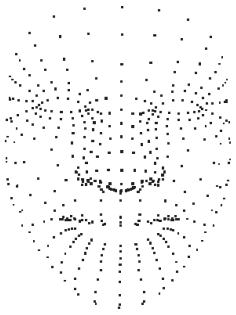
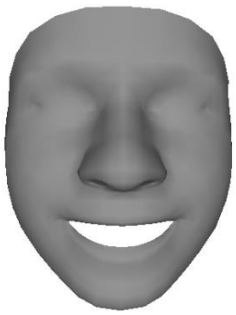
| Input | Sommet | Mesh | Résultat avec texture |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*Tableau. IV.1 Exemples de notre Modèle.*

## 5. Validation

Pour démontrer l'efficacité de notre approche, nous allons comparer notre solution avec deux méthodes récentes [36][37] en effectuant une évaluation quantitative et qualitative en comparant la géométrie obtenue par notre méthode avec celles dites de références (florence [57], AFLW2000-3D [37])

Nous avons utilisé la distance de Hausdorff [58] comme métrique de comparaison, dans cette métrique nous calculons l'erreur comme étant la distance entre l'ensemble des points entre deux géométries, celle dite de référence (ground truth) et la géométrie générée par notre méthode, la distance Hausdorff est calculée par la formule suivante :

$$d_H(X,Y) = \max\{ \sup_{x \in X} \inf_{y \in Y} d(x,y) , \sup_{y \in Y} \inf_{x \in X} d(x,y) \}$$

### 5.1.La Comparaison Quantitative

Le tableau IV.2 montre une comparaison entre notre méthode et les deux méthodes de l'état de l'art [36][37] sur deux jeux de données (dataset) différents utilisés comme objet de référence.

| **Méthodes** | **Florence** | **AFLW2000-3D** |
|---|---|---|
| [36] | RMS : 0.107337 | / |
| [37] | RMS : 0.312927 | RMS : 0.161662 |
| Notre Modelé | RMS : 0.054579 | RMS : 0.380293 |

*Tableau. IV.2 Comparaison* quantitative.

Nous constatons que notre méthode offre une bonne approximation de l'objet généré, notre méthode offre une reconstruction avec un erreur de 0.054 ce qui prouve son efficacité pour la reconstruction 3D de visage.

### 5.2.La Comparaison Qualitative

La figure ci-dessous présente une comparaison qualitative de la forme géométrique générée entre notre méthode et les méthodes de l'état de l'art [36][37].

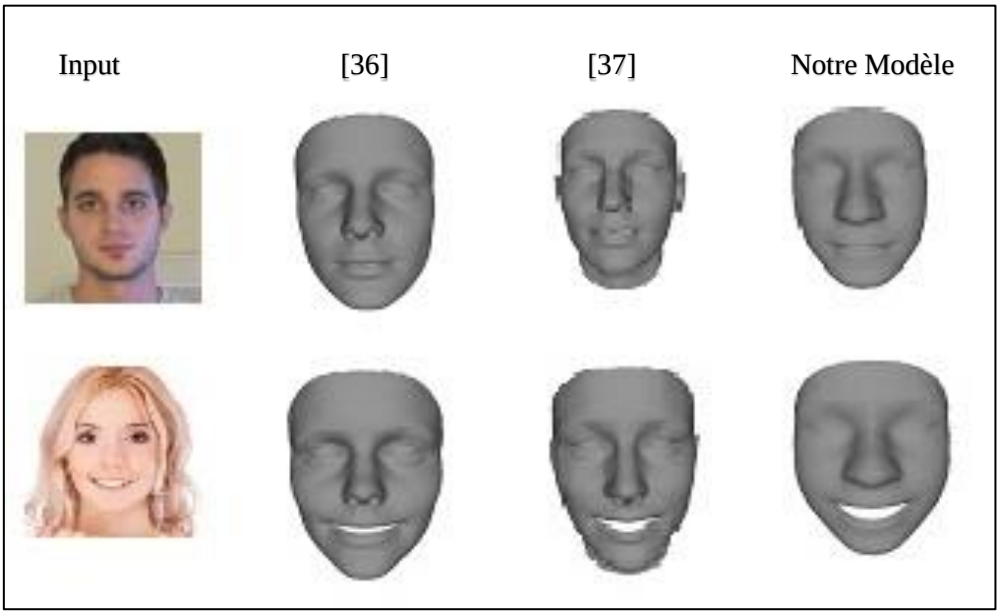



*Figure IV.8 Comparaison qualitative des formes géométriques générées par notre méthode et [36][37].*

Nous constatons que notre approche offre une bonne approximation du mesh générée.

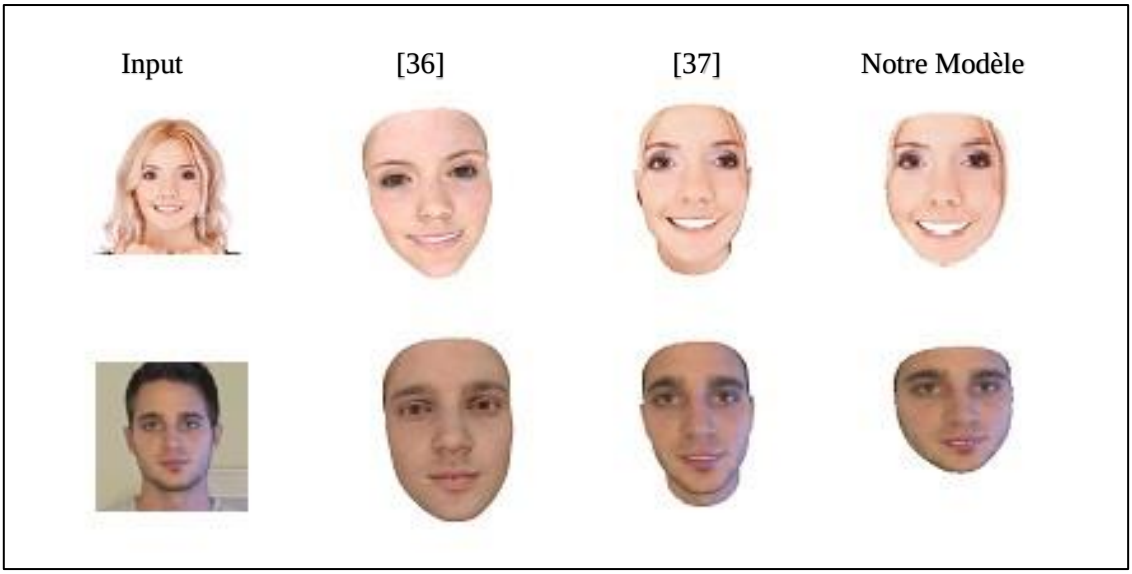



*Figure IV.9 Comparaison qualitative des formes géométriques avec textures générées par notre méthode et [36][37].*

La figure montre que notre méthode obtient de bonnes estimations de tous les paramètres du modèle, c.-à-d. pose, forme, expression, réflectance et éclairage de la scène. Nous obtenons des

résultats de qualité similaire à ces deux approches. En effet notre approche obtient également une estimation de la réflexion et l'éclairage de la peau colorée.

## 6. Conclusion

Dans ce chapitre nous avons présenté les résultats obtenus ainsi une comparaison quantitative et qualitative. Notre méthode basée sur l'utilisation du réseau de neurones convolutif capable de générer une reconstruction 3D du visage à partir d'une image 2D, en effet notre approche produit de bons résultats en termes de précision et de qualité visuel, notre méthode simple à mette en œuvre et offre une reconstruction 3D du visage en temps réel.

# Conclusion et Perspectives

# Conclusion générale et perspectives

La reconstruction 3D d'un visage à partir d'une image 2D en temps réel a toujours été un problème complexe, cette complexité réside dans la variation d'illumination, la mise en correspondance ainsi la profondeur de l'objet à reconstruire, afin de remédier aux insuffisances des techniques traditionnelles, nous présentons une méthode basée sur les réseaux de neurones convolutifs, en effet dans un première temps nous avons introduit une nouvelle base de données composée d'images réel collectés depuis le web et filtrée manuellement, notre méthode opère en trois parties la première consiste à entrainer un réseau de neurone convolutif sur notre propre base de données afin de produire une approximation des point Landmark dans l'espace image ensuite dans une seconde étape une mesh de visage est reconstruite, cependant la troisième étape consiste à effectuer une translation entre l'espace 3D de l'objet et l'espace 2D image (texture) afin de déterminer les cordonnées de texture qui correspondent à chaque polygone de visage. Notre méthode produit de bons résultats en termes de précision et de qualité visuel, et cela en prenant en compte tous les paramètres du modèle (forme, expression, réflectance et éclairage) en entrée, notre méthode simple à mette en œuvre et offre une reconstruction 3D du visage en temps réel. Pour les pistes futures nous envisagions une reconstruction 3D multi-visages depuis l'image d'entrée, l'optimisation de l'étape d'apprentissage afin d'améliorer la précisons de la reconstruction.

# Bibliographie

**[1]** Esling, P., & Devis, N. (2020). Creativity in the era of artificial intelligence. *arXiv preprint arXiv:2008.05959*.

**[2]** Cornuéjols, A., & Miclet, L. (2002). Apprentissage artificiel, concepts et algorithms, Eyrolles. ISBN 2-212-11020-0.

**[3]** Ezratty, O. (2018). Les usages de l'intelligence artificielle. Olivier Ezratty.

**[4]** https://www.avanade.com/-/media/asset/other/extrait-note-ia-banque.pdf

**[5]** Negrello, L. (1991). *Systèmes experts et intelligence artificielle*. Schneider Electric España SA.

**[6]** Golinska, P., Fertsch, M., Gómez, J. M., & Oleskow, J. (2007). The concept of closed-loop supply chain integration through agents-based system. In Information Technologies in Environmental Engineering (pp. 189-202). Springer, Berlin, Heidelberg.

**[7]** Touzet, C. (1992). *les réseaux de neurones artificiels, introduction au connexionnisme*.

**[8]** Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

**[9]** Luxton, D. D. (2016). An introduction to artificial intelligence in behavioral and mental health care. In Artificial intelligence in behavioral and mental health care (pp. 1-26). Academic Press.

**[10]** Ayodele, T. O. (2010). Types of machine learning algorithms. New advances in machine learning, 3, 19-48.

**[11]** Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), 1-130.

**[12]** http://www.psychomedia.qc.ca/lexique/definition/apprentissage-profond

**[13]** Rohrer, B. (2016). How do Convolutional Neural Networks work?. End-to-End Machine Learning, 18.

**[14]** Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

**[15]** Yang, G., & Huang, T. S. (1994). Human face detection in a complex background. Pattern recognition, 27(1), 53-63.

**[16]** Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. IEEE transactions on pattern analysis and machine intelligence, 15(10), 1042-1052.

**[17]** Sinha, P. (1994). Object recognition via image invariance a case study. Investigative ophthalmology and visual science, 35, 1735-1740.

**[18]** Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86.

**[19]** Bradski, G., Kaehler, A., & Pisarevsky, V. (2005). Learning-based computer vision with intel's open source computer vision library. Intel technology journal, 9(2).

**[20]** Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE.

**[21]** Gomez, G., & Morales, E. (2002, July). Automatic feature construction and a simple rule induction algorithm for skin detection. In Proc. of the ICML workshop on Machine Learning in Computer Vision (Vol. 31).

**[22]** Blanz, V., & Vetter, T. (2003). Face recognition based on fitting a 3d morphable model. IEEE Transactions on pattern analysis and machine intelligence, 25(9), 1063-1074.

**[23]** Blanz, V., Romdhani, S., & Vetter, T. (2002, May). Face identification across different poses and illuminations with a 3d morphable model. In Proceedings of fifth IEEE international conference on automatic face gesture recognition (pp. 202-207).

**[24]** Vetter, T., & Poggio, T. (1997). Face recognition by elastic bunch graph matching. IEEE Trans. Pattern Analysis and Machine Intelligence, 19(7), 733-742.

**[25]** Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification second edition john wiley & sons. New York, 58, 16.

**[26]** Besl, P. J., & Mckay, N. D. (1992). A method for registration of 3-D shapes, IEEE Trans. P flattern Anal. and M ac h ine I ntell, 1(4), 23.

**[27]** Tsalakanidou, F., Malassiotis, S., & Strintzis, M. G. (2004, May). Integration of 2D and 3D images for enhanced face authentication. In Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings. (pp. 266-271). IEEE.

**[28]** Gökberk, B., Salah, A. A., & Akarun, L. (2005, July). Rank-based decision fusion for 3D shape-based face recognition. In International Conference on Audio-and Video-Based Biometric Person Authentication (pp. 1019-1028). Springer, Berlin, Heidelberg.

**[29]** Kim, T. K., Kim, H., Hwang, W., Kee, S. C., & Kittler, J. (2003, June). Independent component analysis in a facial local residue space. In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. (Vol. 1, pp. I-I). IEEE.

**[30]** Meyer, A., Briceno, H. M., & Bouakaz, S. (2007, November). User-guided shape from shading to reconstruct fine details from a single photograph. In Asian Conference on Computer Vision (pp. 738-747). Springer, Berlin, Heidelberg.

**[31]** Verbin, D., & Zickler, T. (2020). Toward a Universal Model for Shape from Texture. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 422-430).

**[32]** Favaro, P., Jin, H., Yezzi, A., & Soatto, S. (2002, May). A variational approach to shape from defocus. In Proc. of the European Conference on Computer Vision.

**[33]** Klaus, A., Sormann, M., & Karner, K. (2006, August). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 3, pp. 15-18). IEEE.

**[34]** Dipanda, A., & Woo, S. (2005). Towards a real-time 3D shape reconstruction using a structured light system. Pattern recognition, 38(10), 1632-1650.

**[35]** Goldluecke, B., & Magnor, M. (2004, June). Space-time isosurface evolution for temporally coherent 3D reconstruction. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. (Vol. 1, pp. I-I). IEEE.

**[36]** Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., & Tong, X. (2019). Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

**[37]** Zhu, X., Liu, X., Lei, Z., & Li, S. Z. (2017). Face alignment in full pose range: A 3d total solution. IEEE transactions on pattern analysis and machine intelligence, 41(1), 78-92.

**[38]** Feng, Y., Wu, F., Shao, X., Wang, Y., & Zhou, X. (2018). Joint 3d face reconstruction and dense alignment with position map regression network. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 534-551).

**[39]** Richardson, E., Sela, M., Or-El, R., & Kimmel, R. (2017). Learning detailed face reconstruction from a single image. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1259-1268).

**[40]** Shang, J., Shen, T., Li, S., Zhou, L., Zhen, M., Fang, T., & Quan, L. (2020). Self-Supervised Monocular 3D Face Reconstruction by Occlusion-Aware Multi-view Geometry Consistency. arXiv preprint arXiv:2007.12494.

**[41]** Luo, Y., Tu, X., & Xie, M. (2019, September). Learning Robust 3D Face Reconstruction and Discriminative Identity Representation. In 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP) (pp. 317-321). IEEE.

**[42]** Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4401-4410).

**[43]** https://thispersondoesnotexist.com/

**[44]** Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T. S. (2012, October). Interactive facial feature localization. In European conference on computer vision (pp. 679-692). Springer, Berlin, Heidelberg.

**[45]** Kartynnik, Y., Ablavatski, A., Grishchenko, I., & Grundmann, M. (2019). Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs. arXiv preprint arXiv:1907.06724.

**[46]** Catmull, E., ssss& Clark, J. (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-aided design, 10(6), 350-355.

**[47]** https://lambdalabs.com/

**[48]** Python Software Foundation. Python Language Reference, version 3.6. Available at http://www.python.org

**[49]** https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript

**[50]** http://nodejs.org

**[51]** Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

**[52]** Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8110-8119).

**[53]** Threejs: Three.js (2020). https://threejs.org/

**[54]** https://www.kaggle.com/c/deepfake-detection-challenge/discussion/122786

**[55]** https://thispersondoesnotexist.com/

**[56]** Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T. S. (2012, October). Interactive facial feature localization. In *European conference on computer vision* (pp. 679-692). Springer, Berlin, Heidelberg.

**[57]** Bagdanov, A. D., Del Bimbo, A., & Masi, I. (2011, December). The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding* (pp. 79-80).

**[58]** Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, *15*(9), 850-863.