REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD2/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information Optimisation et Décision (SIOD)**

Deep learning pour le suivi des objets 3D réels dans la réalité augmentée

---

Par :

AICHA FERDJELLI

Soutenu le ../../…. devant le jury composé de :

Nom Prénom	grade	Président
Mohamed Chaouki Babahenini	Pr	Rapporteur
Nom Prénom	grade	Examinateur

Année universitaire 2020-2021

Remerciements

- *Mes sincères remerciements vont à Mr Mohamed Chaouki Babahenini pour son encadrement, ses conseils et sa disponibilité qu'il m'a témoignés pour me permettre de mener à bien ce travail.*
- *J'exprime mes vifs remerciements à :*
  - *.....a accepté de présider le Jury de soutenance,*
  - *..... pour m'avoir fait l'honneur d'accepter d'examiner ce travail.*

Enfin, je remercie tous ceux qui nous ont aidés de près ou de loin dans l'élaboration de ce travail.

Dedicace

A ma très chère mère Yakouta

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bien vaillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles

A mon très cher père Mohamed

Tu as toujours été à mes côtés pour me soutenir et m'encourager.

Que ce travail traduit ma gratitude et mon affection

A ma chère sœur : **Souad**

A mon très chère amie : **Ferielle**

Ma source de force et encouragement pour affronter les différents obstacles.

A mes sœurs : **Chaima ; Faten ; Rokaya**

A mes deux petites frères : **Mohamed et wassim**

Aux les familles : **Ferdjelli et cherguie**

Ferdjelli Aicha

Table de matière

Liste des figures

Liste des tableaux

Résumé

Le concept de Réalité Augmentée vise à accroître la perception du monde réel en y ajoutant des éléments non perceptibles à priori par l'œil humain. Plusieurs problèmes doivent être résolus pour obtenir une incrustation réaliste. Il faut tout d'abord pouvoir déterminer le point de vue adopté pour chaque prise de vue (alignement des caméras réelle et virtuelle) afin d'incruster l'objet de synthèse au bon endroit. Il faut ensuite tenir compte des interactions entre les éléments virtuels insérés et la scène réelle : traiter le problème d'occultation qui est dû essentiellement au mixage du monde réel avec les objets virtuels utilisés pour l'augmentation de la scène.
Le présent projet se propose pour traiter ce problème, pour cela plusieurs techniques ont été proposé dans la littérature, nous avons choisi d'utiliser l'apprentissage profond pour d'une part effectuer une segmentation d'image afin d'identifier les objets d'intérêts et enlever l'arrière-plan et d'autre part gérer les occultations.

Mots-clés : Réalité augmentée, Segmentation, Occultation, auto-encodeur, convolution, Réseau de neurone (CNN).

Abstract

The concept of Augmented Reality aims to increase the perception of the real world by adding elements not perceptible a priori by the human eye. There are several issues that need to be resolved to achieve a realistic overlay. First of all, it is necessary to be able to determine the point of view adopted for each shot (alignment of the real and virtual cameras) in order to embed the synthetic object in the right place. It is then necessary to take into account the interactions between the inserted virtual elements and the real scene: to deal with the occultation problem which is mainly due to the mixing of the real world with the virtual objects used for the augmentation of the scene.

The present project proposes to deal with this problem, for this several techniques have been proposed in the literature, we have chosen to use deep learning to on the one hand perform an image segmentation in order to identify the objects of interests and remove the background and on the other hand manage the occultations.

Keywords : Augmented reality, Segmentation, Occultation, convolution auto-encoder, Neurone network (CNN

ملخص

يهدف مفهوم الواقع المعزز إلى زيادة إدراك العالم الحقيقي عن طريق إضافة عناصر غير محسوسة مسبقًا بالعين البشرية. هناك العديد من القضايا التي يجب حلها لتحقيق تراكب واقعي. بادئ ذي بدء ، من الضروري أن تكون قادرًا على تحديد وجهة النظر المعتمدة لكل لقطة (محاذاة الكاميرات الحقيقية والافتراضية) من أجل تضمين الكائن الاصطناعي في المكان المناسب. من الضروري بعد ذلك مراعاة التفاعلات بين العناصر الافتراضية المدرجة والمشهد الحقيقي: للتعامل مع مشكلة الاختفاء التي ترجع أساسًا إلى اختلاط العالم الحقيقي بالأشياء الافتراضية المستخدمة لتكبير المشهد.

يقترح المشروع الحالي التعامل مع هذه المشكلة، نظرًا لأنه تم اقتراح العديد من التقنيات في الأدبيات، فقد اخترنا استخدام التعلم العميق من ناحية إجراء تجزئة للصورة من أجل تحديد الأشياء ذات الاهتمام وإزالة الخلفية و من ناحية أخرى إدارة الإخفاء..

الكلمات المفتاحية: الواقع المعزز، التقسيم، الانسداد، التشفير التلقائي للتلف، شبكة الخلايا العصبية.

Introduction générale

Les applications de réalité augmentée (RA) existantes, ignorent souvent l'occultation entre des mains réelles et des objets virtuels lorsqu'elle incorpore des objets virtuels dans les vues de l'utilisateur.

Les défis proviennent du manque de profondeur précise et du décalage entre profondeur réelle et la profondeur virtuelle.

Ce travail propose une nouvelle approche qui prédit directement l'occultation réelle et virtuelle et contourne l'acquisition et l'inférence de profondeur.

Notre objectif est d'améliorer les applications AR avec des interactions entre les mains (réels) et objets saisissables (virtuels). Avec des images appariées de main et objet comme entrées, nous formulons un neural profond CNN pour la segmentation et l'extraction des caractéristiques et un réseau compact qui apprend à générer le masque d'occultation.

Pour entraîner le réseau, nous compilons un grand ensemble de données, y compris des données synthétiques et des données réelles.

Nous intégrons ensuite le réseau formé dans un système AR de prototypage pour prendre en charge la saisie en temps réel d'objets virtuels. De plus, nous démontrons la performance de la méthode sur divers objets virtuels.

Dans ce mémoire, nous allons discuter toutes les étapes importantes permettant de réaliser ce système.

Ainsi, nous avons partitionné notre manuscrit en quatre chapitres. Dans le premier chapitre, nous allons parler de la RA et son fonctionnement en un cadre général, ensuite au deuxième chapitre nous allons exposer l'apprentissage profond pour la RA et les méthodes principales utilisées alors que le troisième chapitre contiennent la conception générale et détaillée pour résoudre le problème de l'occultation et la segmentation des objets virtuels par rapport à la scène réelle. Enfin nous terminerons par le quatrième chapitre de notre mémoire par la présentation des résultats obtenus.

Chapitre 1.
La réalité augmentée

1.1 Introduction

La réalité augmentée (RA) est une expérience interactive d'un environnement du monde réel où les objets qui résident dans le monde réel sont améliorés par des informations perceptives générées par ordinateur, parfois à travers de multiples modalités sensorielles, notamment visuelles, auditives, hépatiques, somatosensorielles et olfactives (ababsa)

Dans ce chapitre, nous allons parler de la RA, de ses fonctionnements et de ses relations avec l'apprentissage profond. Nous allons aussi approfondir la notion de l'occultation des objets due à l'intégration des objets virtuels avec la scène réelle.

1.2 Continuum de Milgram

Le terme « Réalité augmentée » (RA) est introduit par (Caudell & Mizell, 1992) en 1992. L'auteur prototype un casque de type Head-up display permettant de connaître la position de la tête dans le monde réel et d'afficher du contenu virtuel à travers un système Optical see-through en temps réel. En 1994, Milgram propose une taxonomie pour la réalité mixte (Milgram & Kishino, 1994). Il émet l'hypothèse qu'aucune frontière franche n'existe entre le monde réel et le monde virtuel, mais plutôt qu'il est possible de passer d'un monde à l'autre à travers un continuum, appelé la réalité mixte (Milgram P (1994)).

Figure 1.1: Milgram's Reality-Virtuality Continuum (K. A. Milgram P 1994)).

En se référant à sa définition, nous pouvons placer la réalité augmentée plutôt du côté du monde réel. Par la suite (Azuma & others, 1997) propose une définition plus structurée de la réalité

augmentée et qui ne dépend pas de la technique employée. Tout système de réalité augmentée doit respecter ces trois règles :

- Combiner le réel et le virtuel.
- Être interactif en temps réel.
- Être synchronisé en 3D.

Contrairement à la réalité virtuelle, où l'utilisateur est entièrement immergé dans un monde calculé numériquement, la réalité augmentée a pour but d'ajouter quelques contenus virtuels au monde réel. L'utilisateur voit donc principalement le monde réel.

L'objectif de l'intégration des objets virtuels est d'apporter une information pertinente et contextualisée donnant un sens nouveau à l'utilisateur du système de réalité augmentée.

Le sens donné dépendra évidemment de l'application souhaitée : marketing, médical, communication, maintenance, jeu, formation, loisir, sport, éducation (Van Krevelen & Poelman, 2010) (Nee, et al. 2012) (Cieutat, 2013). Afin de faciliter la désignation des objets virtuels ajoutés, nous proposons cette définition : « Augmentation : nom féminin, objet virtuel ajouté au monde réel au moyen d'un système d'affichage de réalité augmentée. Cet objet est synchronisé en 3D et en temps réel. ». L'utilisation de ce terme dans la suite du document se référera à cette définition.

Il convient de noter que la réalité augmentée est essentiellement basée sur la vision, et c'est cet usage qui nous intéressera dans ce mémoire. Cependant, il est aussi possible de créer des systèmes de réalité augmentée pour les autres sens. Par exemple, (Joseph, et al. 2013) crée un système permettant à un malvoyant d'obtenir une information par retour sonore ou haptique afin de le diriger dans un bâtiment (Lee B 2010))

Le modèle expose les concepts suivants : la réalité, la réalité augmentée, la virtualité augmentée, la virtualité et la réalité mixée**.** (] Ababsa F (2008))

La réalité : c'est note environnement réel**,** ce que l'on voit. Elle indique que quelque chose existe concrètement (Huang Y (2009))**.**

La virtualité : représentation de ce qui est produit par une unité numérique et qui n'a pas de conséquence sur l'actuel (Nilsson J 2010)

La virtualité augmentée : c'est une simulation informatique d'espace, de lieux réels ou non au niveau de cinq sens où l'on immerge un individu.

La réalité mixte : est une combinaison d'objet de modèle réel et objet virtuelle, la RM englobe un large éventail de technologie, allant de la réalité augmentée à la VA. (Bangor 2009)

1.3 Principes de fonctionnement de la réalité augmentée

Tout projet mis en place doit évaluer les contraintes et les risques. La réalité augmentée n'échappe pas à ce modèle. Afin de mieux comprendre les contraintes liées à la réalité augmentée basée sur la vision, il est nécessaire d'en détailler le principe de fonctionnement. L'objectif principal pour un système de réalité augmentée est de connaître la pose (i.e. position et orientation) de la caméra filmant la scène par rapport à un objet connu dans la scène. Une fois cela déterminé, il est possible d'ajouter tout type d'augmentation mixant réel et virtuel. Le principe de fonctionnement peut être décomposé en quatre étapes (voir figure 1.2).

- La caméra filme la scène.
- L'objectif étant de connaître la pose de la caméra (i.e. position, orientation), dans chaque image filmée, un objet (marqueur, image de référence, modèle 3D) est reconnu.
- Connaissant la pose de la caméra, les augmentations (objets virtuels) sont ajoutées en cohérence avec le monde réel
- Le rendu est obtenu sur l'appareil d'affichage (PC, smartphone, tablette, lunettes intelligentes).

La figure 1.2 illustre ce principe de fonctionnement. Tout d'abord, la caméra de la tablette ou des lunettes intelligentes filment la scène. Ensuite, l'algorithme utilisé reconnaît le marqueur placé sur le système à opérer. Enfin, l'écran de la tablette ou des lunettes intelligentes affiche le contenu en réalité augmentée

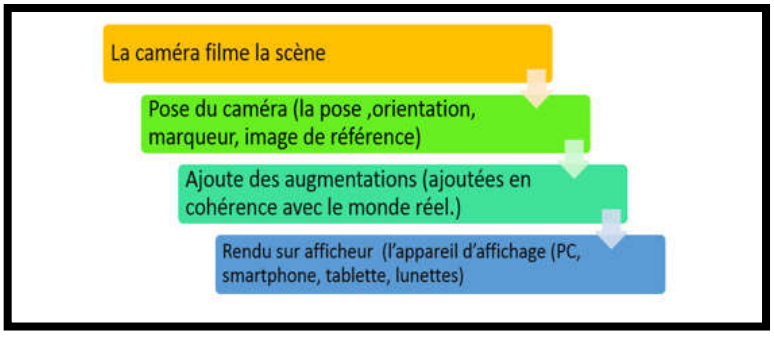


Figure 1.2 : Principes de fonctionnement de la réalité augmentée

La seconde étape « recalage et suivi », est réalisée avec deux types d'algorithme relativement similaires (ARToolKit, 1999) (Lowe, 1999) (Comport, et al 2006) :

- **Le recalage** (ou registration en anglais) Dans cette étape, un objet cible est recherché dans la scène. Une fois détecté, la pose de la caméra est trouvée. Cette étape est la plus exigeante en calcul car elle nécessite de rechercher l'objet cible dans toute l'image récupérée par la caméra.
- **Le suivi** (ou le tracking en anglais) En supposant que les mouvements de la caméra sont petits, la pose de la caméra peut être estimée à chaque rafraîchissement en utilisant des algorithmes optimisés. Par exemple, avec du suivi de points d'intérêts, les points d'intérêts ne sont plus recherchés dans toute l'image, mais seulement dans une zone autour du point d'intérêt détecté dans l'image précédente.

Cette étape nécessite moins de temps de calcul et elle est évidemment privilégiée, une fois le recalage effectué.

Dans la littérature, les deux termes ne sont pas toujours distingués. La plupart du temps, on utilise plutôt le terme suivi (ou tracking en anglais) pour regrouper les deux concepts. Dans la suite du document**.** (Bartlett 1937)

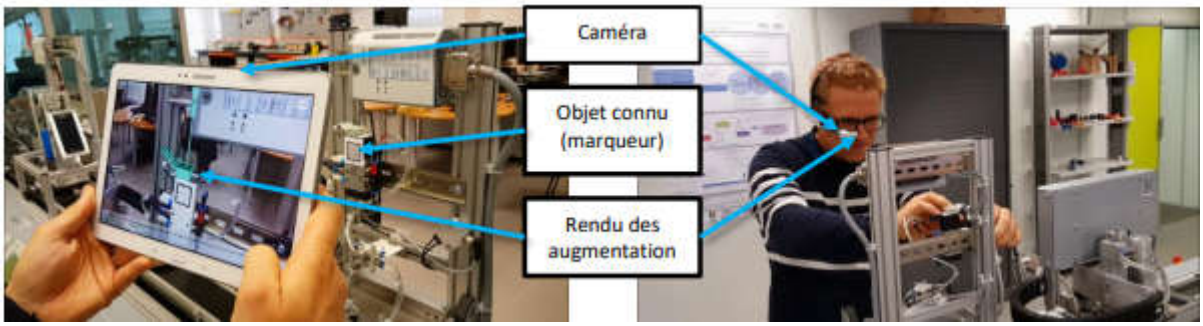


Figure 1.3 : Le suivi (ou le tracking).

1.4 Domaines d'application

Au fil des ans, les chercheurs et les développeurs trouvent de plus en plus de domaines qui pourraient bénéficier d'une augmentation. La première des systèmes axés sur les applications militaires, industrielles et médicales, mais des systèmes AR à des fins commerciales et de divertissement est apparu peu de temps après. Laquelle de ces applications déclenchera une utilisation largement répandue est une supposition pour quiconque. Cette section traite certains domaines d'application regroupés similaire à l'ISMAR 2007 Symposium30 catégorisation.

1.4.1. Publicité et commercial

La réalité augmentée est principalement utilisée par les spécialistes du marketing pour promouvoir de nouveaux produits en ligne. Les plus les techniques utilisent des marqueurs que les utilisateurs présentent devant leur webcam soit sur des logiciels spéciaux ou simplement sur le site Web de la société de publicité.

Par exemple, en décembre 2008, MINI (Mini. . Official Homepage. Retrieved January 29, 2020, from https://www.mini.com/ (2018, March 27)) Le célèbre constructeur automobile, a diffusé une publicité en réalité augmentée dans plusieurs automobiles allemandes magazines (Cool: Augmented Reality Advertisements. 29, 2020, from https://geekologie.com/2008/12/cool-augmented-reality-adverti.php 2008, December 19))**.** Le lecteur devait simplement se rendre sur le site Web de la MINI**,** afficher l'annonce devant de leur webcam, et une MINI 3D est apparue sur leur écran, comme le montre la figure 1.4. Au-delà de la réalité (Marco Sacco s.d.) a publié un magazine publicitaire sans marqueur de 12 pages qui pourrait être reconnu et animé par un logiciel que l'utilisateur peut télécharger sur le site Web de l'éditeur comme point de départ à leurs jeux de réalité augmentée. Ils voient qu'avec un tel système, ils pourraient ajouter un "payant" option sur le logiciel qui permettrait à l'utilisateur d'accéder à du contenu supplémentaire, comme voir une bande-annonce et puis être en mesure de cliquer sur un lien pour voir le film complet, transformant le magazine en un de cinéma (Marco Sacco s.d.)

AR offre également une solution au problème coûteux de la construction de prototypes. En effet, industriel les entreprises sont confrontées à la nécessité coûteuse de fabriquer un produit avant sa commercialisation pour déterminer si des modifications doivent être apportées et voir si le produit répond aux attentes. Si on décide que des modifications doivent être apportées, et c'est le plus souvent le cas, un nouveau prototype doit être fabriqué et du temps et de l'argent supplémentaires sont gaspillés.

Figure 1.4 : MINI advertisement (Marco Sacco s.d.)

Un groupe de l'Institut des technologies industrielles et de l'automatisation (ITIA) du National Le Conseil de la recherche (CNR) d'Italie (Marco Sacco s.d.) de Milan travaille sur les systèmes

AR et VR en tant qu'outil pour prise en charge du prototypage virtuel. L'ITIA-CNR s'implique dans la recherche de contextes industriels et application utilisant la RV, la RA, la 3D en temps réel, etc. comme support pour les tests et le développement de produits et évaluation. Quelques exemples de projets de recherche appliquée où les technologies ci-dessus ont été appliqués incluent le prototypage de motos (Fig. 1.5), l'aménagement virtuel d'une usine et d'un bureau (Fig. 1.6) et 1.8), simulation de lumière virtuelle (Fig. 1.7) et essai virtuel de chaussures avec le Magique Interface miroir, qui sera discutée ensuite (Fig. 1.9).

Figure 1.5 : Photo d'un prototype de moto virtuel (à droite) à côté d'un prototype de moto physique (à gauche) dans l'environnement réel ; image de gauche : prototype virtuel dans un environnement virtuel (Marco Sacco s.d.)**.**

Figure 1.6 : Prototype d'usine virtuelle (Marco Sacco s.d.)**.**

La figure 1.7 montre des exemples similaires à l'utilisation de la RA par Magique Mirror pour la publicité et les applications commerciales en remplaçant totalement le besoin d'essayer en magasin, ce qui permet de gagner un temps considérable pour les clients, qui serait très probablement utilisé pour essayer plus de vêtements (chemises, robes, montres, pantalons, etc.) et augmentant ainsi les chances de vente des magasins.

La réalité augmentée n'a pas pleinement atteint le marché industriel des applications publicitaires principalement parce que quelques améliorations doivent être apportées aux systèmes similaires au Magique Mirror (Fig. 1.7) ou la cabine d'essayage de Cisco (Fig. 1.8).

Figure 1.7 : L'utilisateur essaie des chaussures virtuelles devant le Magique Mirror (Marco Sacco s.d.)**.**

Figure 1.8 : Publicité AR de Cisco où un client s'habille devant un écran « magique » (Marco Sacco s.d.)**.**

En effet, pour que le produit soit viable sur le marché, il doit fournir à l'utilisateur une représentation du prototype ; l'utilisateur doit avoir l'impression de regarder un prototype physique. Dans le cas du système Magique Mirror, cela signifierait un suivi sans faille de sorte que lorsque l'utilisateur regarde le miroir magique, il a l'impression de porter les chaussures et peut vraiment voir à quoi ressembleraient les chaussures.

1.4.2. Divertissement et éducation

Les applications de divertissement et d'éducation incluent des applications culturelles avec des visites touristiques et des musées guidage, applications de jeu avec des jeux traditionnels utilisant des interfaces AR et certaines applications pour téléphones intelligents qui utilisent la RA à des fins de divertissement et / ou éducatives. (Voir figure 1.6 : prototype de bureau virtuel **[17]**, figure 1.6 : Test virtuel du système d'allumage (Marco Sacco s.d.)**, figure** 1.7 : Utilisateur essayant des chaussures virtuelles devant le Magique Mirror (Marco Sacco s.d.)**, f**igure 1.8 : Publicité de Cisco AR où se trouve un client essayer des vêtements devant un écran «magique»).

Dans l'application culturelle, il existe quelques systèmes qui utilisent la RA pour reconstruire virtuellement des ruines antiques, comme dans (Malaka R (2004))**,** ou pour informer virtuellement l'utilisateur sur l'histoire du site, comme dans (Bruna E (2007)).

Figure 1.9 : Vue augmentée de Dashuifa depuis (Miyashita T (2008))

Figure 1.10 : Guidage par téléphone portable dans un musée de (Bruna E (2007))**.**

Il existe également quelques systèmes qui exploitent la RA pour guider les musées tels que et (Bichlmeier C 2007)**.** Dans (Bruna E (2007)) et (Bichlmeier C 2007)**,** les deux systèmes sont mobiles, mais (Bruna E (2007)) utilise également un téléphone mobile comme interface (Fig1.10) tandis que **[20]** utilise simplement une configuration de lentille magique (Fig. 1.11). Dans (Bruna E (2007))**,** les auteurs identifiés les avantages de l'utilisation de la réalité augmentée comme interface pour leurs applications culturelles comme une communication efficace avec l'utilisateur à travers des présentations multimédias naturelles et une technique intuitive et de faibles coûts d'entretien et d'acquisition pour les exploitants du musée. Et en effet, l'utilisation d'un smartphone ou même d'un autre écran portatif est une technique plus

intuitive et naturelle que de rechercher un numéro attribué au hasard à l'objet Dans un petit guide écrit, en particulier lorsque l'utilisateur peut simplement utiliser son propre téléphone dans un monde où tout le monde en possède un. De même, les utilisateurs peuvent s'identifier plus facilement aux présentations multimédias qui leur sont apportées et écouteront, regarderont et / ou liront plus volontiers les informations qu'ils peuvent acquérir en pointant simplement un objet à l'aide de son téléphone plutôt que d'avoir à le rechercher dans un guide.

Figure 1.11 : Visiteur avec système de guidage de **[**(Bichlmeier C 2007)

1.4.3. Applications médicales

La plupart des applications médicales concernent la chirurgie guidée par l'image et assistée par robot. Par conséquent, des recherches importantes ont été menées pour intégrer la RA avec l'imagerie et les instruments médicaux incorporant les capacités intuitives du médecin. Une percée significative a été fournie par l'utilisation de divers types d'imagerie médicale et d'instruments, tels que des images vidéo enregistrées par un dispositif de caméra endoscopique présenté sur un moniteur visualisant le site opératoire à l'intérieur du patient. Cependant, ces avancées limitent également la 3D naturelle, intuitive et directe du chirurgien pour la vision du corps humain car les chirurgiens doivent désormais gérer les repères visuels d'un autre environnement fourni sur le moniteur. La RA peut être appliqué afin que l'équipe chirurgicale puisse voir les données d'imagerie en temps réel pendant la progression de la procédure. (Bichlmeier C 2007) a introduit un système AR pour visualiser à travers la « vraie » peau sur l'anatomie virtuelle à l'aide de modèles de surface polygonale pour permettent une visualisation en temps réel (Fig. 1.12). Les auteurs ont également intégré l'utilisation de la navigation outils chirurgicaux pour augmenter la vue du médecin à l'intérieur du corps humain pendant la chirurgie.

Figure 1.12 : Bichlmeier et. Al. Système de visualisation à travers la peau (Bichlmeier C 2007)

1.4.4. Applications mobiles

C'est le projet légendaire de Niantic qui a introduit les jeux de réalité augmentée dans les foules. La nostalgie des joueurs a permis à Pokémon GO de détenir plusieurs records Guinness et gagner des revenus incroyables. Le rôle de ce jeu AR en termes de réalisation de réalité augmentée la technologie connue de millions de personnes est difficile à exagérer (Stutzman B (2009))**.**Pokémon GO est un jeu mobile de type aventure (Fig. 1.13), qui place le champ de bataille sur l'environnement réel et superpose des objets virtuels animés dans l'environnement de l'utilisateur. Des millions, voire des milliards de Pokémons ont été capturés depuis 2016, et le plaisir continue encore (Stutzman B (2009)) **.**

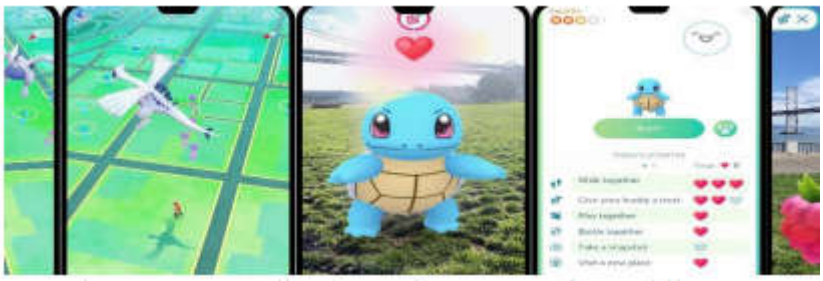


Figure 1.13: Application Pokémon GO for mobile (Stutzman B (2009))**.**

Figure 1.15: WikitudeDrive (Stutzman B (2009)) **Figure 1.14**: Restaurant Guide (Stutzman B (2009))

1.5 Occultation pour RA

Le problème de l'occultation se produit lorsque les objets réels sont devant les objets virtuels dans la scène. Sans gestion de l'occultation, les utilisateurs auront l'idée fausse que l'objet réel est plus loin du point de vue que les objets virtuels lorsque les objets virtuels sont occlus par les objets réels de la scène (N. Parmar J. Uszkoreit L. Jones A. N. Gomez L. Kaiser I. Polosukhin Aswani 12 2018)**.** Un exemple de problème d'occultation est montré sur la figure 1.16. Dans cette figure, l'objet virtuel est une cible. La figure 1.16. (a) est l'image composée vue du haut vers le bas de la scène. Nous pouvons constater que la cible est derrière le caddy rouge si nous voyons du point de vue comme le montre la figure 1.16 (c). Cela signifie qu'une partie de la cible doit être obstruée par le caddy rouge. La figure 1.16 (b) est l'image composée superposée avec une cible virtuelle sans gestion d'occlusion. Le l'image donne l'impression que la cible est devant le caddy, ce qui n'est pas le cas En réalité

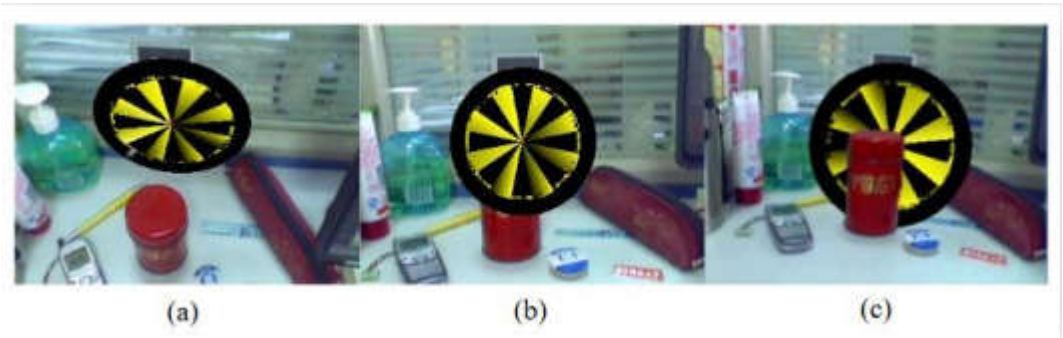


Figure 1.16 : Problème de l'occultation pour RA

1.6 Réalité augmentée et deep Learning

L'apprentissage en profondeur constitue une des technologies les plus importantes qui peuvent renforcer les applications et les expériences de RA.

L'apprentissage en profondeur peut inculquer l'intelligence dans les systèmes de RA et peut être utilisé comme un moyen d'améliorer la vision par ordinateur.

Certes c'est en termes de développement une compétence particulière, parfois un peu éloignée de celle des développeurs 3D : il faut utiliser le bon modèle de réseau neuronal, adapté à l'activité cible mais aussi au processeur cible. Produire le dataset, savoir entraîner correctement le réseau sont essentiels.

Nous présentons dans ce qui suit quelques travaux récents, qui ont utilisés l'apprentissage profond pour la réalité augmentée :

Akgul et al. (2016) ont mené une étude afin de résoudre certains problèmes de détection AR en modifiant les architectures actuelles de deep Learning. Ils ont examiné des méthodes de détection de pointe similaires pour le suivi AR. Et ont présenté leur détecteur CNN profond appelé DeepAR et ont suivi des approches de détection basées sur les fonctionnalités (A Akgul 2016).

Limmer et al. (2016) ont proposé une approche approfondie basée sur CNN à plusieurs échelles, capable de prédire et de localiser le parcours routier la nuit à l'aide de capteurs de caméra tirant parti des techniques d'apprentissage en profondeur (M Limmer 2016).

Abdi et Meddeb (2017) ont présenté une nouvelle approche pour la reconnaissance en temps réel des panneaux de signalisation (TSR) basée sur l'apprentissage profond en cascade et la RA (Meddeb (2017)).

1.7 Conclusion

Les systèmes de réalité augmentée (RA) visent à ajouter des objets virtuels à une scène réelle pour rendre le virtuel les objets fusionnent avec le monde existant de manière transparente de manière à apparaître comme faisant partie du visionné la scène 3D. Contrairement à la réalité virtuelle, les utilisateurs peuvent voir des objets virtuels et le monde réel simultanément dans le système de réalité augmentée. Pour renforcer l'illusion que les objets virtuels sont réellement présents dans la scène réelle, les chercheurs ont accordé de plus en plus d'attention au problème de l'occlusion. Dans le chapitre suivant, nous présenterons les méthodes utilisées dans l'apprentissage profond avec RA.

Chapitre 2.
.Apprentissage profond pour la RA

2.1 Introduction

La réalité augmentée consiste à enrichir le contenu visuel du monde réel avec des objets virtuels. L'obtention de résultats réalistes implique la résolution de tâches difficiles de vision par ordinateur, telles que la segmentation et la gestion de l'occultation d'une scène.

Pour ce faire plusieurs techniques ont été proposées dans la littérature, mais la tendance de recherche actuelle à partir de 2015 se dirige vers les techniques d'apprentissage automatique principalement l'apprentissage profond pour résoudre les problèmes posés par le mixage du monde réel par les objets virtuels.

Dans ce chapitre, nous présentons comment ces deux tâches difficiles peuvent être résolues de manière robuste et grâce à l'apprentissage en profondeur. Dans les deux cas, les réseaux de neurones convolutés profonds s'enregistrent sur de grandes quantités de données et obtiennent des résultats de pointe.

2.2 Apprentissage profond

2.2.1 Principe

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique dans l'intelligence artificielle qui dispose de réseaux capables d'apprendre sans surveillance à partir de données non structurées ou non étiquetées. Également connu sous le nom d'apprentissage neuronal profond ou de réseau neuronal profond. (A. G. Baydin (2015))

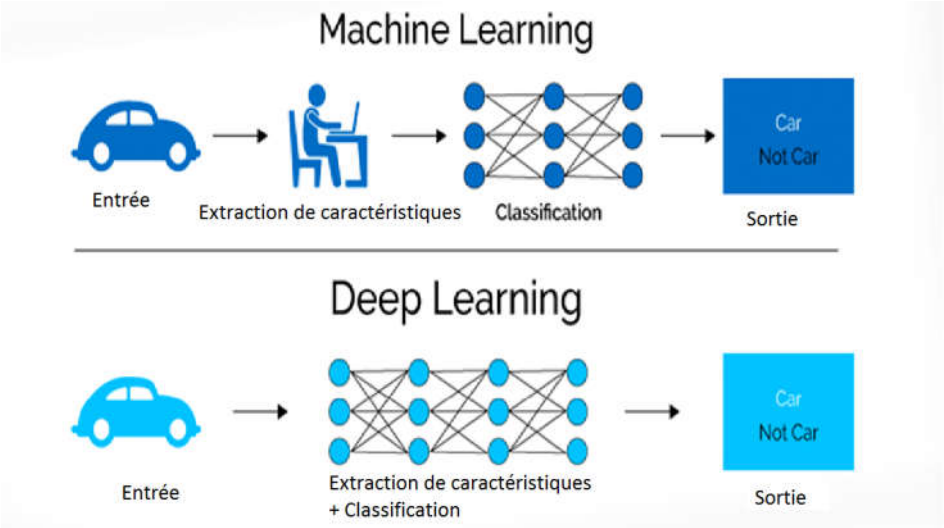


Figure 2.1 : Apprentissage profond et apprentissage automatique (A. G. Baydin (2015)).

2.2.2 Types d'apprentissage automatique

2.2.2.1 Méthodes supervisées

Dans le cas de l'apprentissage supervisé, on dispose d'un ensemble de données étiquetées, ou d'exemples qui se sont vus associés une classe par un professeur ou un expert. Cet ensemble d'exemples constitue la base d'apprentissage. Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de construire à partir de la base d'apprentissage, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier, la classe (Figure.2.2) c

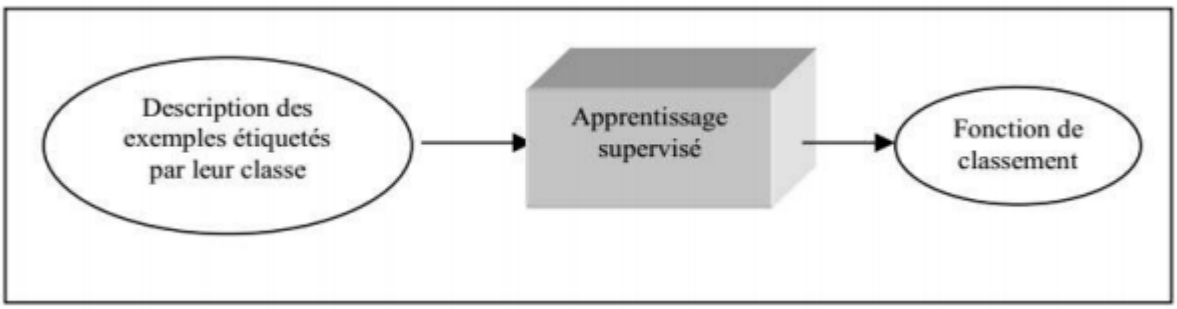


Figure 2.2 -L'apprentissage supervisé (Sinno Jialin Pan et Qiang Yang 2010,)

Dans **Figure 2.2** l'inférence inductive est définie comme un processus qui à partir d'une connaissance spécifique observée sur certains objets et d'une hypothèse inductive initiale, permet d'obtenir une assertion inductive impliquant ou rendant compte fortement ou faiblement des observations. Dans le cas de l'apprentissage inductif supervisé, qui est un sous domaine de l'inférence inductive, la connaissance spécifique consiste en un ensemble d'objets appartenant à des classes connues. L'assertion inductive est exprimée par une règle de classification qui

assigne une classe à chaque objet. L'implication forte est satisfaite si la règle classe correctement tous les objets connus (Sinno Jialin Pan et Qiang Yang 2010,)

2 2.2.2 Méthodes non-supervisées

L'apprentissage non-supervisé, encore appelé apprentissage à partir d'observations ou découverte, consiste à déterminer une classification « sensée » à partir d'un ensemble d'objets ou de situations données (des exemples non étiquetés).

On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes. Ce type d'apprentissage, encore appelé Cluster ING ou Cluster Analysais, se trouve en classification automatique et en taxinomie numérique. Cette forme de classification existe depuis des temps immémoriaux. Elle concerne notamment les sciences de la nature (Figure 2.3), les classifications des documents et des livres mais également la classification des sciences élaborées au cours des siècles par les philosophes (Sinno Jialin Pan et Qiang Yang, « A Survey on Transfer Learning », IEEE Transactions on Knowledge and Data Engineering, 2010)

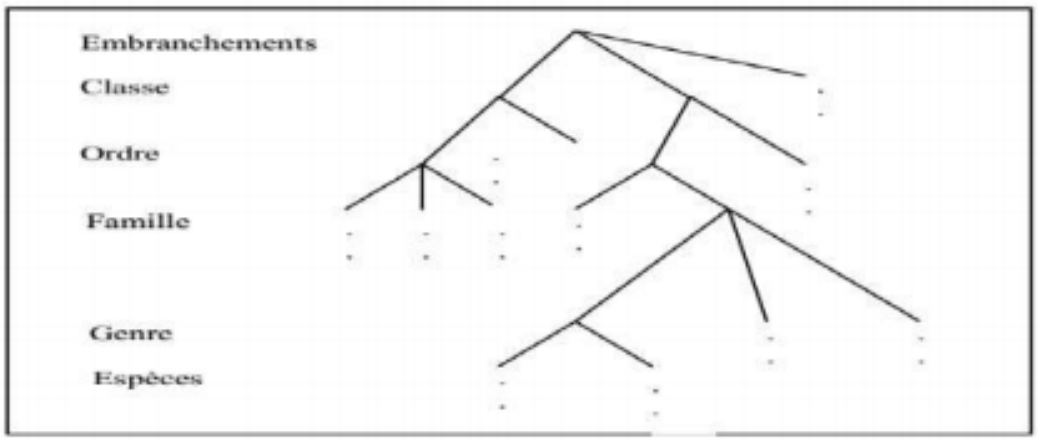


Figure 2.3- Extrait de la classification taxinomique de Linné.

L'automatisation de la construction de classification constitue aujourd'hui un véritable domaine de recherche. La notion clé utilisée pour créer des classes d'objets est une mesure de la similarité entre les objets. Les classes ou concepts sont construits de façon à maximiser la similarité intra-classes et à minimiser la similarité interclasses. L'apprentissage non supervisé correspond également à la classification conceptuelle, où une collection d'objets forme une classe si cette classe peut être décrite par un concept, compte tenu d'un ensemble de concepts prédéfinis.

2.2.3 Les Réseaux de Neurones

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate : le neurone formel. Les réseaux neuronaux sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone. (Masakazu Matusugu s.d.)

Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage marketing, reconnaissance de formes, traitement de signal,) d'estimation (modélisation de phénomènes complexes,…) et prévision (bourse, ventes,…). Il existe un compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à

Comprendre, mais moins il sera capable de prendre en compte des dépendances trop variées (Masakazu Matusugu s.d.)

2.2.4- Présentation de quelques Types de réseaux neuronaux

Il existe beaucoup de types de réseaux neurones, chaque type étant développe pour un objectif particulier (Masakazu Matusugu s.d.)

2.2.4.1 Neurone Formel

Un neurone formel est une représentation mathématique et informatique d'un neurone biologique. Le neurone formel possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et au cône d'émergence du neurone biologique (point de départ de l'axone). Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées. Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage. Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone. Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle. (Rock (1990))

2.2.4.2 Neurones multicouche

Le perceptron multicouche (multi layer perceptron MLP) est un classifier linéaire de type réseau neuronal formel organisé en plusieurs couches (Figure 2.4) au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau de type feedforward (en). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système.

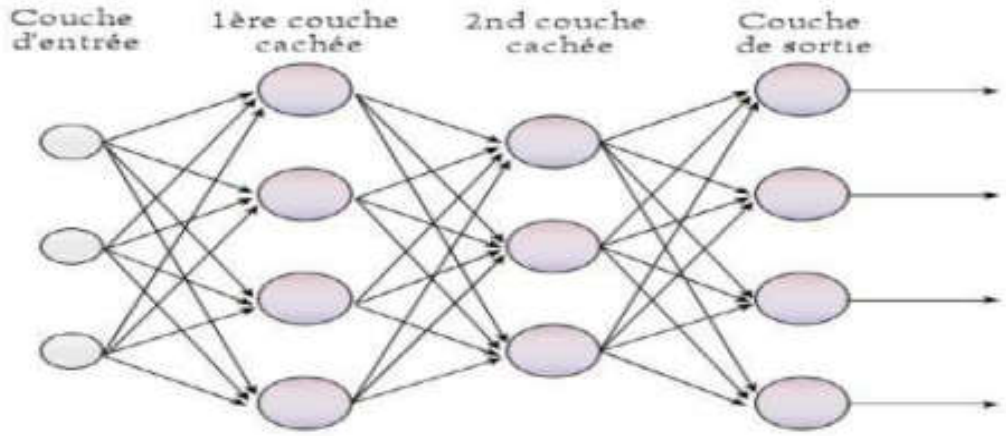


Figure 2.4-Perceptron multicouche (Rock (1990))

2.2.4-3 Neurones récurrents

Les réseaux de Neurones récurrents (RNNs) permettent d'analyser les séquences de vecteurs tout comme les modèles de Markov cachés. Le temps entre ici en ligne de compte car les sorties (de la coche de sortie et/ou de la coche cachées) calculées à l'instant t sont réinjectées en entrée du réseau et/ou en entrée de la couche cachée. On peut en théorie conserver dans le réseau la mémoire de ce qui s'y est passé depuis le début**.** (Rock (1990))

2.2.4. 4.Réseaux de Hop Field

Le réseau de neurones d'Hop Field est un modèle de réseau de neurones récurrents à temps discret dont la matrice des connexions est symétrique et nulle sur la diagonale et où la dynamique est asynchrone (un seul neurone est mis à jour à chaque unité de temps). Il a été découvert par le physicien John Hop Field en 1982.

Sa découverte a permis de relancer l'intérêt dans les réseaux de neurones qui s'était essoufflé durant les années 1970 à la suite d'un article de Marvin Minsky et Seymour Papert.

Un réseau de Hopfield est une mémoire adressable par son contenu : une forme mémorisée est retrouvée par une stabilisation du réseau, s'il a été stimulé par une partie adéquate de cette forme. (Maddison, et al. (2014))

2.2.4.5-Réseaux Neurones Convolution els

En apprentissage automatique, un réseau de neurone convolutés (ou réseau de neurones à convolution, ou CNN ou ConvNet) est un type de réseau de neurones artificiels acycliques dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter3 de petites quantités d'informations. Les réseaux neuronaux convolutés ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel**.** (Rock (1990))

Principe d'architecture d'un CNN

Les réseaux de neurones convolutés sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolution Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. **[41]**

La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. La première partie d'un CNN est la partie convolutée à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Figure 2.5).certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local.

En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. (Rock (1990))

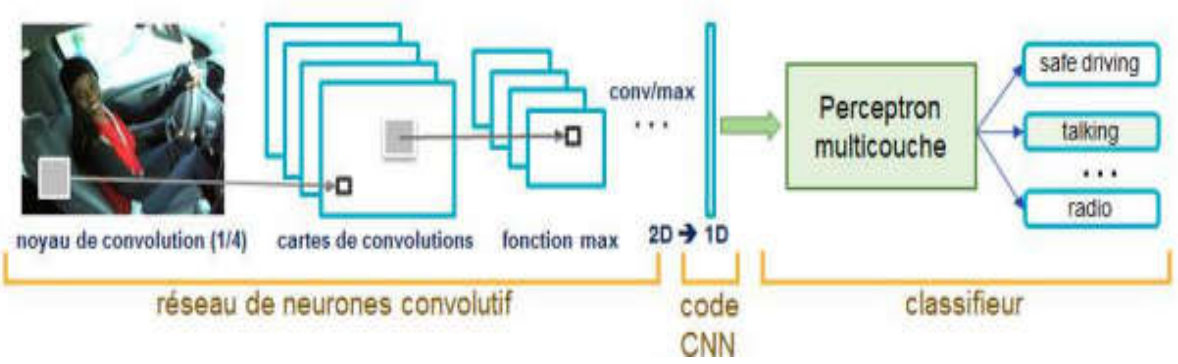


Figure 2.5-Les réseaux de neurones convolutifs **[**(Rock (1990))

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche page 10). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories**.** (Liou , Cheng et Liou 2014)

2.3. Auto-encodeur

2.3.1 Définition

Un auto-encodeur est un type de réseau de neurones artificiels utilisés pour apprendre des codages de données efficaces dans un non supervisé manière. Le but d'un auto-encodeur est d'apprendre une représentation (codage) pour un ensemble de données, typiquement pour la réduction de dimensionnalité , en apprenant au réseau à ignorer le «bruit» du signal.

Parallèlement au côté réduction, un côté reconstruction est appris, où l'auto-encodeur essaie de générer à partir du codage réduit une représentation aussi proche que possible de son entrée d'origine, d'où son nom. Des variantes existent, visant à forcer les représentations apprises à assumer des propriétés utiles. Des exemples sont des auto-encodeurs régularisés (*Sparse*, *Denoising* and *Contractive*), qui sont efficaces dans l'apprentissage des représentations pour les tâches de classification ultérieures et des auto-encodeurs *vibrationnels*, avec des applications en tant que modèles génératifs **.**Auto-encodeurs sont appliqués à de nombreux problèmes, de la reconnaissance faciale à l'acquisition de la signification sémantique des mots (Pascal Vincent 2010, p. 3371–3408).

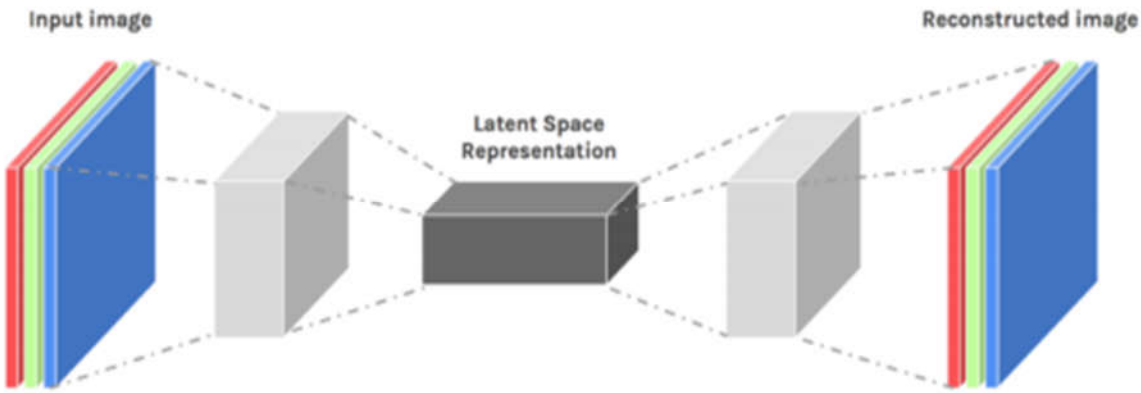


Figure2.6 : Architecture d'auto-encodeur (Pascal Vincent 2010, p. 3371–3408).

2.3.2 Composants de l'auto-encodeur

- **Encoder** modélise apprend comment réduire les dimensions d'entrée et de compresser les données d'entrées en une représentation codée. Ce qui induit de compresser l'entrée dans un espace latent de dimension généralement plus petite. h = f (x)
- **Goulot d'étranglement** qui est la couche qui contient la représentation compressée des données d'entrée. Il s'agit de la dimension la plus basse possible des données d'entrée.
- **Décodeur** dans lequel le modèle apprend à reconstruire les données à partir de la représentation codée pour être aussi proche que possible de l'entrée d'origine. Ce qui induit de reconstruire l'entrée à partir de l'espace latent. r = g (f (x)) avec r aussi proche que possible de x
- **Perte de reconstruction** c'est la méthode qui mesure les performances du décodeur et la proximité de la sortie par rapport à l'entrée d'origine.

L'apprentissage consiste ensuite à utiliser la rétro-propagation afin de minimiser la perte de reconstruction du réseau (Pascal Vincent 2010, p. 3371–3408).

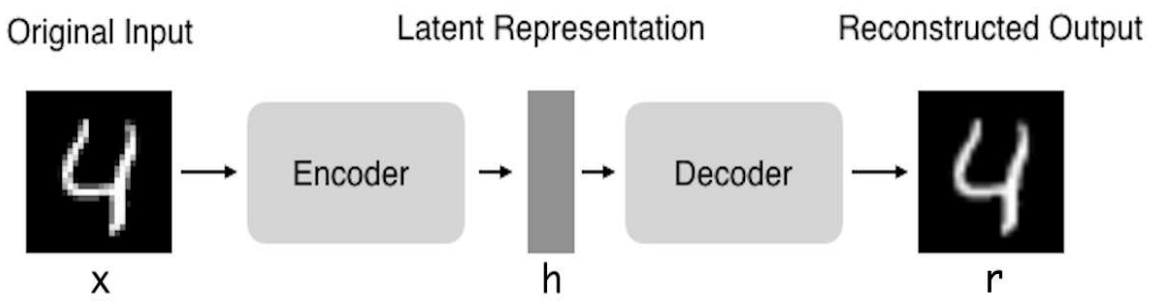


Figure 2.7 : Architecture auto-encodeur (Pascal Vincent 2010, p. 3371–3408).

2.3.3 Types d'auto-encodeurs

2.3.3.1 Auto encodeurs convolutifs

Les auto-encodeurs convolutifs (CAE) apprennent à encoder l'entrée dans un ensemble de signaux simples, puis à reconstruire l'entrée à partir d'eux. De plus, nous pouvons modifier la géométrie ou générer la réflectance de l'image en utilisant CAE**.** Dans ce type d'auto-encodeur, les couches de codeur sont appelées couches de convolution et les couches de décodeur sont également appelées couches de déconvolution. Le côté déconvolution est également connu sous le nom de suréchantillonnage ou de transposition de convolution (Alireza Makhzani et Brendan Frey 2013)

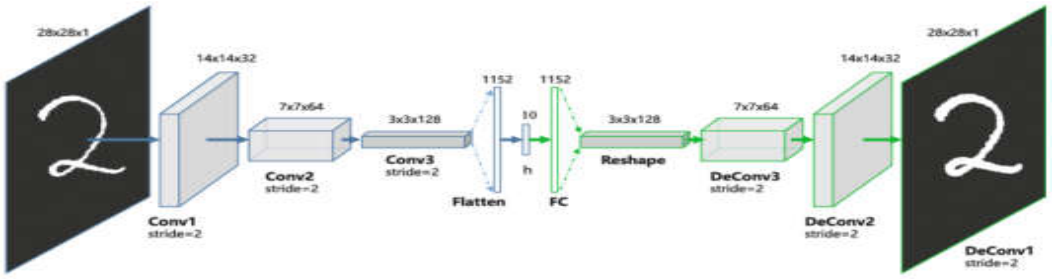


Figure2.8 : Architecture de convolution al auto-encodeur

2.3.3.2 Auto-encodeurs vibrationnels

Ce type d'auto-encodeur peut générer de nouvelles images tout comme les GAN. Les modèles d'auto-encodeur vibrationnels ont tendance à faire des hypothèses fortes liées à la distribution des variables latentes. Ils utilisent une approche variationnelle pour l'apprentissage de la représentation latente, qui se traduit par une composante de perte supplémentaire et un estimateur spécifique pour l'algorithme d'apprentissage appelé l'estimateur de Bayes variationnel à gradient stochastique. La distribution de probabilité du vecteur latent d'un auto-encodeur variationnel correspond généralement aux données d'apprentissage beaucoup plus étroitement qu'un auto-encodeur standard. Comme les VAE sont beaucoup plus flexibles et personnalisables dans leur comportement de génération que les GAN, ils conviennent à la génération d'art de tout type.Figure 25 Architecture auto-encodeur variationnel (Alireza Makhzani et Brendan Frey 2013)

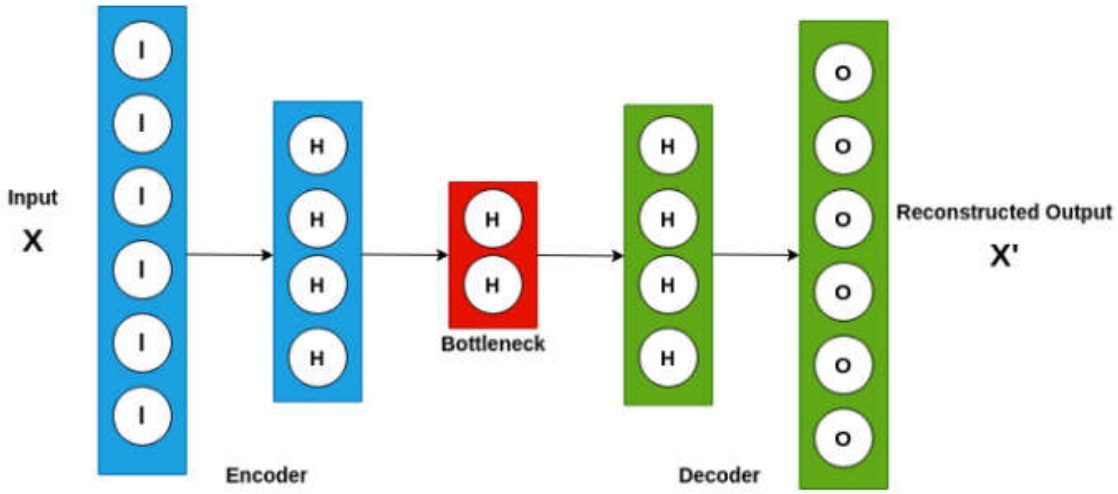


Figure2.9 : Architecture auto-encodeur variationnel **.**

2.3.3.3 Auto-encodeurs de dé bruitage

Les auto-encodeurs de débruitage ajoutent du bruit à l'image d'entrée et apprennent à le supprimer. Évitant ainsi de copier l'entrée vers la sortie sans apprendre les fonctionnalités sur les données. Ces auto-encodeurs prennent une entrée partiellement corrompue pendant l'entraînement pour récupérer l'entrée d'origine non déformée. Le modèle apprend un champ vectoriel pour mapper les données d'entrée vers une variété de dimension inférieure qui décrit les données naturelles pour annuler le bruit ajouté. Par ce moyen, l'encodeur extraira les caractéristiques les plus importantes et apprendra une représentation plus robuste de la donnée (H. L.-A. Pascal Vincent 2010)

2.3.3.4 Auto-encodeurs profond

Un auto-encodeur profond est composé de deux réseaux symétriques de croyance profonde ayant quatre à cinq couches peu profondes. L'un des réseaux représente la moitié de codage du réseau et le second réseau constitue la moitié de décodage. Ils ont plus de couches qu'un simple encodeur automatique et sont donc capables d'apprendre des fonctionnalités plus complexes. Les couches sont des machines de Boltzmann restreintes, les éléments constitutifs des réseaux de croyances profondes**.** (H. L.-A. Pascal Vincent 2010)

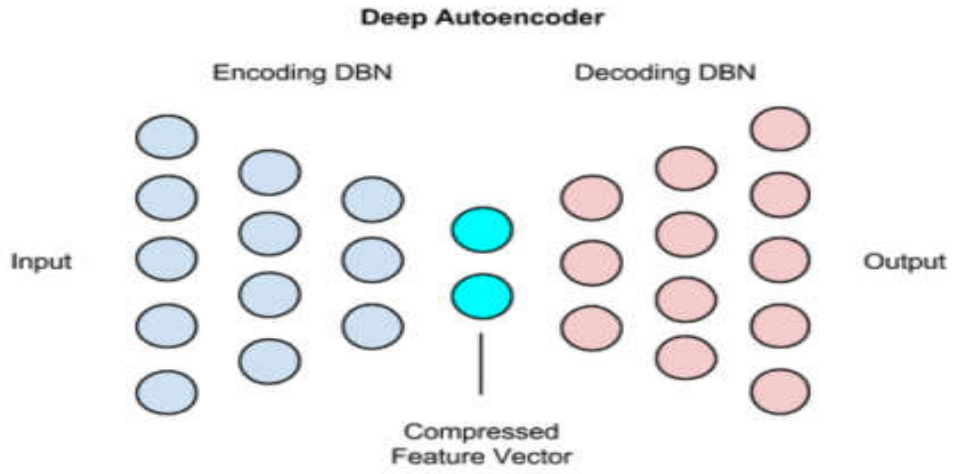


Figure2.10 : Architecture auto-encodeur profond (H. L.-A. Pascal Vincent 2010)

2.4 Application des auto-encodeurs

2.4.1 Bruitage d'image

Les auto-encodeurs sont très bons pour réduire le bruit des images. Lorsqu'une image est corrompue ou qu'elle contient un peu de bruit, nous appelons cette image une image bruyante.

Pour obtenir des informations appropriées sur le contenu de l'image, nous effectuons un dé bruitage de l'image. (H. L.-A. Pascal Vincent 2010)

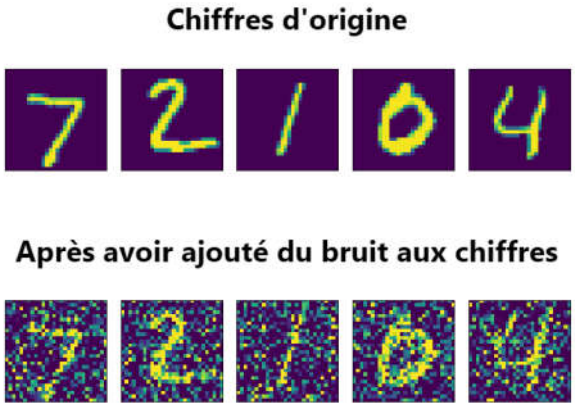


Figure2.11 : Bruitage d'image (H. L.-A. Pascal Vincent 2010)

2.4.2 Réduction de dimensionnalité

Les auto-encodeurs convertissent l'entrée en une représentation réduite qui est stockée dans la couche intermédiaire appelée code. C'est là que les informations de l'entrée ont été compressées et en extrayant cette couche du modèle, chaque nœud peut maintenant être traité comme une

variable. Ainsi, nous pouvons conclure qu'en éliminant la partie décodeur, un auto-encodeur peut être utilisé pour la réduction de dimensionnalité, la sortie étant la couche de code**.** (Alireza Makhzani et Brendan Frey 2013)

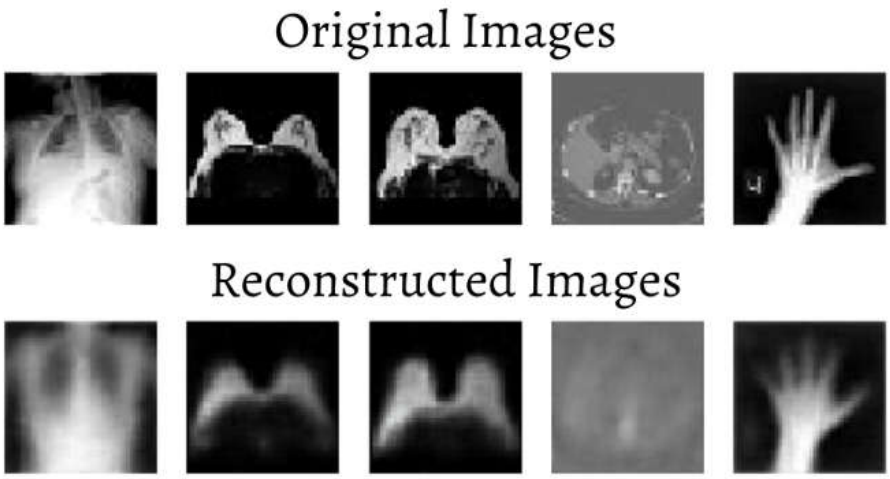


Figure2.12 : Réduction de dimensionnalité (H. L.-A. Pascal Vincent 2010)

2.4.3 Extraction de caractéristiques

L'encodage d'une partie des auto-encodeurs permet d'apprendre les fonctionnalités cachées importantes présentes dans les données d'entrée, dans le processus de réduction de l'erreur de reconstruction. Lors de l'encodage, un nouvel ensemble de combinaisons d'entités d'origine est généré**.** (H. L.-A. Pascal Vincent 2010)

Figure2.13 : Image Extraction de caractéristiques (H. L.-A. Pascal Vincent 2010)

2.4.4 Génération d'image

L'auto encodeur variationnel (VAE) décrit ci-dessus est un modèle génératif, utilisé pour générer des images qui n'ont pas encore été vues par le modèle

. L'idée est que, étant donné les images d'entrée comme des images de visage ou de paysage, le système générera des images similaires. L'utilité est de :

- Générer de nouveaux personnages d'animation
- Générer de fausses images humaines (H. L.-A. Pascal Vincent 2010)

Figure2.14 : Génération d'image (H. L.-A. Pascal Vincent 2010)

2.4.5 Coloration de l'image

L'une des applications des auto-encodeurs est de convertir une image en noir et blanc en une image en couleur. Ou nous pouvons convertir une image colorée en une image en niveaux de gris**.** (H. L.-A. Pascal Vincent 2010)

Figure2.14 : Coloration de l'image (H. L.-A. Pascal Vincent 2010)

2.5 Le fonctionnement des auto-encodeurs

Un **Auto-encodeur** est **un réseau de neurones artificiels** qui est souvent utilisé dans l'apprentissage des caractéristiques discriminantes d'un ensemble de données. Il peut être vu comme l'ensemble d'un encodeur et décodeur

En effet, l'encodeur est constitué par un ensemble de couches de neurones, qui traitent les données afin d'obtenir une nouvelle représentation des données tandis que les couches de neurones du décodeur analysent les données encodées pour essayer de reconstruire les données d'origines. Généralement, la nouvelle représentation des données à moins de caractéristiques.

Ce qui permet de réduire la dimensionnalité des données d'origines. La différence entre les données d'origines et les données reconstruites par le décodeur permet d'évaluer l'erreur de reconstruction. Le but de l'entrainement de l'auto-encodeur est de modifier les paramètres afin de minimiser l'erreur de reconstruction (X Liu 2019)

Il ne nécessite pas de données labélisées, et peut ainsi être utilisé pour réduire l'effort de labélisation des données. L'auto-encodeur est donc classé comme une technique d'apprentissage non supervisé. La couche la plus importante est la couche encodée, qui permet d'avoir une nouvelle représentation des données.

Le nombre de neurones dans les couches cachées doit être inférieur à celui des couches d'entrées pour permettre aux couches cachées à apprendre plus de modèles de données et à ignorer les « bruits ». Si le nombre de neurones dans les couches cachées est supérieur à celui des couches d'entrée, le réseau neuronal aura trop de capacité pour apprendre des données. Dans un cas extrême, il pourrait simplement copier l'entrée dans les valeurs de sortie, y compris les bruits, sans extraire aucune information essentielle (X Liu 2019)

2.5.1 L'architecture d'un auto-encodeur

L'image de la figure ci-dessous montre l'architecture d'un auto-encodeur. Elle est constituée des couches de neurones de codage et de décodage. Les couches de codage compriment les données d'origines d'entrée pour permettre d'avoir une représentation comprimée des données. Tandis que Les couches de décodage essayent de reconstruire les données d'origines à partie des données comprimées

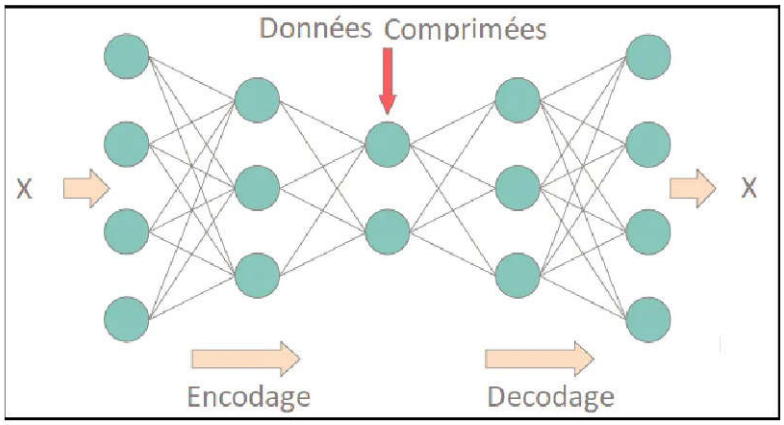


Figure 16 : Architecture général d'auto-encodeur (X Liu 2019)

2.5.2 Les avantages d'auto-encodeurs pour la réduction des dimensions

Bien que l'analyse en composantes principales (ACP) permette de réduire les dimensions, il faut souligner que l'ACP utilise l'algèbre linéaire pour se transformer. En revanche, les techniques d'auto-encodeurs peuvent effectuer des transformations non linéaires avec leur fonction d'activation non linéaire et leurs couches multiples **[66].**Il est plus efficace de former plusieurs couches avec un auto-encodeur, plutôt que de former une énorme transformation avec l'ACP. Les techniques d'auto-encodeur montrent donc leurs avantages lorsque les données sont de natures complexes et non linéaires. De plus, un article jalon de Geoffrey Hinton (2006) a montré qu'un auto-encodeur entraîné produise une erreur plus petite par rapport aux 30 premiers composants principaux d'un ACP et une meilleure séparation des grappes (SS Roy 2018)

2.6 Conclusion

Dans ce chapitre, nous avons introduit les notions sur l'apprentissage et ses applications dans le cadre de l'image et la vision par ordinateur. Nous avons donné une vision générale sur les méthodes et les approches de la classification (supervisées non supervisées etc…). Nous avons aussi présenté quelques méthodes de classification Réseaux de neurones et auto-encodeur et ses applications. Notre choix s'est fixé sur l'utilisation des auto-encodeurs pour la segmentation d'image dans un premier temps, puis pour son utilisation pour le traitement des occultations pour le traitement de l'interaction de la main dans le cadre de la réalité augmentée. Dans le prochain chapitre, nous allons voir le principe de l`apprentissage profond (Deep Learning) et auto-encodeur appliqué à la segmentation d'images et à la gestion du problème de l'occultation.

Chapitre 3.
Technique basée deep learning pour la RA

3.1 Introduction

Comme nous l'avons indiqué dans le chapitre précédent, nous avons choisi d'implémenter une technique à base de l'apprentissage profond, utilisant un auto-encoder pour la segmentation d'images et la gestion du problème de l'occultation dans la réalité augmentée. Notre choix est motivé par le fait que celle-ci offre des résultats convaincants lorsqu'il s'agit d'images proches de celles existantes dans la base d'apprentissage et que la technique est facile à implémenter.

Notre objectif est donc de mettre en œuvre cette technique, nous avons aussi choisi une application très courante dans la réalité augmentée : la gestion de l'interaction de la main (scène réelle) avec des objets virtuels.

Nous allons donc détailler dans ce chapitre, les principaux outils utilisés dans la mise œuvrent de cette technique. Nous allons aussi expliquer les étapes et les composants de la technique proposée.

3.2 Base de données (Data set) utilisée

La base de données utilisée est un sous-ensemble de 3200 images des mains et de 1200 images des mains étiquetées de taille 512*512.

Notre application basée sur 2 modèles principaux (segmentation et l'occultation).

La base de données de modèle segmentation contient quatre fichiers :

Data test All : contient plus que 600 d'images des mains avec le fond (arrière-plan) contenant des vues de notre université. Cette base de données est utilisée comme une entrée du module de segmentation.

Data Result All : contient également plus que 600 images des mains sans fond étiquetées. Cette base de données est utilisée pour l'entrainement de modèle.

SegmentationResults : Cette base de données contient le résultat qui est un ensemble d'images des mains segmentées après l'entrainement du modèle.

SegmentationResults diff : Cette base de données est le résultat de la comparaison entre les images de segmentation et les images étiquetées.

La base de données de modèle segmentation contient trois fichiers de base donnée :

Data testMaskAll : Cette base de données contient plus que 600 images du masque des mains avec le téléphone mobile utilisées comme entrée du modèle d'occultation.

Data testObjectAll : Cette base de données contient également plus que 600 images des mains avec le fond et avec le téléphone mobile qui est l'objet utilisé pour la gestion de l'occultation.

Data AR occlusion : Cette base de données contient le résultat du traitement de l'occultation des images des mains avec le téléphone mobile

Segmentation par auto-encoder

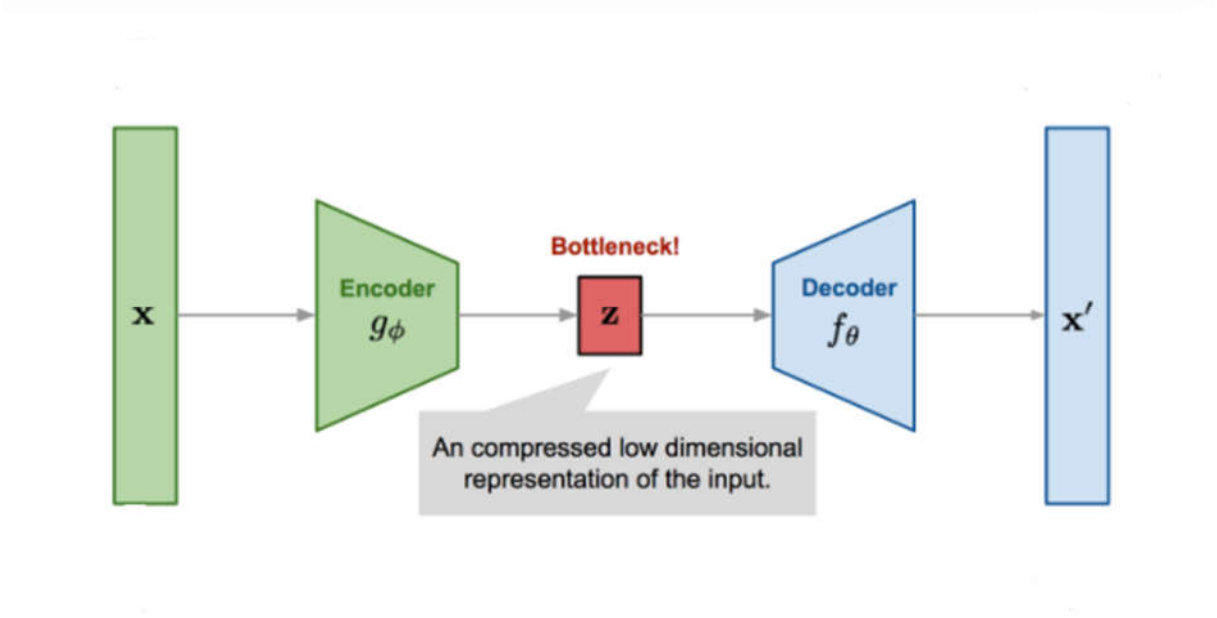


Figure3.1 : Architecture général d'auto-encodeur (Alireza Makhzani et Brendan Frey 2013)

Auto-encodeur sont à ce jour les modèles les plus performants pour classer des images. Désignés par auto-encodeur, de l'anglais auto-encodeur, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un auto-encodeur est la partie encodeur. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions.

Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé flaten. Après cette transformation, on passe à la deuxième partie : décoder Elle est appliquée sur caractéristiques des images Une image est passée à travers une succession de filtres, ou noyaux de convolution.

Notre code contient 10 convolutions pour extraire des caractéristiques.

En entrée, une image est fournie sous la forme d'une matrice de pixels elle 2 dimensions. Elle passe par 5 convolutions. On va expliquer les actions que va réaliser dans chaque convolution en général, ensuite on va détailler chaque action sur notre code.

Sequential conv2D GroupNorm LeakyRelu

Nn.sequentiel

Un conteneur séquentiel. Les modules y seront ajoutés dans l'ordre où ils sont passés dans le constructeur. Alternativement, un dictionnaire ordonné de modules peut également être transmis.

Conv2D

Applique une convolution 2D sur un signal d'entrée composé de plusieurs plans d'entrée.

Paramètres

in_channels (Int) – Nombre de canaux dans l'image d'entrée

Out_channels (Int) – Nombre de canaux produits par la convolution

Kernel_size (Int ou tuple) – Taille du noyau convolutif

Padding (Int ou tuple, facultatif) - Zéro-remplissage ajouté des deux côtés de l'entrée. Par défaut : 0

Stride contrôle la foulée pour la corrélation croisée, un seul nombre ou un tuple.

Padding contrôle la quantité de remplissage implicite des deux côtés pour le paddingnombre de points pour chaque dimension.

Groupnorm

torch.nn.GroupNorm (num_groups, num_channels, eps=1e-05, affine=True)

Applique la normalisation de groupe sur un mini-lot d'entrées comme décrit dans l'article sur la normalisation de groupe

y = \frac{x - \mathrm{E}[x]} { \sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta

Les canaux d'entrée sont séparés en num_groupsgroupes, chacun contenant des canaux. La moyenne et l'écart-type sont calculés séparément sur chaque groupe.num_channels / num_groups\gammaγ et \bêtaβ sont des vecteurs de paramètres de transformation affine par canal pouvant être appris de taille num_channelssi affineest True. L'écart-type est calculé via l'estimateur biaisé, équivalent à torch.var(input, unbiased=False) .

Cette couche utilise des statistiques calculées à partir des données d'entrée dans les modes d'apprentissage et d'évaluation.

Paramètres

Num_groups (int) - nombre de groupes dans lesquels séparer les canaux

Num_channels (Int) - nombre de canaux attendus en entrée

LEAKYRELU

CLASSERtorch.nn.LeakyReLU(negative_slope=0.01 , inplace=False)[LA SOURCE]

On applique la fonction élément par élément :

LeakyReLU (x) = max(0, x) + pente_négative * min(0, x).

Paramètres

- **negative_slope –** Contrôle l'angle de la pente négative**. Par défaut : 1e-2**
- **inplace -** peut éventuellement effectuer l'opération sur place. **Défaut : False**

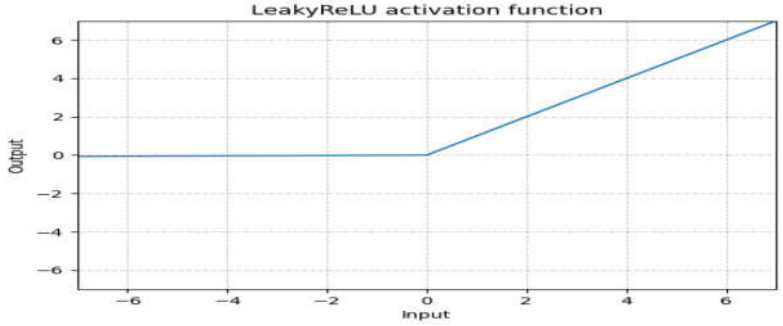


Figure3.2 : Courbe fonction d'activation [2]

3.3 Architecture de segmentation

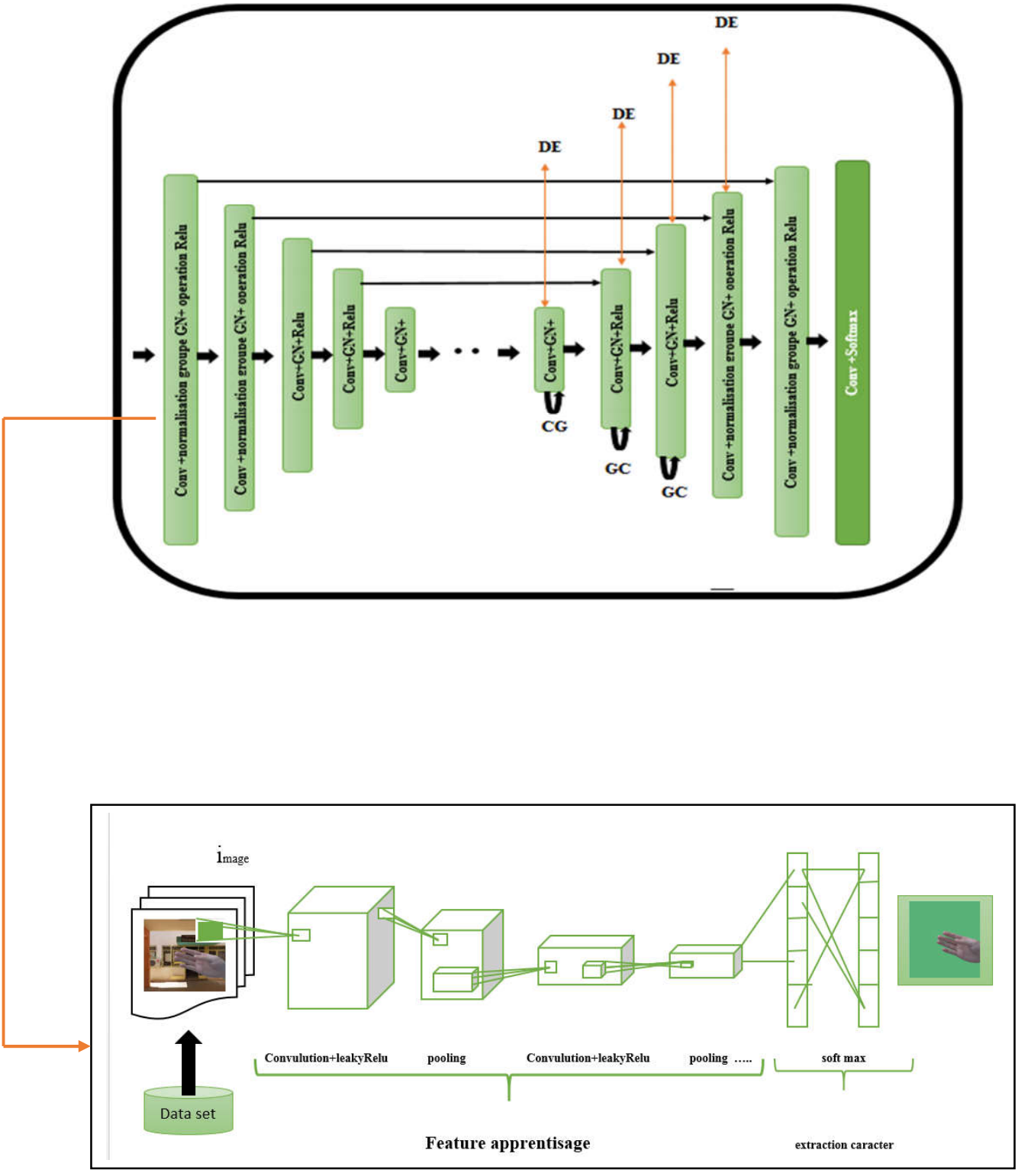


Figure3.3 : Architecture détaillée de segmentation.

3.3.1 Couche convolutive

Dans ce programme, on applique dix convolutions.

La convolution est la première couche à extraire les caractéristiques d'une image d'entrée.

La convolution préserve la relation entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés de données d'entrée.

C'est une opération mathématique qui prend deux entrées telles qu'une matrice d'image et un filtre ou noyau.

Une matrice image (volume) de dimension 64

Un filtre (32)

Ce filtre est convolé (diapositives) sur la largeur et la hauteur du fichier d'entrée, et un produit scalaire est calculé pour donner une carte d'activation

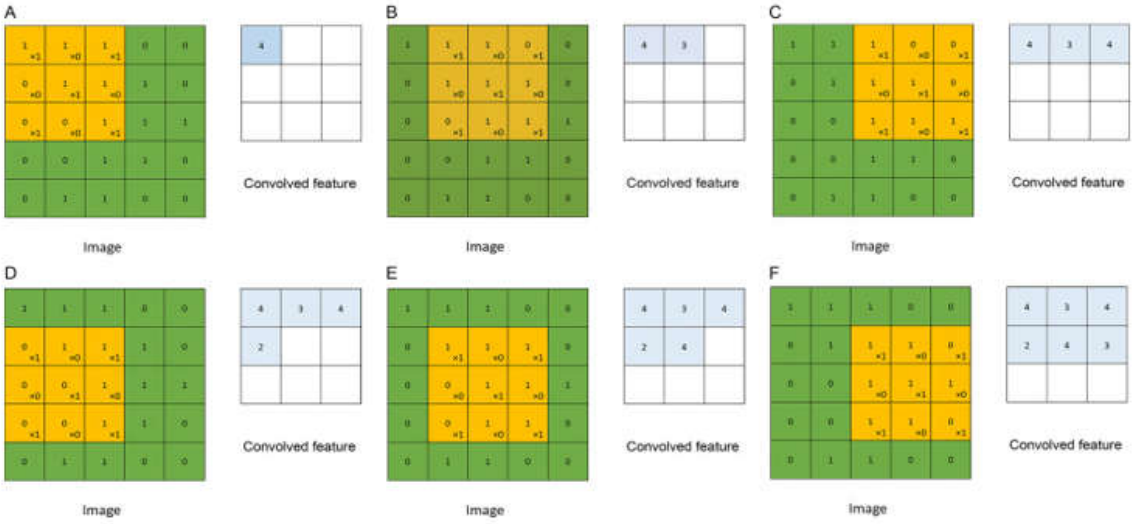


Figure3.3 : Matrice de filtre de Couche convolutive **[48]**

3.3.2 Stride

Est le nombre de décalages de pixels sur la matrice d'entrée. Lorsque stride est de 1, nous déplaçons les filtres sur 1 pixel à la fois

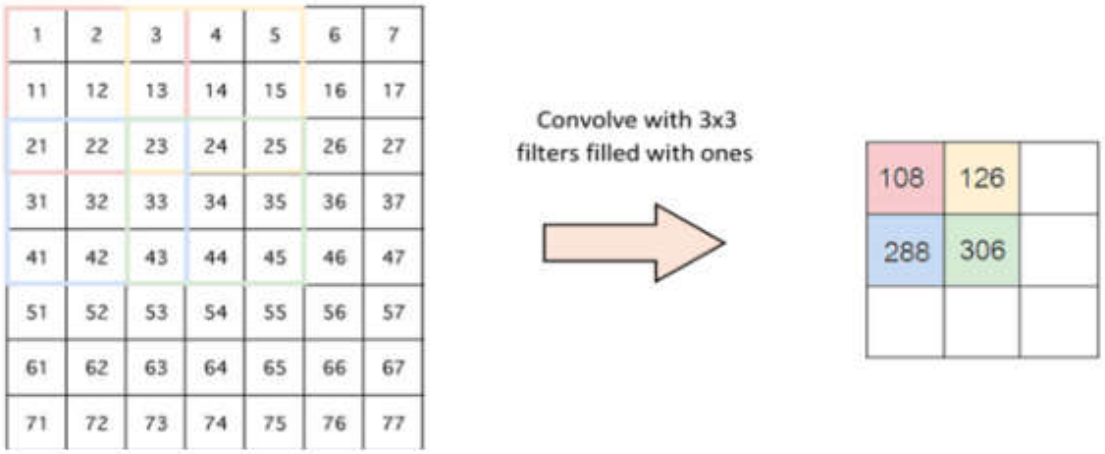


Figure3.4 : Matrice de stride [69]

3.3.3 Pading

Parfois, le filtre ne s'adapte pas parfaitement à l'image d'entrée. Nous avons deux options :

- Complétez l'image avec des zéros (zéro-remplissage) pour qu'elle s'adapte
- Déposez la partie de l'image où le filtre ne tenait pas. C'est ce qu'on appelle un remplissage valide qui ne conserve qu'une partie valide de l'image.

Un rembourrage est ajouté au cadre de l'image pour laisser plus d'espace au noyau pour couvrir l'image. L'ajout de remplissage à une image traitée par un CNN permet une analyse plus précise des images

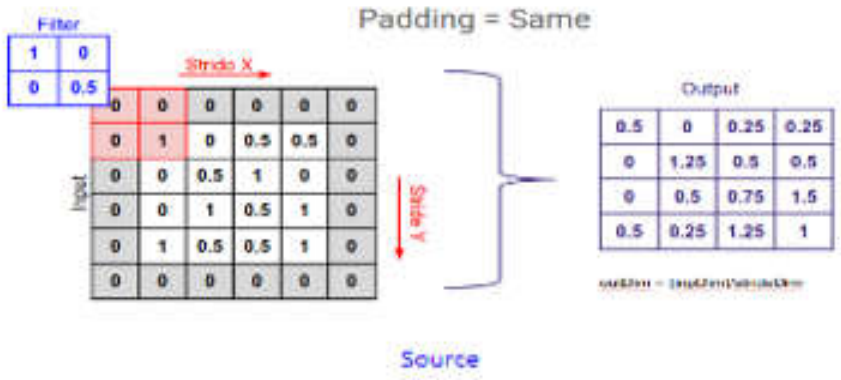


Figure3.4 : Matrice de pading.

Après chaque convolution on fait un pooling.

3.3.4 Pooling

Semblable à la couche convolutive, la couche de regroupement est responsable de la réduction de la taille spatiale de l'entité convoluée.

Il s'agit de réduire la puissance de calcul requise pour traiter les données grâce à la réduction de la dimensionnalité. Il existe trois types de couche de pooling :

Regroupement maximal : renvoie la valeur maximale de la partie de l'image couverte par le noyau.

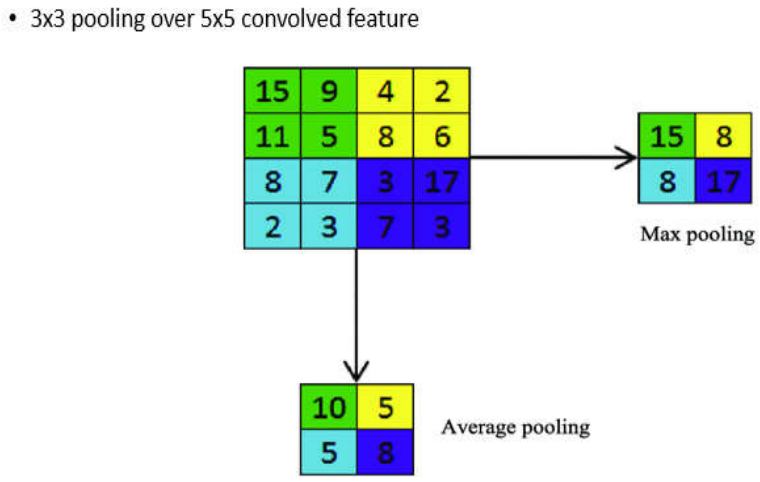


Figure3.5 : Matrice de pooling [71

La sortie de cette opération sera l'entrée de deuxième convolution et elle va se répéter lors de toutes les opérations précédentes mais avec un changement de paramètre au tableau ci-dessous :

Convolution	Conv2d					Group -Norm		Leaky-rellu	
	Chanel-in	Chanel-out	Kernel size	stride	padding	Num-grp	Num-chanel	Negative-slope	implace
Convolution 1	3	32	3	1	1	32	32	0.2	True
Convolution 2	32	64	3	2	1	32	32	0.2	True
Convolution 3	64	128	3	2	1	32	64	0.2	True
Convolution 4	128	256	3	2	1	32	128	0.2	True
Convolution 5	256	256	3	2	2	32	256	0.2	True
Convolution 6	256	256	3	4	4	32	256	0.2	True
Convolution 7	256	256	3	2	2	32	256	0.2	True
Convolution 8	256	128	3	1	1	32	128	/	/
Convolution 9	128	64	3	1	1	32	64	/	/
Convolution10	64	32	3	1	1	32	32	/	/

Tableau 3.1 : Tableau changement de paramètre.

A la sortie des couches précédentes pour les transformer en un seul vecteur peut être utilisée comme entrée pour la couche suivante.

La raison pour laquelle nous faisons cela est que nous devrons insérer ces données dans un réseau de neurones artificiels plus tard.

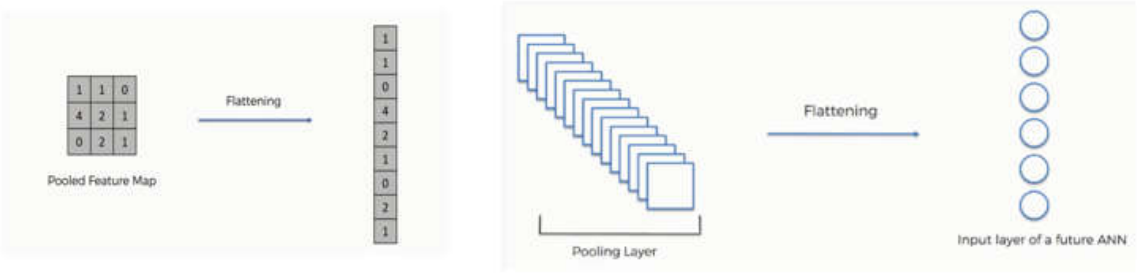


Figure3.6 : Explication de sorties de convulutional [2]

Activation des fonctions

Le but de la fonction d'activation est d'introduire une non-linéarité dans la sortie d'un neurone.

Relu signifie Rectified Linear Unit pour un fonctionnement non linéaire. La sortie est

ƒ(x) = max (0 .1x, x)

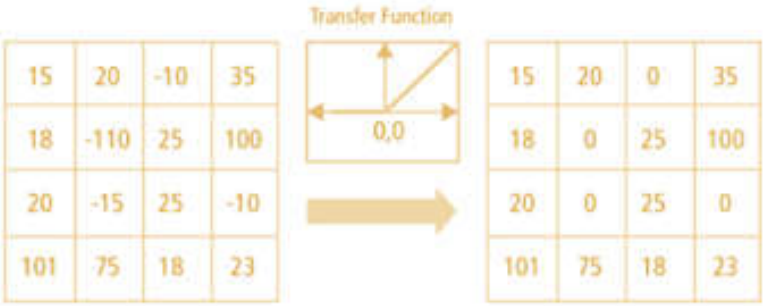


Figure3.7 : Matrice d'activation fonctions [68]

3.3.5 Fonction d'activation : Softmax

On expose chaque élément de la couche de sortie et on additionne les résultats (environ 181,73 dans ce cas)

On prend chaque élément de la couche de sortie, on l'expose et on le divise par la somme obtenue à l'étape 1 (exp (1.3) / 181,37 = 3,67 / 181,37 = 0,02

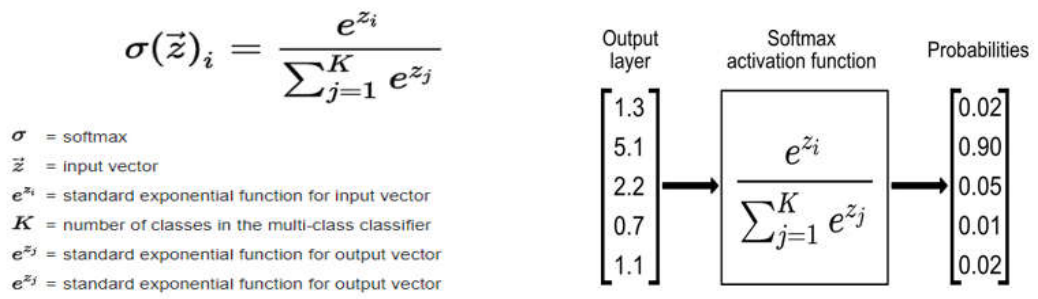


Figure3.8 : Matrice d'activation fonctions.

3.3.6 DenseNet

L'idée derrière les réseaux convolutifs denses est simple : il peut être utile de référencer des cartes de caractéristiques plus tôt dans le réseau.

La carte des caractéristiques de chaque couche est concaténée à l'entrée de chaque couche successive au sein d'un bloc dense.

Un bloc dense est un module utilisé dans les réseaux de neurones convolutifs qui connecte toutes les couches (avec des tailles de carte de caractéristiques correspondantes) directement les unes aux autres.

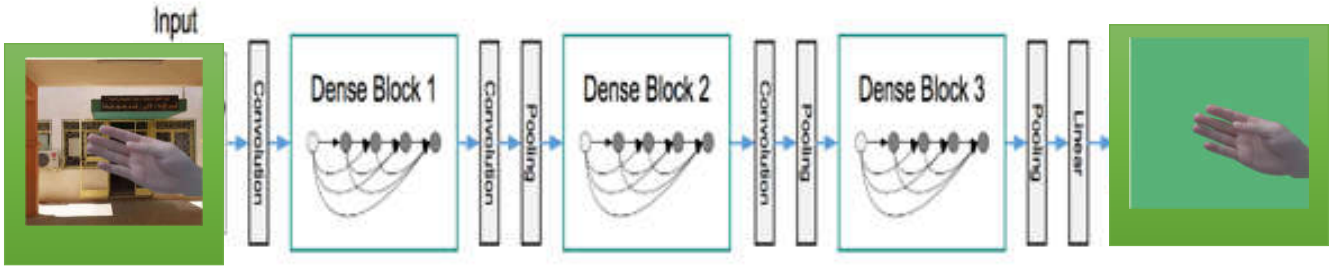


Figure3.8 : Bloc densenet.

3.4 Tests

Pour le test, nous avons choisi des images des mains qui font parties de la base de données segmentationResults, et qui sont des images du même genre mais qui ne font pas parties de la base de données DataResultAll images des mains mais qui rassemble aux images segmentées des mains. Le réseau formé de la comparaison entre deux images. Pour réaliser l'exécution du programme il faut d'abord charger la base de données segmentationResults et DataResultAll, ensuite choisir et redimensionner l'image couleur à la taille 320*320.

C'est cette taille qui est utilisée dans la couche d'entrée. Il faut ensuite comparer entre eux. A la fin obtenir le résultat de différence.

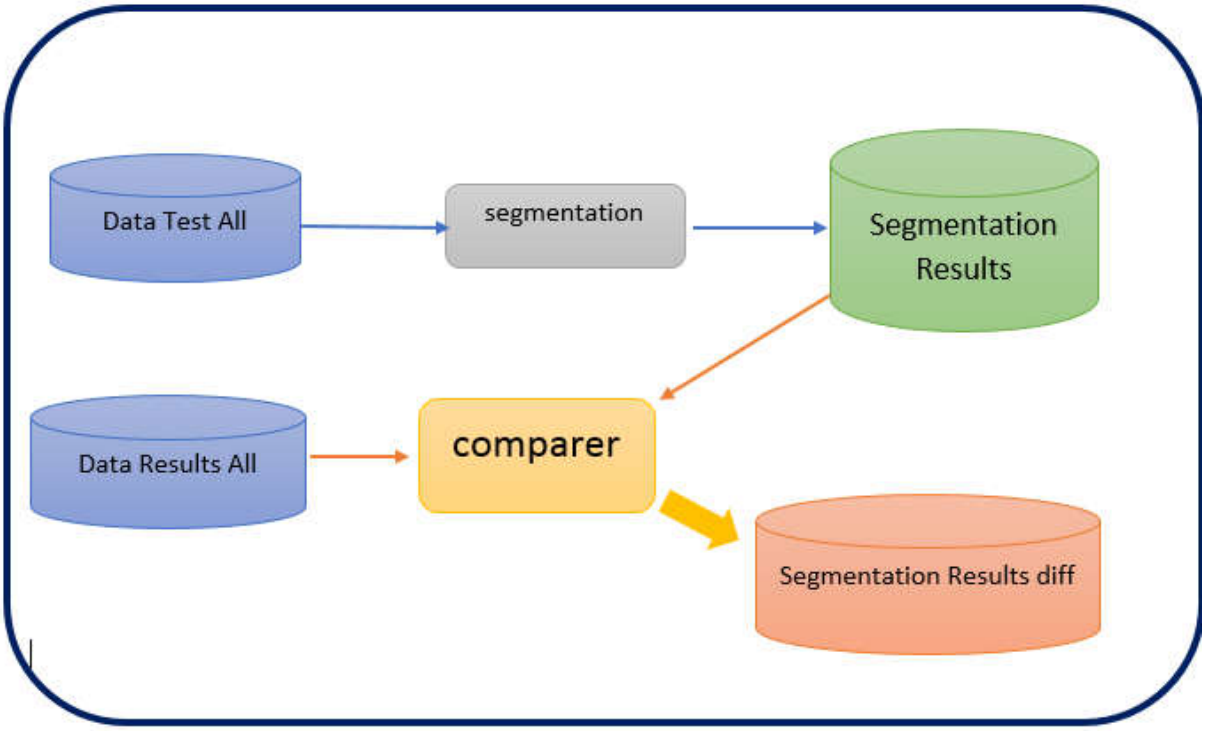


Figure3.9 : Schéma conception de tes

3.5 Architecture d'occultation

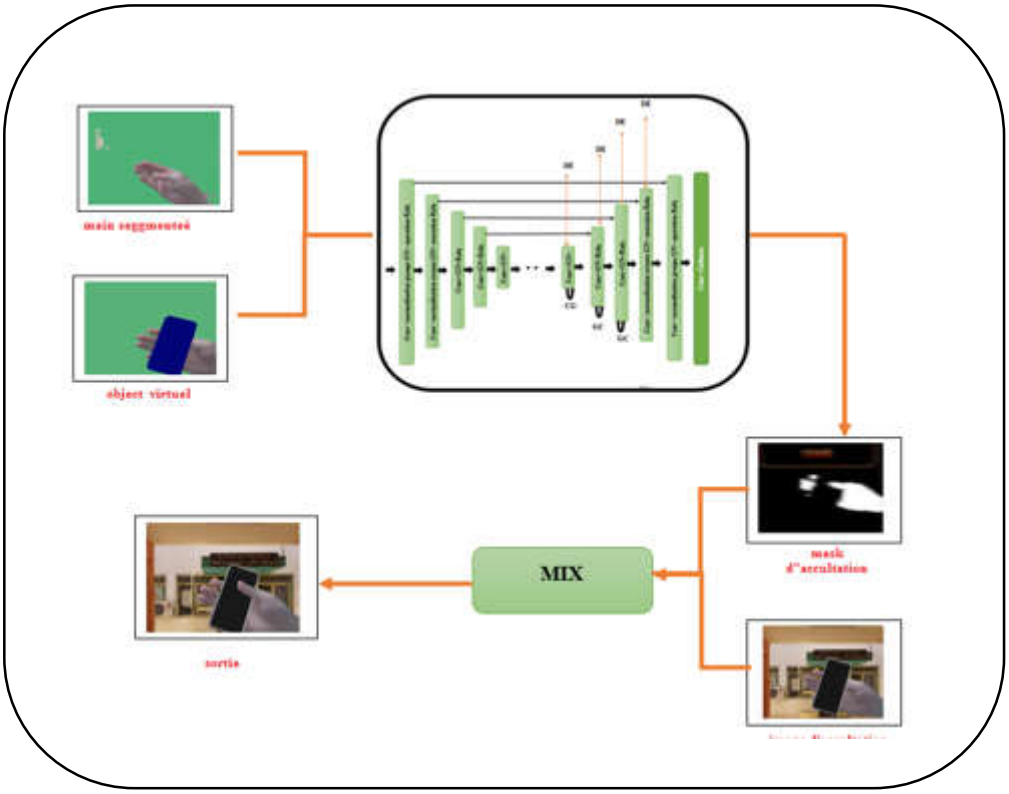


Figure3.9 : Schéma conception d'occultation

Cette phase dépend sur l'apprentissage, du traitement de l'occultation et des bases sur auto-encodeur. Dans cette phase l'entrée est une image d'occultation et la sortie est une image sans occultation.

2. une **normalisation de groupe** (GN)

3. une opération **ReLU** qui fuit.

on a modules de contexte global **(GC)**

Dans les trois premiers blocs du décodeur pour agréger les informations globales et concevoir le modules d'amélioration detailles **(DE)** pour récolter des informations détaillées dans fonctionnalités de bas niveau avec sauts de connexions.

De plus, nous tirons parti d'une autre opération convolutive et **softmax** pour générer le sortie réseau **(masque d'occultation)** et on utilise une supervision approfondie pour calculer la perte et prédire un masque d'occlusion par couche dans le décodeur. Au final, on prend le masque d'occultation prédit à partir de la dernière couche comme sortie réseau finale.

3.6 Conclusion

Dans ce chapitre, nous avons présenté notre méthode proposée, où nous avons présenté la conception de nos deux modèles segmentation main par auto-encodeur et traitement de l'occultation nous avons détaillé les étapes que nous avons franchies pour arriver aux deux modèles, nous avons aussi détaillé les convolutions de l'auto-encodeur pour la segmentation de main et les convolutions essentielles. Enfin nous avons exposé également l'occultation et ses principales phases. Dans le prochain chapitre, nous présenterons la mise en œuvre des deux modèles et les résultats obtenus.

Chapitre 4.
Mise en œuvre et résultats et bilan

4.1 Introduction

Dans ce chapitre, nous présenterons l'environnement de travail, le langage de programmation et les outils que nous avons utilisés pour construire le système (Deep Learning pour le suivi des objets 3D réels dans la réalité augmentée).

Par la suite, nous expliquerons toutes les expériences que nous avons appliquées aux méthodes proposées et aux résultats obtenus.

4.2 Environnements et développement d'outils

Pour développer notre système, nous allons utiliser différents environnements et outils pour le backend langage de programmation comprenant les API, les bibliothèques pour le front-end également nous allons utiliser le langage de programmation, travaillé sur de nombreuses modifications

Pycharm community Edition 2020.3.5

PyCharm est l'un des IDE Python les plus populaires. Il y a une multitude de raisons à cela, y compris le fait qu'il est développé par JetBrains, le développeur derrière le populaire IntelliJ IDEA IDE qui est l'un des 3 grands IDE Java et le «IDE JavaScript le plus intelligent» WebStorm. Avoir le soutien pour le développement Web en tirant parti de Django est une autre raison crédible**.**

PyCharm prend en charge l'intégration d'une gamme d'outils. Ces outils vont de l'aide à l'amélioration de la productivité du code à la gestion des projets de science des données. Certains des outils d'intégration les plus essentiels disponibles pour PyCharm incluent:

Anaconda

Une distribution Python gratuite et open-source orientée vers l'informatique scientifique avec une gestion et un déploiement simplifié des packages.

Python langage

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes

Et d'un système de gestion d'exceptions. Il est ainsi similaire à Perl, Ruby, Schème, Small talk et Tcl.

Bibliothèque

Pour l'exécution de ce projet, nous avons importé quelques bibliothèques :

```
Import cv2 pour cette commande il faut installer open cv
Import numpy as nap ce commande installer pip installe numpy
From PIL import Image il faut installer pip Install Pillow
Import torch pip Install torch
Import transforms il faut Install pip Install transforme
Import Variable
Import Sys il faut Install pip installer os-Sys
```

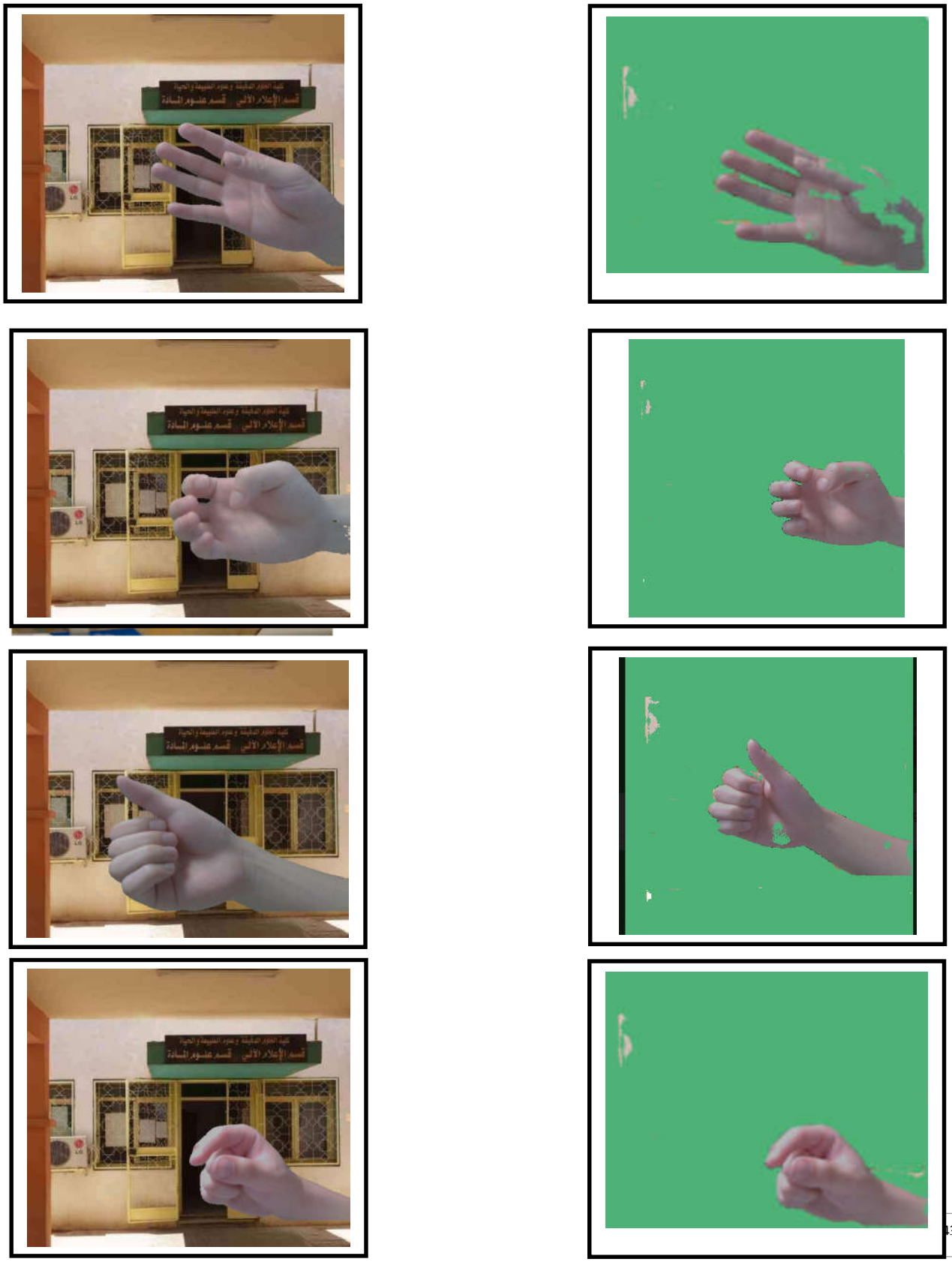
4.3 Segmentation

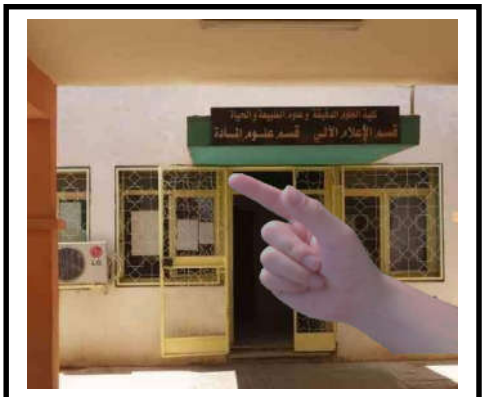
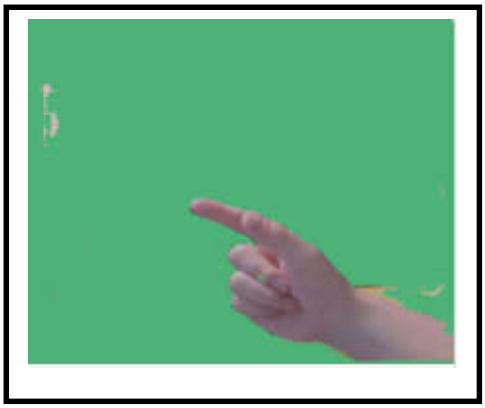
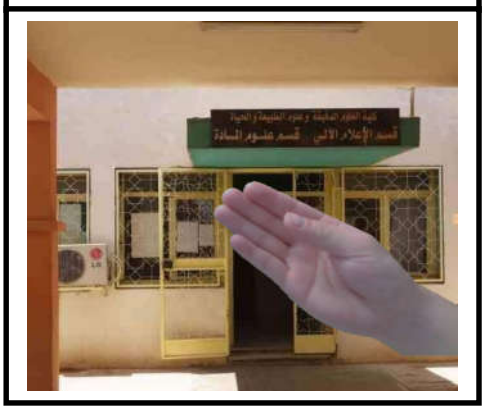
```
##################
  # Hand Segmetation Network
  ##################
  with torch.no_grad():
    res = handseg_net(hand_var)
  confidence = res[0].data.squeeze(0).cpu().numpy()

  hand_mask = np.argmax(confidence, axis=0)
  hand_mask = np.uint8(hand_mask)
  hand_mask[np.where(hand_mask == 1)] = 255

  hand_frame[np.where(hand_mask != 255)] = np.array
((119, 178, 78)).astype(np.uint8)
  cv2.imwrite('/content/drive/MyDrive/TestModelSegm
entation/GrabAR-
master/Image2/hand_segmentation.png', hand_frame)

  pil_hand_frame = opencv_to_pil(hand_frame)
  hand_var = Variable(img_transform(pil_hand_frame)
.unsqueeze(0)).to(device)
```

Comme vu ci-dessus, lorsque nous n'utilisons que l'opération de convolution et que nous répétons naïvement les pixels pour effectuer un sur-échantillonnage, les fonds générés sont peu

clairs et lisses. Cependant, nous pouvons observer des points aléatoires dans le fond généré. Mais en général ce réseau donne des bons résultats.

Image original **Image segmentée**



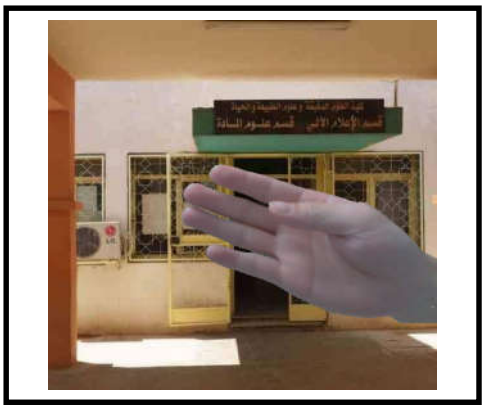



Figure 4.1 : Résultat de segmentation

4.4 Résultats de test

```
##################
    # Hand Segmetation  test result
    ##################

    # load the two input images
    # load image hand segmentation result
    hand_frame_net = cv2.imread('/content/drive/MyDrive/TestMod
elSegmentation/GrabAR-master/DataResults/'
     + file.split('_')[0] + '_' +file.split('_')[1] + '_hand.pn
g')

    # resize for faster processing
    resized_orig = cv2.resize(hand_frame_net, (320, 320))
    resized_mod = cv2.resize(hand_frame, (320, 320))
    #COLOR_BGR2GRAY nivue de gray
    gray_orig = cv2.cvtColor(resized_orig, cv2.COLOR_BGR2GRAY)
    gray_mod = cv2.cvtColor(resized_mod, cv2.COLOR_BGR2GRAY)

    (score, diff) = compare_ssim(gray_orig, gray_mod, full=True
)
    diff = (diff * 255).astype("uint8")

    cv2.imwrite('/content/drive/MyDrive/TestModelSegmentation/G
rabAR-master/SegmentationResultsdiff/'
     + file.split('_')[0] + '_' +file.split('_')[1] + '_hand_se
gmentation.png', diff)

    #score = score * 100
    print(" \n Correspondance des résultats entre segmentation
et Data Results : {:.2f} % ".format(score))
```

Nous faisons correspondre les deux images Image **segmentée** et image **de test** et nous comparons les pixels du tableau pour que nous mettions 0 dans les pixels identiques et 255 dans les différents pixels à la fin, nous obtenons l'image de la différence entre eux en noir et blanc, de sorte que le noir soit l'erreur et le blanc est les bons pixels. Et on obtient les résultats suivants.

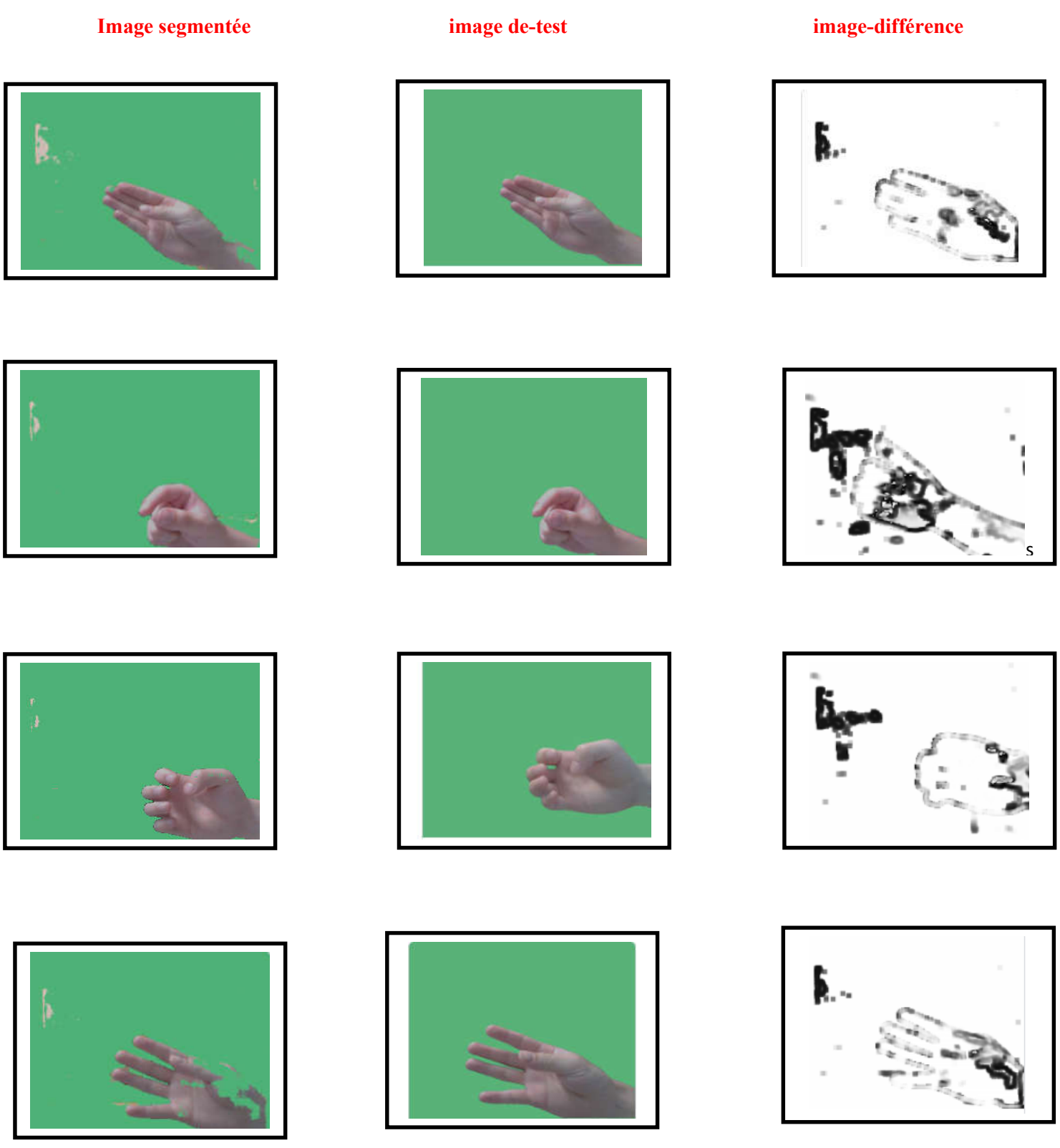


Figure 4.2 : Résultat de déférence.

4.5 Résultats d'occultation

```
##################
# Occlusion Estimation Network
##################
with torch.no_grad():
res, _ = net(hand_var, object_var, object_var, torch.tensor([1
]))
confidence = res[0].data.squeeze(0).cpu().numpy()

## mask ##
mask = np.argmax(confidence, axis=0)
mask = np.uint8(mask)
mask[np.where(mask == 1)] = 128  # object
mask[np.where(mask == 2)] = 255  # hand

mask = cv2.medianBlur(mask, 7)
ret, mask = cv2.threshold(mask, 129, 255, cv2.THRESH_BINARY)
kernel = np.ones((3, 3), np.uint8)
mask = cv2.erode(mask, kernel, iterations=1)

##################
# Generating final results based on the predicted mask
##################
object_frame[np.where(np.logical_and(mask==255, hand_mask==255
))] = \
hand_frame[np.where(np.logical_and(mask==255, hand_mask==255))
]
cv2.imwrite('/content/drive/MyDrive/TestModelSegmentation/Grab
AR-master/Image2/final_result.png', object_frame)
```

Dans cette partie nous travaillons sur une scène réelle de la main (avec fond) et nous avons choisi un téléphone portable comme un objet virtuel et à travers l'apprentissage profond, nous résolvons le problème d'occultation et nous obtenons les résultats suivants :

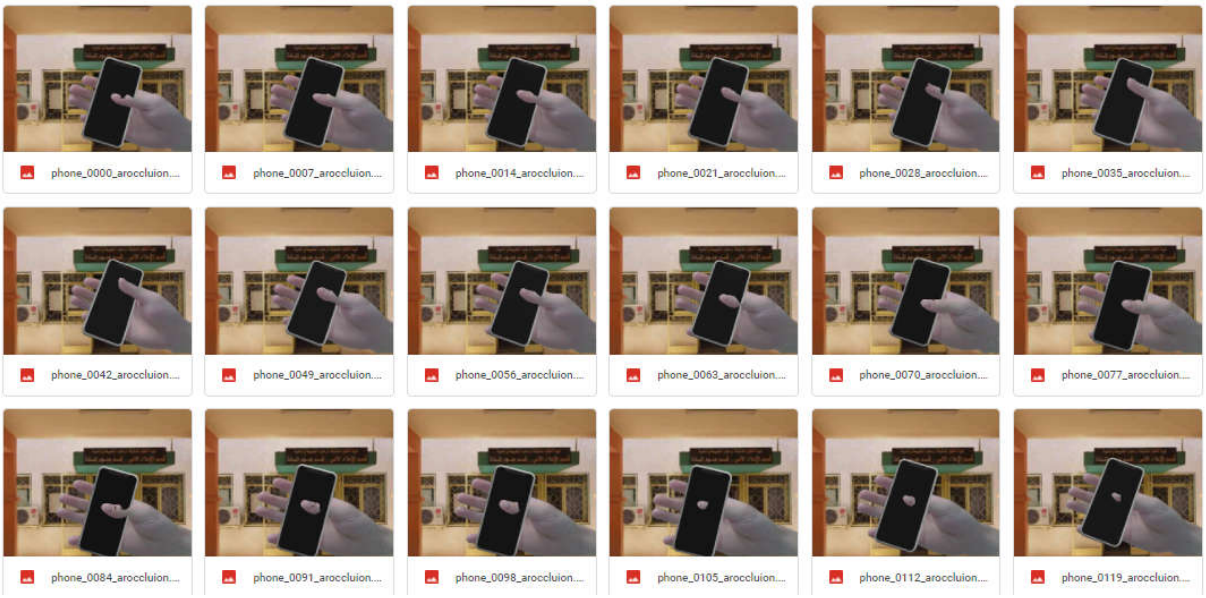


Figure 4.3 : Résultat d'occultation

4.6 Résultat de Test forme courbe

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

plt_score= np.zeros(3)

x = np.linspace(0, 10, 3)
score[0]= 2
score[1]= 10
score[2]= 20
fig = plt.figure()
plt.plot(x, score, '-')
fig.savefig('/content/drive/MyDrive/TestModelSegmentation/GrabAR-
master/Image/my_figure22.png')
```

4.6.1 Discussion des résultats :

Les résultats de notre application segmentation mains sont présentés sur la courbe précédente (figure 18) pour les méthodes de segmentation.

Nous avons essayé le test sur 1000 images et le taux de précision variait de 0 à 100, cette courbe représente le pourcentage de précision pour chaque image, nous avons obtenu un résultat de

98.52% comme meilleur résultat d'exécution nous remarquons que la méthode segmentation donne un bon résultat.

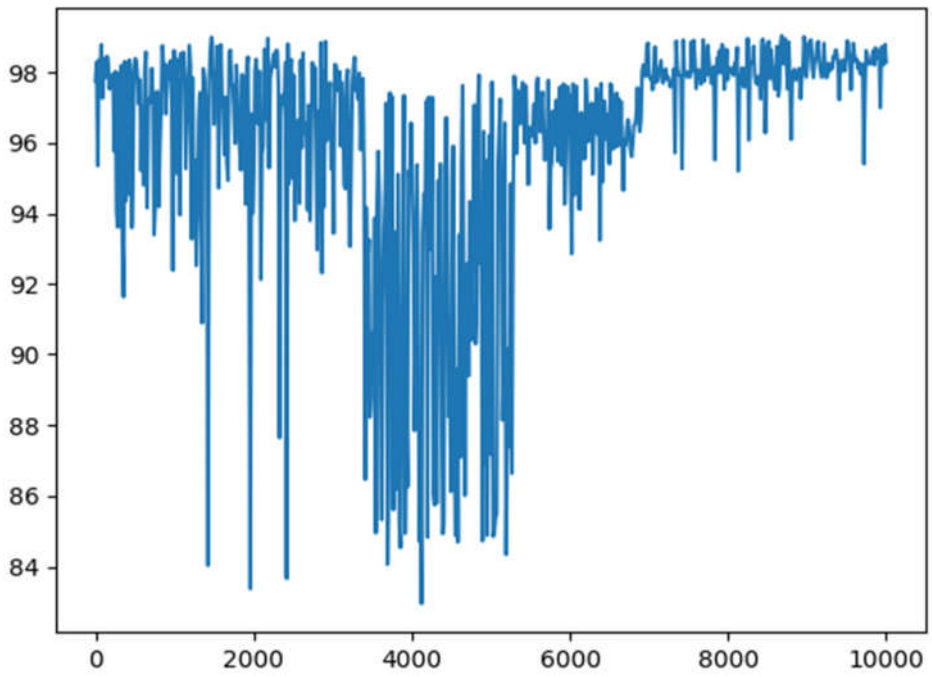


Figure 4.4 : Courbe de test**.**

4.4-Conclusion

Après avoir achevé notre conception nous avons donné les outils nécessaires pour la réalisation de notre travail. Nous avons présenté aussi l'environnement de développement. A la fin nous avons présenté notre application en donnant quelques captures d'écran qui expliquent le fonctionnement de notre travail, ainsi que les résultats obtenus. Nous avons donné un exemple de segmentation par notre application d'un ensemble d'images qui sont déjà étiquetées par modèle de segmentation et le résultat de test et également nous avons donné un exemple sur l'occultation

Conclusion générale

La réalité augmentée est devenue une destination pour les gens car elle leur offre une expérience agréable et interagir avec des objets virtuels. La réalité augmentée offre désormais à l'utilisateur une expérience réelle du produit et du service sans le coût de l'achat et aide à fournir un service sans preuve.

En raison de la prolifération des téléphones portables, la réalité augmentée reçoit beaucoup d'attention en raison de son importance pour de nombreuses entreprises qui cherchent à fournir une meilleure expérience pour leurs services. Malgré la volonté de fournir le meilleur service pour la réalité augmentée, le problème de l'occultation se tient devant les chercheurs.

Le concept de Réalité Augmentée vise donc à accroître la perception du monde réel en y ajoutant des éléments non perceptibles a priori par l'œil humain. Plusieurs problèmes doivent être résolus pour obtenir une incrustation réaliste. Il faut tout d'abord pouvoir déterminer le point de vue adopté pour chaque prise de vue (alignement des caméras réelle et virtuelle) afin d'incruster l'objet de synthèse au bon endroit. Il faut ensuite tenir compte des interactions entre les éléments virtuels insérés et la scène réelle : les parties occultées de ces éléments doivent être déterminées, ainsi que les modifications de l'éclairage induites par l'insertion de ces éléments.

Le problème d'occultation est donc est dû essentiellement au mixage du monde réel avec les objets virtuels utilisés pour l'augmentation de la scène.

Le présent projet se propose pour traiter ce problème, pour cela plusieurs techniques ont été proposé dans la littérature, nous avons choisi d'utiliser l'apprentissage profond pour d'une part effectuer une segmentation d'image afin d'identifier les objets d'intérêts et enlever l'arrière-plan et d'autre part gérer les occultations. Nous avons testé notre application sur un ensemble de datasses et les premiers résultats semble très prometteurs, ce qui nous encourage à améliorer de plus en plus nos recherches.

Pour les perspectives et travaux futurs, nous proposons des idées qui peuvent améliorer les méthodes de gestion des occultations en réalité augmentée, comme :

Augmentez le réalisme des objets virtuels en leur ajoutant un shaker.

Ajoutez un apprentissage approfondi pour connaître les sources d'éclairage de la scène.

L'étude de l'éclairage et de ses places dans la scène et son effet sur les objets virtuels.

Fusionner la réalité augmentée avec la réalité virtuelle pour créer une réalité mixte.

Références bibliographiques

1. Ababsa F, Mallem M. *Robust camera pose estimation combining 2D/3D points.* (2008).

2. A Akgul, I Moroz , I Pehlivan , S Vaidyanathan - Optik,. *Un nouvel attracteur chaotique à quatre rouleaux et ses applications d'ingénierie.* 2016.

3. A. G. Baydin, B. A. Pearlmutter, A. A. Radul et J. M. *« Automatic differentiation in machine learning: a survey », arXiv preprint arXiv:1502.05767 . 185.* (2015).

4. ababsa, f., didier, j. y., tazi, a. & mallem, m. «Software Architecture andCalibration Framework For Hybrid Optical IR and Vision Tracking System. The 15th.» (2007).

5. Alireza Makhzani et Brendan Frey. «« k-Sparse Autoencoders »,.» 2013.

6. Bangor, A., Kortum, P. & Miller, J. «Determining what individual SUS scoresJournal of usability studies,.» *Journal of usability studies*, 2009: Volume 4,123.

5. Bartlett, M. S. *Properties of sufficiency and statistical tests. Proceedings of the .* 1937.

6. Bichlmeier C, Wimmer F, Heining SM, Navab N. *Contextual anatomic mimesis: hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. p.87.* 2007.

7. Bruna E, Brombach B, Zeidler T, Bimber O. *Enabling mobile phones to support large-scale museum guidance. Multimedia, IEEE 14(2):16–25.* (2007) .

8. *Cool: Augmented Reality Advertisements. 29, 2020, from https://geekologie.com/2008/12/cool-augmented-reality-adverti.php.* 2008, December 19).

9. Huang Y, Liu Y, Wang Y. *AR-View: and Augmented Reality Device for Digital.* (2009).

10. *AR-View: and Augmented Reality Device for Digital Reconstruction of Yuangmingyuan, IEEE International Symposium on Mixed and Augmented Reality.* (2009) .

11. Lee B, Chun J. «nteractive manipulation of augmented objects in marker-less AR.» *Conference on InformationTechnology.* 2010).

12. Liou , C.Y., L.W. Cheng, et J.W. Liou. «Autoencoder for Words.» *Neurocomputing* 139 (2014): 84-96.

13. M Limmer, HPA Lensch. *Colorisation infrarouge à l'aide de réseaux de neurones à convolution profonde.* 2016.

14. Maddison, Chris J., Aja Huang, Ilya Sutskever, et David Silver. *"Move Evaluation in Go Using Deep Convolutional Neural Networks".* (2014).

15. Malaka R, Schneider K, Kretschmer U. *Stage-based augmented edutainment Smart Graphics.* p.54–65 vols. (2004).

16. Marco Sacco, Stefano Mottura, Luca Greco, Giampaolo Viganņ,. *Institute of Industrial Technologies and Automation, National Research Council.* Italy, s.d.

17. Masakazu Matusugu, Katsuhiko Mori, Yusuke Mitari et Yuji Kaneda,. *« Subject independent facial expression recognition with robust face detection using a convolutional neural network », Neural Networks.* Vol. . 16. 5 vols. 2003, s.d.

18. Meddeb, Abdi et. *Deep Learning Traffic Sign Detection, Recognition and.* (2017).

19. Milgram P, Kishino AF. *Taxonomy of mixed reality visual displays.* (1994).

20. Milgram P, Kishino AF (1994). *Taxonomy of mixed reality visual displays.* 1994) .

21. *Mini. . Official Homepage. Retrieved January 29, 2020, from https://www.mini.com/.* (2018, March 27).

22. Miyashita T, Meier P, Tachikawa T, Orlic S, Eble T, Scholz V, Gapel A, Gerl O, Arnaudov S,Lieberknecht S An augmented reality museum guide. Mixed and Augmented Reality,2008. ISMAR 2008. 7th IEEE/ACM International Symposium on, p.103–106, 15–18 S. *An augmented reality museum guide. Mixed and Augmented Reality,2008. ISMAR 2008. 7th IEEE/ACM International Symposium on,.* (2008).

23. N. Parmar J. Uszkoreit L. Jones A. N. Gomez L. Kaiser I. Polosukhin Aswani, N. Shazeer. *Attention is all you need. arXiv preprint arXiv:1706.03762,.* 12 2018.

24. Nilsson J, Odblom ACE, Fredriksson J, Zafar A, Ahmed F. *Performance evaluation method for mobile computer vision systems using augmented reality.* 2010.

25. Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio et Pierre- Antoine Manzagol, « Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion ». «« Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local

Denoising Criterion ».» *The Journal of Machine Learning Re*, 2010, p. 3371–3408 : p. 3371–3408.

26. Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio et Pierre-Antoine Manzagol. «« Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion ».» *The Journal of Machine Learning Research*, 2010.

27. Rock, Irvin.. *"The frame of reference." The legacy of Solomon Asch: Essays in cognition and social psychology* . (1990).

28. Sinno Jialin Pan et Qiang Yang. *« A Survey on Transfer Learning »,* . Vol. ,vol. 22. 2010,.

29. *« A Survey on Transfer Learning », IEEE Transactions on Knowledge and Data Engineering,.* Vol. . 22. 10 vols. 2010.

30. SS Roy, SI Hossain , MAH Akhand …. «Un système robuste de classification d'images bruitées combinant un auto - encodeur de débruitage et un réseau de neurones convolutifs.» *Journal of Advanced* , 2018.

31. Stutzman B, Nilsen D, Broderick T, Neubert J MARTI: Mobile Augmented Reality Tool for Industry. Computer Science and Information Engineering, 2009 WRI World Congress on, vol.5,p.425–429, March 31 2009-April 2. *MARTI: Mobile Augmented Reality Tool for Industry. Computer Science and Information Engineering, 2009 WRI World Congress on* . Vol. vol.5. (2009).

32. X Liu, M Wang , ZJ Zha, R Hong. *Apprentissage des fonctionnalités intermodalités via l' auto-encodeur convolutif.* 2019.