

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABBES LAGHROUR DE KHENCHELA
FACULTÉ 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

Thème:

*Apprentissage profond pour la
reconstruction des objets 3D*

Présenté et soutenu publiquement par :

KOB ALLA EDDINE et DEBBACHE BOUAGUAL

Le :14/09/2020

Membres de jury :

Président du jury : Bekhouch Abd el Ali.

Examineur : Bardo Dalel.

Encadreur : Abbas Fayçal.

Année universitaire 2019/2020

Résumé

La reconstruction d'objet 3D est un problème largement étudié par la communauté scientifique vu son importance dans plusieurs domaines d'applications tel : la sécurité, reconnaissance d'objets et compréhension des scènes, la modélisation 3D, l'animation, le contrôle industriel et le diagnostic médical. La reconstruction d'objets en utilisant les techniques traditionnelles est une tâche à la fois complexe et longue vu la quantité de caractéristiques dans des images, le problème d'échelle ainsi le problème de variation d'illumination et la récupération de la dimension de profondeur.

Ce mémoire présente une nouvelle méthode automatique de reconstruction d'objet en 3D à partir d'une seule image 2D, notre méthode permet de relever la complexité imposée par l'étape de modélisation tridimensionnelle des objets. En effet notre méthode permet d'offrir une reconstruction 3d en temps réel des objets qui ont une forme symétrique, notre méthode est basée sur l'utilisation des approches de deep learning afin d'atteindre notre objectif.

La proposition que nous avons faite se base sur le travail [1], dans un première temps une étape de détection et de segmentation d'instance est effectuée pour localiser l'objet 3D en utilisant un transfert d'apprentissage, puis dans un second temps une étape de reconstruction et de plaquage de texture est effectuée afin de générer l'objet final. Notre méthode produit de bons résultats en temps réel avec une précision satisfaisante.

Mots clés : Reconstruction 3D, reconstruction objet, Mask R-CNN, Deep learning.

Abstract

3D Object reconstruction is a problem widely studied by the scientific Community given its importance in several areas of application such as Security, Object recognition and scene Understanding, 3D modeling, animation, industrial control and medical diagnosis. Rebuilding objects using traditional techniques is a complex and time-consuming task given the amount of features in images, the scale problem and the problem of illumination variation and the recovery of depth dimension.

This memory presents a new automatic method of reconstructing objects in 3D from a single 2D image, our method allows us to identify the complexity imposed by the three-dimensional modeling stage of objects. Indeed our method allows to offer a 3D reconstruction in real time of objects that have a symmetrical shape, our method is based on the use of deep learning approaches in order to achieve our goal.

The proposal we made is based on the work [1], first a instance detection and segmentation step is performed to locate the 3D object using a learning transfer, and then in a second time a reconstruction and texture plate step is performed in order to generate the final object. Our method produces good results in real time with satisfactory accuracy.

Key words: 3D Reconstruction , object reconstruction , Mask R-CNN, Deep learning.

Dédicace

*D'un sentiment plein d'amour, de sincérité de fidélité, Je dédie ce
modeste travail à :*

*À mon bras, mon exemple éternel, mon soutien moral et source de joie
et de bonheur, celui qui s'est toujours sacrifié pour me*

voir réussir, que dieu te garde dans son vaste paradis, à toi

*Mon père (**LAZHAR**).*

*À la lumière de mes jours, la source de mes efforts, la femme de mon
cœur, ma vie et mon bonheur ; maman que j'adore (**HAKIMA**).*

*À mes frères (**Lamine, Fofu, Atef, Zizo**)*

Pour l'âme de mes grands pères et mères

*À mon binôme '**oussama**' et tous les membres de la famille
'DEBBACHE'*

À mes très chers amis(es).

Alla Eddine

Je dédie ce travail :

À mes chers parents qui m'ont toujours épaulé et soutenu, sans eux je n'y serai jamais arrivé, Tous les mots du monde ne sauraient exprimer

l'immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour tous les efforts et les sacrifices que vous Navez

jamais cessé de consentir pour mon instruction et mon bien-être.

*À mon cher frère : **YOUCEF.***

À ma grand-mère qui prie toujours pour moi et à laquelle je tiens tant.

*À tous les membres de ma grande famille (**DEBBACHE**)*

À mon cher binôme ~~ALLA~~ pour tout ce qu'elle a fait pour la réussite de ce travail.

*À mes plus proches amies : **RAMZI, MEHDI***

À mes amis d'études.

À tous mes enseignants sans exception.

Bouagual

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné le courage durant ces longues années d'étude, la force et la patience d'accomplir ce Modeste travail. Nous tenons à remercier sincèrement notre encadreur Dr : Abbes Faycel, pour ces précieux conseils, orientation, confiance et patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Ces remerciements vont tout d'abord au corps professoral et administratif de la Faculté des Science et Technologie, ensuit à tous

les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche.

Nous remercions de tous nos cœurs nos parents, pour nous avoir toujours soutenu et cru en nos.

A nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Merci à tous et à toute

Table des matières

INTRODUCTION GENERAL	1
----------------------------	---

CHAPITRE 1: L'intelligence artificielle

1.L'intelligence artificielle	4
1.1 Quelques Définitions	4
1.2 Un court historique d'IA	4
1.3 Grands domaines d'IA	5
2.L'apprentissage automatique.....	6
2.1 Quelques définitions :.....	6
2.2 Pourquoi l'apprentissage automatique ?	6
2.3 Types de systèmes d'apprentissage automatique	7
2.3.1 L'apprentissage supervisé	7
2.3.2 L'apprentissage non supervisé.....	8
2.3.3 L'apprentissage par renforcement	9
2.3.4 L'apprentissage semi-supervisé	10
2.3.5 L'apprentissage en profondeur.....	11
3.La différence entre l'apprentissage automatique et l'apprentissage en profon.....	16
4.Conclusion	17

Chapitre 2 : La détection et la reconstruction des objets 3D

1.Introduction	19
2.Détection d'objets.....	19
2.1 Les défis de la détection.....	19
2.2 Le principe général de la détection	20
2.3 Méthodes de détection	20
a. Méthode de R-CNN	20
b. Méthode de YOLO	21
c. Méthode de Fast R-CNN	21
d. Méthode de Faster R-CNN.....	22
e. Méthode de SSD	23
f. Mask R-CNN	24

g. Segmentation d'instance	24
3.Larecognition d'objets dans les images	25
3.1 Introduction.....	25
3.2 Les principaux domaines d'application.....	25
3.3 Les défis majeurs de la reconnaissance d'objets	26
3.4 Extraction des descripteurs.....	26
i. Les points d'intérêt.....	27
ii. Les descripteurs contours	28
iii. Les descripteurs régions.....	29
iv. Les descripteurs de formes.....	30
4.La Reconstruction 3D.....	32
4.1 Introduction	32
4.2 Domaines d'applications de la reconstruction 3D	32
4.3 Reconstruction 3D à partir d'une image	32
4.4 Reconstruction 3D à partir de plusieurs images	33
4.5 Reconstruction 3D basé sur l'apprentissage profond	33
5.Conclusion.....	35

Chapitre 3 : Implémentation

1.INTRODUCTION	37
2.Notre Modèle	37
2.1 L'Architecture	37
2.2 L'image d'entrée	38
2.3 La Segmentation de l'objet	38
2.4 La matrice (type,score).....	39
2.5 Matrice position pixels	40
2.6 Appliquer un filtre.....	41
2.7 Triangulation	42
2.8 Placage de texture	43
3.Conclusion.....	43

Chapitre 4 Résultat

1.Introduction	45
2.LA CONFIGURATION DU MATERIEL UTILISE	45
3.L'ENVIRONNEMENT DE TRAVAIL	45
3.1 Python	45
3.2 JavaScript & NodeJS	45
3.3 Pytorch.....	46
3.4 Open cv	46
4.Résultat	47
5.Conclusion.....	55
Conclusion générale et perspectives.....	56
Bibliographie	56

Liste des figures

Fig1.1 Processus d'apprentissage automatique.....	7
Fig1.2 Explique le déroulement d'apprentissage supervisé	8
Fig1.3 Explique le déroulement d'apprentissage non-supervisé	9
Fig1.4 Explique le déroulement d'apprentissage par renforcement	10
Fig1.5 un processus d'auto apprentissage.....	12
Fig1.6 Schéma très simplifié d'un réseau neuronal.	13
Fig1.7 Réseaux de neurones convolutifs	15
Fig1.8 Structure schématique d'un auto-encodeur avec 3 couches cachées entièrement connectées.	15
Fig1.9 Comparaison de méthodes de catégorisation de véhicules d'apprentissage automatique.....	16
<hr/>	
Fig2.1 Exemples de détections d'objets Renvoyées par un détecteur multi-classes.....	19
Fig2.2 Architecture de R-CNN	20
Fig2.3 Architecture de YOLO	21
Fig2.4 Architecture de Fast R-CNN	22
Fig2.5 Architecture de Faster R-CNN.....	22
Fig2.6 Faster R-CNN : Fenêtre coulissante et résultat	23
Fig2.7 Architecture of Single Shot MultiBox detector (input is 300x300x3]	23
Fig2.8 Architecture Mask R-CNN.	24
Fig2.9 Exemple de segmentation d'instanc	24
Fig2.10 Les coins détectés sur une image satellitaire par le détecteur de Harris et Stephen] ...	27
Fig2.11 Les points d'intérêt et les descripteurs locaux, dans Schmid et Mohr	28
Fig2.12 – Les fragments de contours de quelques objets courants, dans Shotton et al.	28
Fig2.13 – Illustration de l'algorithme Watershed sur un signal 1D	30
Fig2.14 Architecture du modèle Mesh RCNN.....	34
Fig2.15 mesure de paramètres avec la grille de référence.	34

Fig3.1 L'architecture de notre méthode.....	37
Fig3.2 Image d'entrée contient plusieurs objets	38
Fig3.3 Architecture masque R-CNN	39
Fig3.4 la segmentation de l'objet bouteille.....	39
Fig3.5 matrice contient des informations sur l'objet (score et true or false de l'objet cible). .	40
Fig3.6 matrice true or false de l'objet cible	40
Fig3.7 matrice de X et Y de chaque pixel de l'objet cible	41
Fig3.8 la Triangulation de l'objet cible (Bouteille) sans texture	42
Fig3.9 Placage de texture de l'objet cible (Bouteille)	43

Fig4.1 Exemple d'objet orange (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	47
Fig4.2 Exemple d'objet pomme (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	48
Fig4.3 Exemple d'objet ballon de sport : (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	49
Fig4.4 Exemple d'objet œuf (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	50
Fig4.5 Exemple d'objet bouteille (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	51
Fig4.6 Exemple d'objet cup (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	52
Fig4.7 Exemple d'objet Montgolfière (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	53
Fig4.8 Exemple d'objet Vase (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.....	54

INTRODUCTION GENERAL

La reconstruction des objets 3D en vision par ordinateur et en synthèse d'image n'est pas un sujet récent, cependant avec l'arrivée des approches de deep learning ce dernier a récemment beaucoup évolué, cela est due aux tendances actuelles du matériel en particulier les GPU.

La reconstruction 3D est une partie essentielle de la compréhension de la scène ainsi permet d'établir une relation spatiale entre les objets composants la scène 3D, le défi major de reconstruction d'objet 3D réside dans la récupération avec précision est efficacité les caractéristiques pertinentes, la texture et la profondeur à partir d'une image 2D seulement, cette difficulté correspond à un passage à partir un espace 2D vers un espace 3D.

Avec l'avenue des techniques d'apprentissage profond, cette évolution a conduit à une nouvelle génération de méthodes capables de produire une représentation géométrie 3D avec différentes poses et cela en se basant seulement sur une d'entrée unique. Bien qu'elles soient récentes, ces méthodes ont démontré des résultats prometteurs dans la reconstruction 3D.

Dans ce mémoire, nous proposons une méthode capable d'automatiser le processus de reconstruction d'objet 3D à partir d'une image 2D, notre méthode opère en deux étapes,

La première étape consiste à effectuer une détection et reconnaissance d'objet sur l'image d'entrée à travers une segmentation d'instance est effectuée afin de récupérer les pixels de l'objet, la seconde étape consiste à produire une représentation géométrique (mesh) de l'objet ainsi un plaquage de texture afin d'obtenir le rendu final de l'objet 3D.

Ce document est divisé en 4 chapitres, il est organisé comme suit :

Chapitre1 : le premier chapitre est divisé en trois parties, la première partie contient l'intelligence artificielle, l'historique d'intelligence artificielle et ses domaines, la deuxième partie inclut l'apprentissage automatique, les types de systèmes d'apprentissage automatique et leur modèle. La troisième partie contient l'apprentissage profond et quelques algorithmes de ce apprentissage et enfin la différence entre l'apprentissage automatique et l'apprentissage profond.

Chapitre2 : le deuxième chapitre expose la détection d'objets et la Reconnaissance 3D et puis la Reconstruction 3D.

Chapitre3 : le troisième chapitre présente les détails d'implémentation. Ce chapitre définit l'architecture de modèle qu'on a créé pour détecter et reconstruction 3D des objets .

Chapitre4 : le quatrième chapitre conclut ce mémoire en présentant l'ensemble des résultats expérimentaux obtenus de reconstruction 3D

Une conclusion synthétiser les contributions et les résultats obtenus et proposera des perspectives à nos travaux.

Chapitre I :
L'APPRENTISSAGE AUTOMATIQUE

1. L'intelligence artificielle

Depuis l'avènement de la robotique et de l'informatique, les chercheurs ont essayé d'introduire les concepts de l'intelligence humaine dans les machines. Créé et fabriqué par l'homme, nous classons cette forme d'intelligence comme « l'intelligence artificielle » ou IA. [2]

L'intelligence artificielle est l'un des domaines les plus importants et passionnants de l'informatique. Ce dernier couvre les fondamentaux de l'informatique et des systèmes d'information afin de maximiser vos chances d'avoir des emplois futurs. Il offre également la possibilité de mener des recherches de pointe dans des domaines de haut niveau...

1.1 Quelques Définitions

Qu'est-ce que l'intelligence ?

« La faculté de connaître et comprendre, incluant la perception, l'apprentissage, l'intuition, le jugement et la conception. » (Petit Robert).

« La faculté de connaître et de raisonner. » (Dictionnaire American Heritage).

« Application de la connaissance à la résolution de problèmes. » (Newell et Simon).

L'intelligence artificielle (IA) c'est une science dont le but est de résoudre avec la machine les tâches qu'une personne effectue en utilisant son intelligence.

1.2 Un court historique d'IA

L'intelligence artificielle (IA) est un domaine qui a une longue histoire et qui est toujours en constante évolution.

- 1943 : McCulloch et Pitts créent le modèle du neurone formel.
- 1948 : Création de la cybernétique (science des systèmes) par Norbert Wiener.
- 1949 : Hebb établit la première règle d'apprentissage neuronal.
- 1950 Shannon, 1952 Samuel, 1953 Turing : machine pour jouer aux échecs.
- 1956 Workshop où est né le terme "intelligence artificielle".
- 1959 Rochester : Geometry Theorem Prover.
- 1958 McCarthy au MIT crée le LISP et le "time sharing". Créé DIGITAL.
- 1960 John McCarthy, Allen Bewell & Herbert Simon: L'ordinateur peut être utilisé pour autre chose que des calculs « manipuler des symboles » (idée proposée par Ada Lovelace amie de Babbage 1842).
- 1969 arrêt des RNs (Minsky & Paper 1969) limitations des perceptrons.
- 1969-1979 : systèmes experts.
- Depuis 1986 : retour des réseaux de neurone. [3]

1.3 Grands domaines d'IA

L'intelligence artificielle apparaît dans une variété de domaines, c'est un principe universel. Il est utilisé dans des domaines où l'on ne le voit pas, comme dans la vie quotidienne (GPS, informatique, reconnaissance vocale, météo...), et nous le trouvons dans des domaines beaucoup plus impressionnants à la limite de la science-fiction, comme la robotique ou le jeu. Voici quelques exemples :

Apprentissage / Adaptation :

Exemples : construction de systèmes experts, classification automatique de galaxie, contrôleurs de robots....

Reconnaissance et synthèse de la parole :

Exemples : réservation d'hôtel, annuaire téléphone.

Reconnaissance et synthèse d'images :

Exemples : effets spéciaux au cinéma, vidéo surveillance.

Reconnaissance de l'écriture :

Exemples : reconnaissance chèques, codes postaux.

Langage naturel :

Exemples : interface, text mining, web mining.

Aide à la décision :

Exemple : contrôle de trajectoire du satellite voyagé, (Systèmes experts).

Aide à la programmation :

Exemple : agents d'interface.

Jeux :

Exemples : Echecs (DeeperBlue à 2600), Checkers (Champion), Othello (Champion), Back Gammon (champion), GO (bon amateur) [2].

- Flach Peter (2012) a défini l'apprentissage automatique comme une étude systématique des algorithmes et des systèmes dans le but d'améliorer leurs performances ou bien leur connaissance avec l'expérience
- Mohammad Saeid Mahdavinejad et al. (2018) ont considéré l'apprentissage automatique comme un type de l'intelligence artificiel (IA) qui permet aux machines d'apprendre sans l'intervention de l'être humain
- De notre point de vue, nous constatons que l'apprentissage automatique est un ensemble d'algorithmes qui consiste à programmer les ordinateurs afin d'optimiser les critères de performance avec l'utilisation des exemples de données ou une expérience précédente afin de donner aux ordinateurs la capacité pour apprendre [4]

2. L'apprentissage automatique

2.1 Quelques définitions :

Un ensemble de théories et de techniques mises en œuvre pour créer des machines capables d'imiter l'intelligence. [5]

L'apprentissage est une étape dans le développement d'un réseau neuronal, au cours de laquelle le comportement du réseau change jusqu'à ce que le comportement souhaité est obtenu. L'apprentissage neuronal utilise des exemples comportementaux. [6]

- **Apprentissage Automatique** (Machine Learning) est 'une méthode de science des données qui permet aux ordinateurs d'utiliser les données existantes pour prédire les tendances, les résultats et les comportements futurs. Avec l'apprentissage automatique, les ordinateurs apprennent sans programmation explicite. [7] Le but est double :
 - Prédire un comportement face à une nouvelle donnée.
 - Approximer une fonction ou une densité de probabilité.

Voici une définition un peu plus générale :

- **L'apprentissage automatique** c'est le domaine d'étude qui permet aux ordinateurs d'apprendre sans être explicitement programmés. (Arthur Samuel, 1959).

Et un plus orienté vers l'ingénierie :

On dit qu'un programme informatique apprend de l'expérience E par rapport à une tâche T et à une mesure de performance P, si sa performance sur T, telle que mesurée par P, s'améliore avec l'expérience E. (Tom Mitchell, 1997)

2.2 Pourquoi l'apprentissage automatique ?

L'apprentissage automatique (en anglais ML) utilise des ordinateurs pour simuler l'apprentissage humain et permet aux ordinateurs d'identifier et d'acquérir des connaissances du monde réel, et d'améliorer les performances de certaines tâches en fonction de ces nouvelles connaissances [8]

Pour mieux comprendre l'utilisation de l'apprentissage automatique, nous allons sites quelques domaines de ce dernier :

- Auto-conduite de Google.
- Détection de cyber-fraude.
- Moteurs de recommandation en ligne comme des suggestions d'amis sur Facebook.

Netflix présentant les films et émissions que vous aimerez et « plus d'éléments à prendre en compte » et « obtenez-vous un petit quelque chose » sur Amazon sont autant d'exemples d'apprentissage automatique appliqué.

Tous ces exemples font écho au rôle essentiel que l'apprentissage automatique a commencé à prendre dans le monde riche en données d'aujourd'hui. Les machines peuvent aider à filtrer des informations utiles qui aident à des avancées majeures, et nous voyons déjà comment cette technologie est mise en œuvre dans une grande variété d'industries.

Le flux de processus représenté ici représente le fonctionnement de l'apprentissage automatique [9]

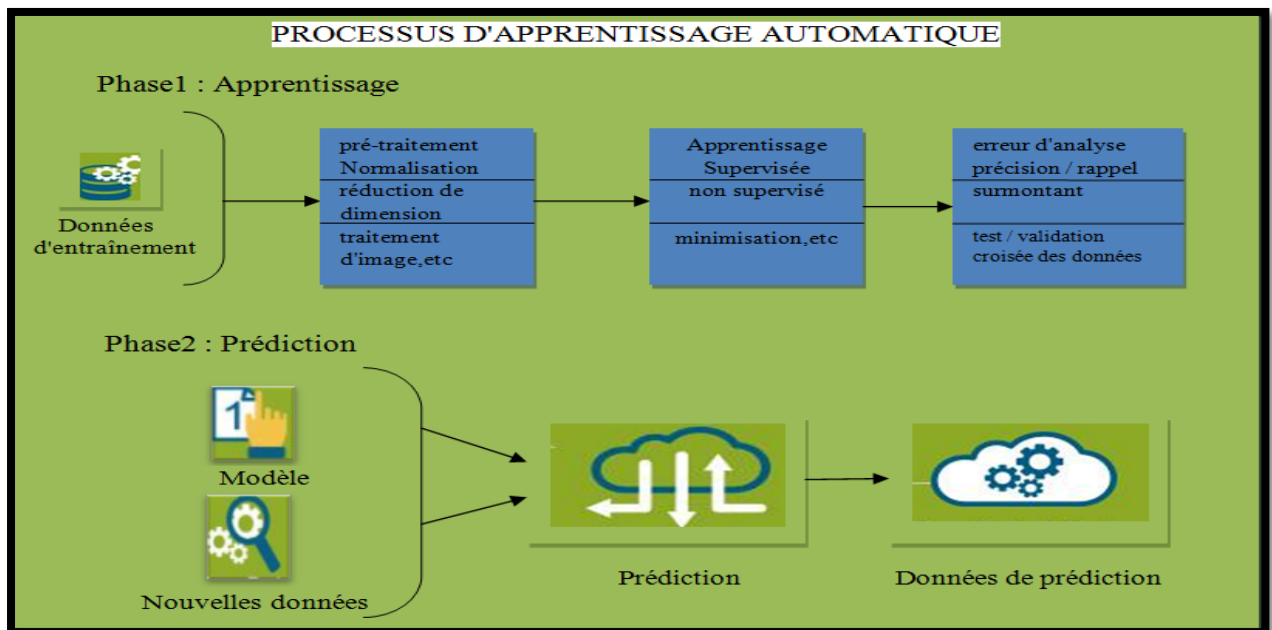


Fig1.1 Processus d'apprentissage automatique [9]

2.3 Types de systèmes d'apprentissage automatique

Nous allons introduire différents types d'apprentissage utilisés en apprentissage automatique. Nous verrons l'apprentissage supervisé, non supervisé, semi-supervisé et par renforcement. Chaque type d'apprentissage a leur spécificité et son utilisé en fonction du problème que nous rencontrons.

2.3.1 L'apprentissage supervisé

L'apprentissage supervisé commence généralement par un ensemble de données bien défini et une certaine compréhension de la façon dont ces données sont classifiées. L'apprentissage supervisé a pour but de déceler des modèles au sein des données et de les appliquer à un processus analytique.

Ces données comportent des caractéristiques associées à des libellés qui définissent leur signification.

Vous pouvez, par exemple, créer une application d'apprentissage automatique capable de faire la distinction entre plusieurs millions d'animaux, en se basant sur des images et des descriptions écrites. [10]

Voici quelques algorithmes d'apprentissage supervisé

- Régression linéaire
- Régression logistique
- Arbres de classification et de régression
- K-NN
- Naïve Bayes Classifier[11]

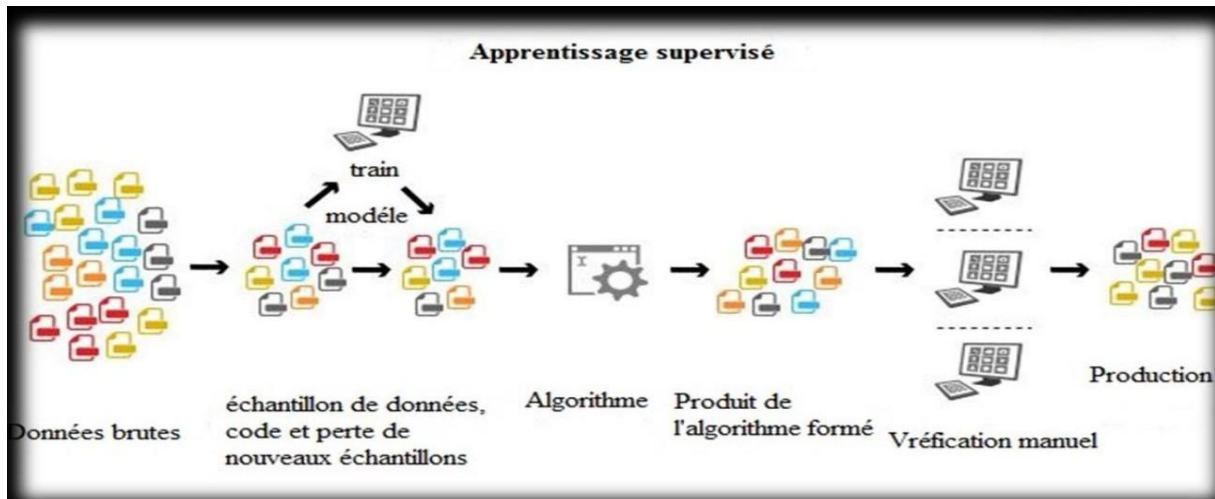


Fig1.2 Explique le déroulement d'apprentissage supervisé [11]

2.3.2 L'apprentissage non supervisé

L'apprentissage non supervisé est utilisé lorsqu'un problème nécessite une grande quantité de données non marquées. Par exemple, toutes les applications de médias sociaux, telles que Twitter, Instagram et Snapchat, génèrent de très grandes quantités de données non marquées. Pour comprendre la valeur de ces données, vous devez utiliser des algorithmes qui classent les données en fonction des tendances ou des clusters qu'ils détectent. L'apprentissage non supervisé conduit à un processus itératif où les données sont analysées sans intervention humaine. Il est utilisé avec la technologie pour détecter les spams envoyés par e-mail. Les e-mails ordinaires et le spam contiennent trop de variables pour que l'analyste signale les messages de spam massifs. En revanche, les discriminateurs d'apprentissage automatique basés sur le regroupement et l'association sont utilisés pour identifier les e-mails indésirables. [10]

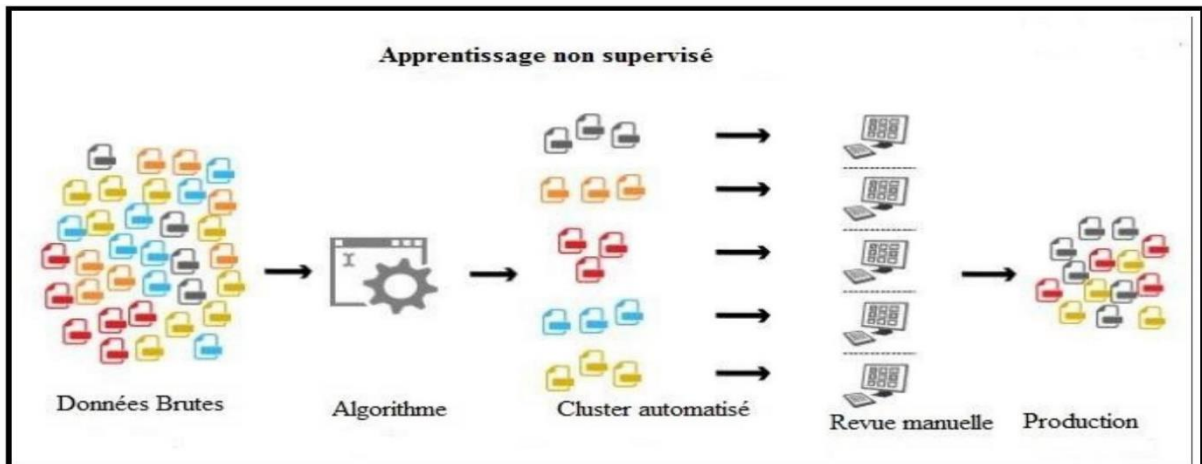


Fig1.3 Explique le déroulement d'apprentissage non-supervisé.[11]

Voici quelques-uns des algorithmes d'apprentissage non supervisé les plus importants :

Clustering :

- K-Moyens.
- Hierarchical Cluster Analysis (HCA).
- Visualisation et réduction de la dimensionnalité :
 - Analyse des composants principaux (PCA).
 - Inclusion linéaire-linéaire (LLE).
 - Intégration de voisin stochastique t distribuée (t-SNE).
- Apprentissage des règles d'association :
 - A priori.
 - Éclat

2.3.3 L'apprentissage par renforcement

L'apprentissage par renforcement est un modèle d'apprentissage comportemental. L'algorithme reçoit la rétroaction de l'analyse des données et dirige l'utilisateur vers le meilleur résultat. L'apprentissage par renforcement diffère des autres types d'apprentissage supervisé car le système n'est pas formé sur un ensemble de données sélectives. Au lieu de cela, le système apprend par tâtonnements. Par conséquent, la séquence des décisions réussies conduit au renforcement du processus, parce que c'est ce qui résout le problème le plus efficacement.[10]

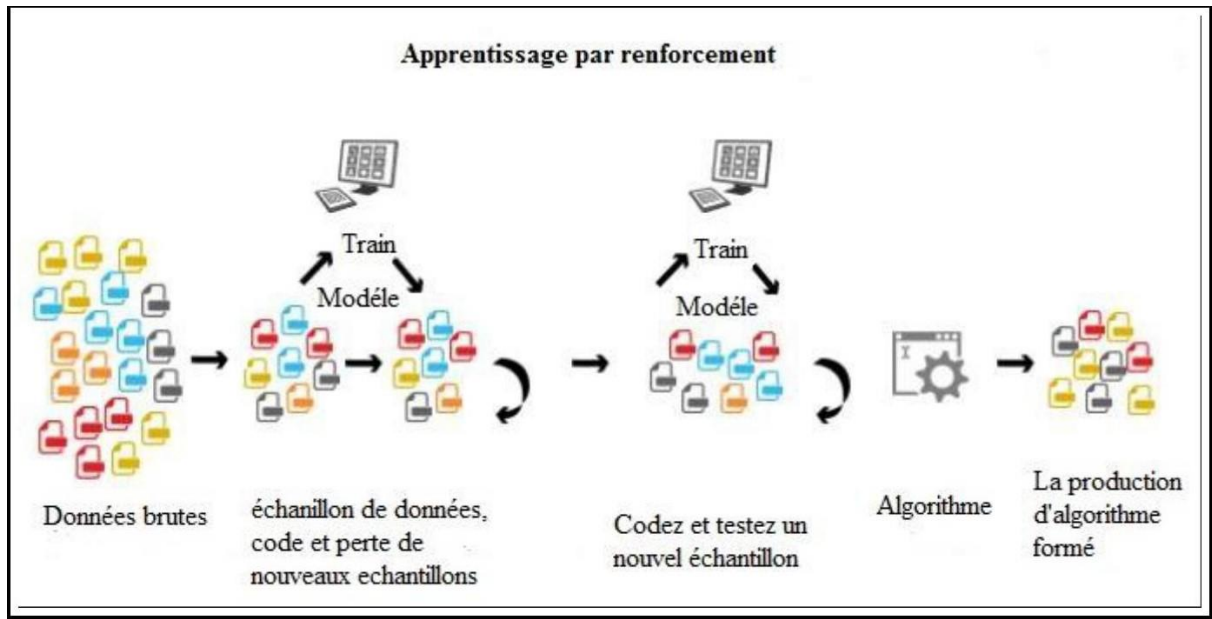


Fig1.4 Explique le déroulement d'apprentissage par renforcement.[11]

2.3.4 L'apprentissage semi-supervisé

L'apprentissage semi-supervisé est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données marquées et non marquées. Ainsi, il s'agit entre l'apprentissage supervisé qui utilise uniquement des données étiquetées, et l'apprentissage non supervisé qui utilise uniquement des données non étiquetées. Il a été démontré que l'utilisation de données non étiquetées, combinée à des données étiquetées, améliore considérablement la qualité de l'apprentissage.

Une autre préoccupation se pose parce que l'étiquetage des données nécessite souvent l'intervention de l'homme et de l'utilisateur. Lorsque les jeux de données deviennent très volumineux, cela peut être fastidieux. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, revêt un intérêt pratique évident.

Un exemple d'apprentissage semi-supervisé est la Co apprentissage, dans laquelle deux classificateurs étudient un ensemble de données, mais chacun utilise un ensemble de caractéristiques différentes et parfaitement indépendantes. Si les données concernent des hommes et des femmes, vous pouvez utiliser la taille et une taille de cheveux différente.[12]

2.3.5 L'apprentissage en profondeur

i. Définition

L'apprentissage profond ou apprentissage en profondeur (en anglais : deep learning, deep structured learning, hierarchical learning) est un ensemble de techniques d'apprentissage automatique qui tentent de modéliser avec un niveau élevé d'abstraction de données en utilisant des architectures articulées de diverses transformations non linéaires. Ces méthodes ont fait des progrès significatifs et rapides dans l'analyse des signaux sonores ou visuels, et en particulier la reconnaissance faciale, la voix, la vision par ordinateur et le traitement automatisé de la parole. Dans les années 2000, ces progrès ont généré d'importants investissements privés, universitaires et publics, en particulier de la part de GAFAM (Google, Apple, Facebook, Amazon, Microsoft). [13]

L'apprentissage profond est une méthode d'apprentissage automatique. Elle permet d'enseigner à la machine des actions, des informations et des concepts qui sont naturellement acquis par l'être humain. Par exemple, cela lui permet de distinguer un piéton d'un panneau de signalisation [15]

ii. Applications du l'Apprentissage en profondeur

Le deep Learning est utilisé dans de nombreux domaines :

- Reconnaissance d'image
- Traduction automatique
- Voiture autonome
- Diagnostic médical
- Recommandations personnalisées
- Modération automatique des réseaux sociaux
- Prédiction financière et trading automatisé
- Identification de pièces défectueuses
- Détection de malwares ou de fraudes
- Chatbots (agents conversationnels)
- Exploration spatiale,
- Robots intelligents.[16]

iii. Fonctionnement du l'Apprentissage en profondeur

L'apprentissage en profondeur s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le

système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.[16]

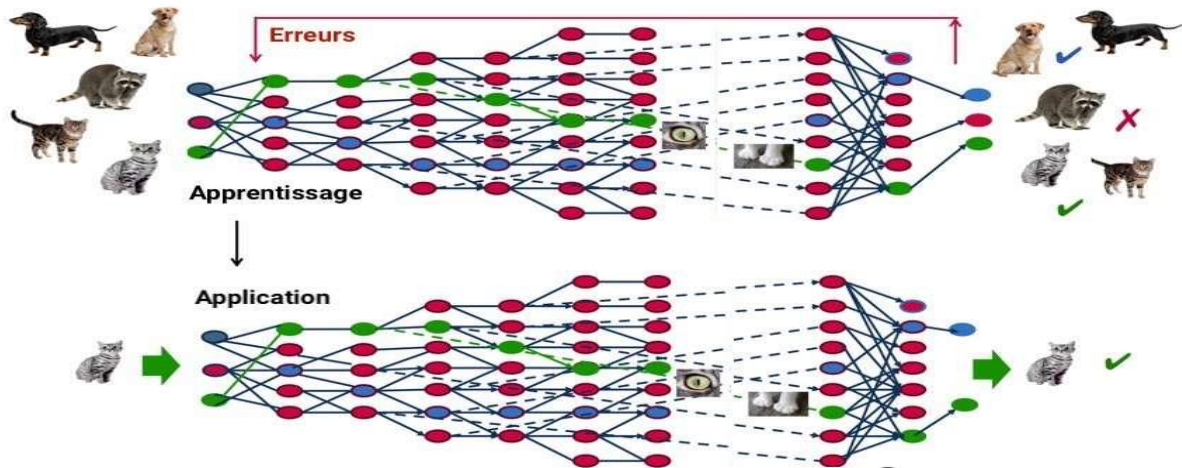


Fig1.5 un processus d'autoapprentissage, le deep Learning est capable d'identifier un chat sur une photo. À chaque couche du réseau neuronal correspond un aspect particulier de l'image.[16]

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ai jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant. [16]

iv. Algorithme du l'Apprentissage en profondeur

L'apprentissage profond consiste à apprendre plusieurs niveaux de représentation et d'abstraction qui aident à donner du sens à des données telles que des images, du son et du texte. Pour plus d'informations sur les algorithmes d'apprentissage profond.

Il existe différents algorithmes de l'apprentissage profond. Nous pouvons ainsi citer :[16]

- a) **Les réseaux de neurone**
- b) **Les Réseau de neurones Dits Récurrents (RNN)**
- c) **Les Réseau de neurones convolutifs (CNN) :**

a) Les réseaux de neurone :

En anglais neural network. C'est un ensemble de neurones formels interconnectés permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage. [17]

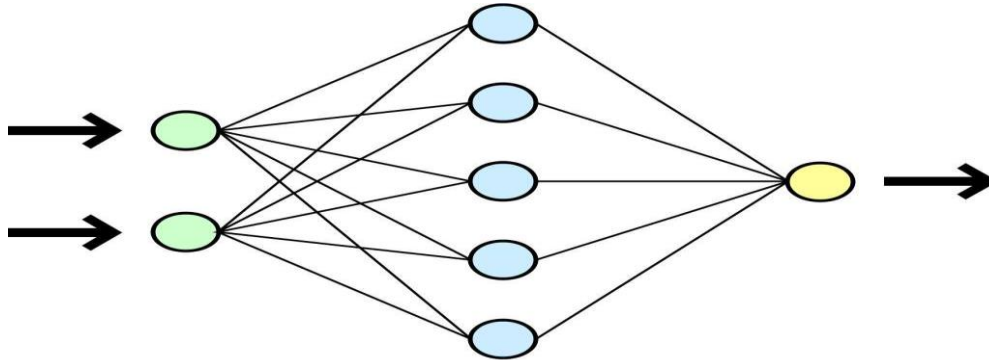


Fig1.6 Schéma très simplifié d'un réseau neuronal. Les deux neurones de gauche (en vert) reçoivent les informations. Le traitement de ces données est déterminé par leurs connexions avec les neurones internes (en bleu). Les neurones qui reçoivent une donnée sont activés. L'information finale est envoyée sur le dernier neurone (en jaune) ou sur l'organe effecteur (un moteur par exemple).[17]

• Fonctionnement Les réseau de neurones

Le réseau neuronal est basé sur le fonctionnement des neurones biologiques et prend forme dans l'algorithme informatique. Un réseau neuronal peut se changer en fonction des résultats de ses actions, vous permettant d'apprendre et de résoudre des problèmes sans algorithmes, donc, sans programmation classique.[17]

• Applications des réseaux de neurones

Les principales applications des réseaux de neurones sont l'optimisation et l'apprentissage. En apprentissage, les réseaux de neurones sont essentiellement utilisés pour :

- L'apprentissage supervisé
- L'apprentissage non supervisé
- L'apprentissage par renforcement

Pour ces trois types d'apprentissage, il y a également un choix traditionnel entre :

- L'apprentissage << off-line >> : toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément

- L'apprentissage << on-line >> : Les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité [18]

- **Classifications Les réseau de neurones**

Un réseau de neurones formels est constitué d'un grand nombre de cellules de base interconnectées. De nombreuses variantes sont définies selon le choix de la cellule élémentaire, de l'architecture du réseau et de la dynamique du réseau. Une cellule élémentaire peut manipuler des valeurs binaires ou réelles. Les valeurs binaires sont représentées par 0 et 1 ou -1 et 1. Différentes fonctions peuvent être utilisées pour le calcul de la sortie. Le calcul de la sortie peut être déterministe ou probabiliste. L'architecture du réseau peut être sans rétroaction, c'est à dire que la sortie d'une cellule ne peut influencer son entrée. Elle peut être avec rétroaction totale ou partielle. La dynamique du réseau peut être synchrone : toutes les cellules calculent leurs sorties respectives simultanément. La dynamique peut être asynchrone. Dans ce dernier cas, on peut avoir une dynamique asynchrone séquentielle : les cellules calculent leurs sorties chacune à son tour en séquence ou avoir une dynamique asynchrone aléatoire. Par exemple, si on considère des neurones à sortie stochastique -1 ou 1 calculée par une fonction à seuil basée sur la fonction sigmoïde, une interconnexions complète et une dynamique synchrone, on obtient le modèle de Hopfield et la notion de mémoire associative. Si on considère des neurones déterministes à sortie réelle calculée à l'aide de la fonction sigmoïde, une architecture sans rétroaction en couches successives avec une couche d'entrées et une couche de sorties, une dynamique asynchrone séquentielle, on obtient le modèle du Perceptron multi-couches (PMC) qui sera étudié dans les paragraphes suivants [18]

- b) Les Réseau de neurones convolutifs (CNN) :**

En apprentissage automatique, un réseau de neurones convolutifs ou réseau de neurones à convolution (en anglais CNN ou ConvNet pour Convolutional Neural Networks) est un type de réseau de neurones artificiels acycliques (feed-forward), 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 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 prétraiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel [19]

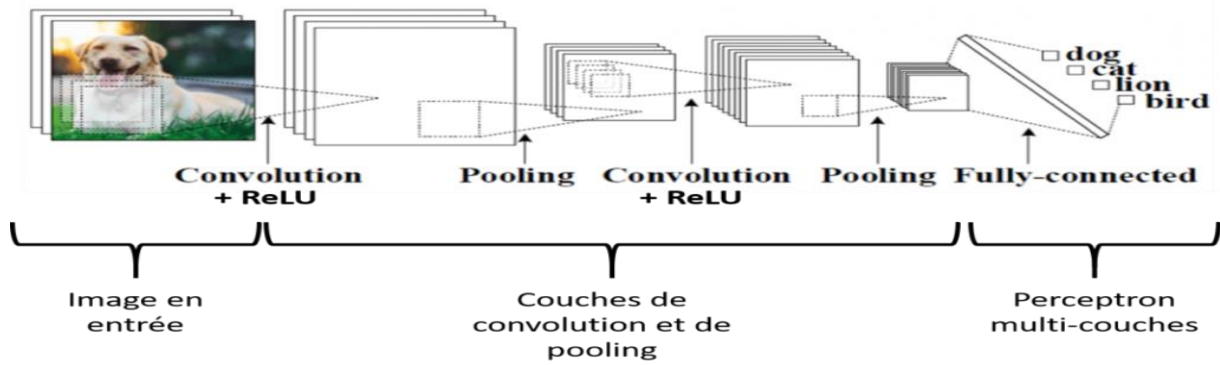


Fig1.7 Réseaux de neurones convolutifs.[20]

c) Les Réseau de neurones Dits Récurrents (RNN):

Un réseau de neurones récurrent (RNN, récurrent neural network) est un type de réseau neuronal artificiel qui est principalement utilisé pour la reconnaissance vocale et le traitement automatique du langage naturel (TAL, NLP, TNL). RNN sont conçus pour reconnaître les caractéristiques séquentielles et les modèles d'utilisation des données nécessaires pour prédire le scénario le plus probable suivant.

Un auto-encodeur, ou auto-associateur est un réseau de neurones artificiels utilisé pour l'apprentissage non supervisé de caractéristiques discriminantes. L'objectif d'un auto-encodeur est d'apprendre une représentation (encodage) d'un ensemble de données, généralement dans le but de réduire la dimension de cet ensemble. Récemment, le concept d'auto-encodeur est devenu plus largement utilisé pour l'apprentissage de modèles génératifs

L'architecture de notre réseau est le suivant : [19]

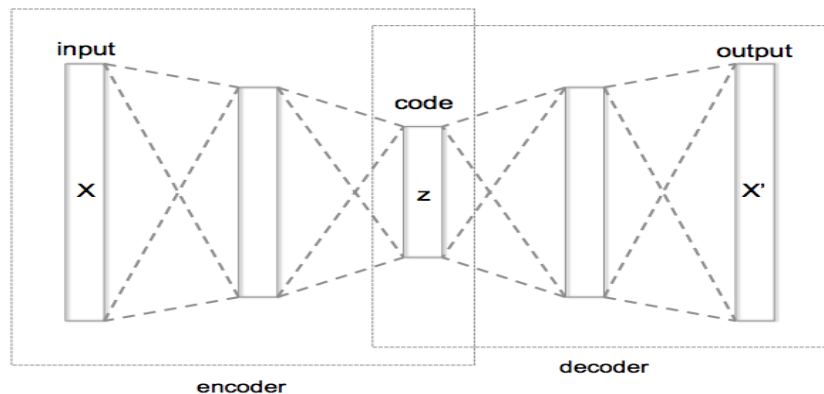


Fig1.8 Structure schématique d'un auto-encodeur avec 3 couches cachées entièrement connectées.[21]

3. La différence entre l'apprentissage automatique et l'apprentissage en profond

L'apprentissage en profond est une branche particulière de l'apprentissage automatique. Un processus de l'apprentissage automatique commence par l'extraction manuelle de caractéristiques pertinentes à partir d'images. En s'appuyant sur ces caractéristiques, un modèle qui catégorise les objets de l'image est ensuite créé. Dans un processus d'apprentissage profond, l'extraction de caractéristiques pertinentes à partir d'images est automatique. En outre, l'apprentissage profond effectue un apprentissage « de bout en bout » : à partir de données brutes, un réseau se voit assigner des tâches à accomplir (une classification, par exemple) et apprend comment les automatiser.

Une autre différence majeure est le fait que les algorithmes d'apprentissage profond évoluent avec les données, tandis que l'apprentissage peu profond converge. L'apprentissage peu profond désigne les méthodes d'apprentissage automatique dont la progression s'arrête à partir d'un certain niveau de performance après l'alimentation du réseau en exemples supplémentaires et en données d'apprentissage.

Un des avantages majeurs des réseaux d'apprentissage profond réside dans leur capacité à continuer à s'améliorer en même temps que le volume de vos données augmente.

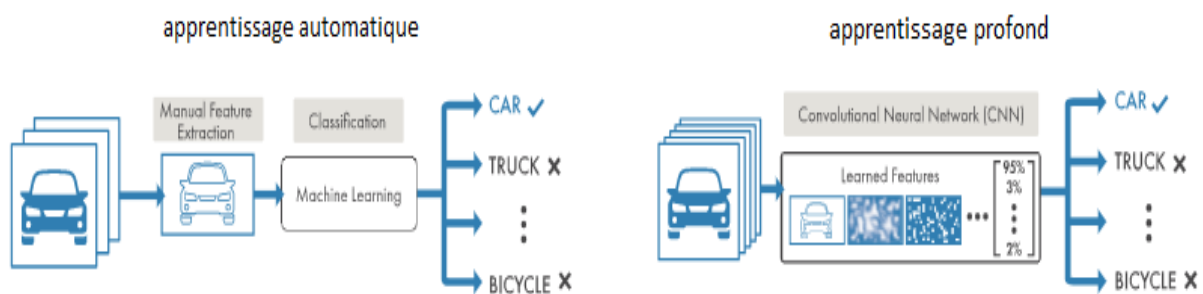


Fig1.9 Comparaison de méthodes de catégorisation de véhicules d'apprentissage automatique (gauche) et d'apprentissage profond (droite).

Pour classer des images avec l'apprentissage automatique, les choix de caractéristiques et de classificateur doivent être effectués manuellement. Avec l'apprentissage profond, l'extraction de caractéristiques et le processus de modélisation sont automatiques.[22]

4. Conclusion

L'intelligence automatique 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,

Dans le chapitre suivant, nous effectuons une étude sur la détection et la reconnaissance puis la reconstruction des objets 3d

Chapitre II :
La détection et la reconstruction des objets 3D

1. Introduction

Ce chapitre a pour l'objectif de présenter les techniques de détection d'objets on général, les défis de la détection, le principe général de la détection, les différentes approches de détection d'objets comme YOLO, R-CNN, puis nous allons présenter La reconnaissance d'objets dans les images ainsi les principaux domaines d'application enfin nous allons présenter la Reconstruction 3D à partir d'une ou plusieurs images.

2. Détection d'objets

La détection d'objets est une des problématiques les plus étudiées en vision par ordinateur. Son objectif est de trouver dans une image d'entrée, des régions contenant des objets. On peut diviser la détection d'objets en deux catégories :

- Détecteur à une classe :

Se concentre sur la détection d'un seul type d'objet. Le détecteur doit être capable de décider si une région de l'image correspond à l'objet ou à du fond.

- Détecteur multi-classes :

Le détecteur doit être capable de séparer le fond et les objets, tout en séparant les objets entre eux Il existe de nombreuses bases de données publiques permettant d'entraîner et d'évaluer des détecteurs . **La Fig 2.1** illustre l'objectif de la détection d'objets. Chaque couleur de boite englobant correspond à une classe d'objets [23].



Fig2.1 Exemples de détections d'objets Renvoyées par un détecteur multi-classes.

2.1 Les défis de la détection

Un détecteur d'objets doit faire face à plusieurs difficultés :

- ❖ La première est le temps de calcul : un bon détecteur est un détecteur qui maximise les performances tout en étant le plus rapide possible à exécuter.

❖ La seconde concerne l'apparence variable des objets. Celle-ci peut varier suivant plusieurs facteurs : l'occultation, le point de vue sous lequel les objets sont observés et la taille des objets dans l'image.

❖ Une troisième difficulté est en lien avec la variation d'apparence des régions qui ne correspondent pas à des objets (le fond). Un détecteur doit être capable de décider si une région correspond à du fond. [24]

2.2 Le principe général de la détection

Un détecteur est un modèle pour distinguer les zones d'une image. Cela peut être appris par exemple, grâce à l'apprentissage supervisé. La détection consiste à extraire des régions dans l'image d'entrée et à leur appliquer ce modèle. Cette opération permet de renvoyer un score de confiance sur la région extraite modélisant la probabilité que la région contienne ou pas un objet. La manière d'extraire les régions est discriminante pour la rapidité et l'efficacité du détecteur : si le modèle est appliqué à toutes les régions possibles dans l'image, le temps de calcul devient énorme. [24]

2.3 Méthodes de détection

Dans cette partie, nous expliquons les divers algorithmes de détection. L'approche et les caractéristiques saillantes de chacun méthode. Il existe plusieurs méthodes de détection comme R-CNN, Faster R-CNN, Fast R-CNN, YOLO, SSD, Segmentation d'instance, Mask R-CNN et d'autres. Nous mentionnerons les suivants :

a. Méthode de R-CNN

R-CNN qui repose sur une détection entièrement réalisée avec des réseaux de neurones de convolution. Cet algorithme va se procéder comme suit : dans la première phase le réseau de neurones de convolution prend en entrée une image de taille quelconque ensuite il va produire des régions d'intérêts dans lesquelles les R-CNN pourraient les détecter. Ces régions seront envoyées au second réseau afin d'identifier la présence de l'objet dans cette région voir la Fig 2.2 [25]

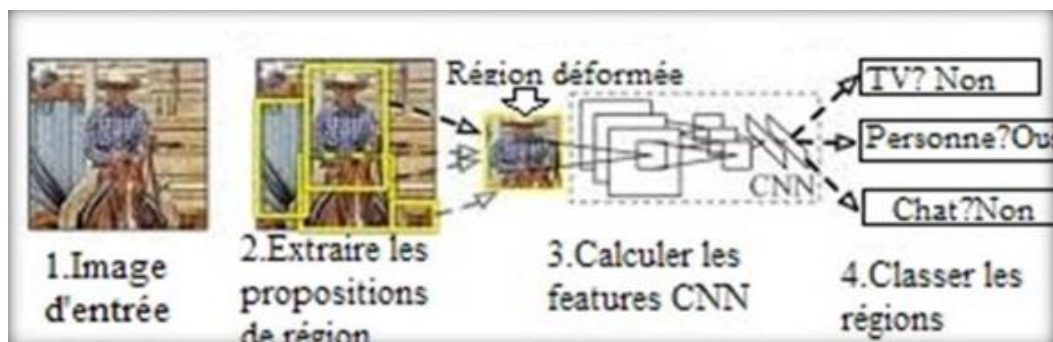


Fig2.2 Architecture de R-CNN. [25]

b. Méthode de YOLO

L'algorithme You Only Look Once (YOLO) est développé par J.Redmon en 2015[26]. Cet algorithme est utilisé pour la détection d'objets, il repose sur deux étapes : la première consiste à détecter les objets par des réseaux de neurones de convolution et la seconde correspond à quadriller l'image et la prédiction de la classe de l'objet détecté s'il existe. L'avantage de cette prédiction est qu'elle peut se faire indépendamment de la première détection mais qu'elle ne cible pas les objets en particulier. L'inconvénient de cette méthode est la difficulté de détecter les plus petits objets ainsi que les objets qui sont en chevauchement voir la **Fig2.3**[25]

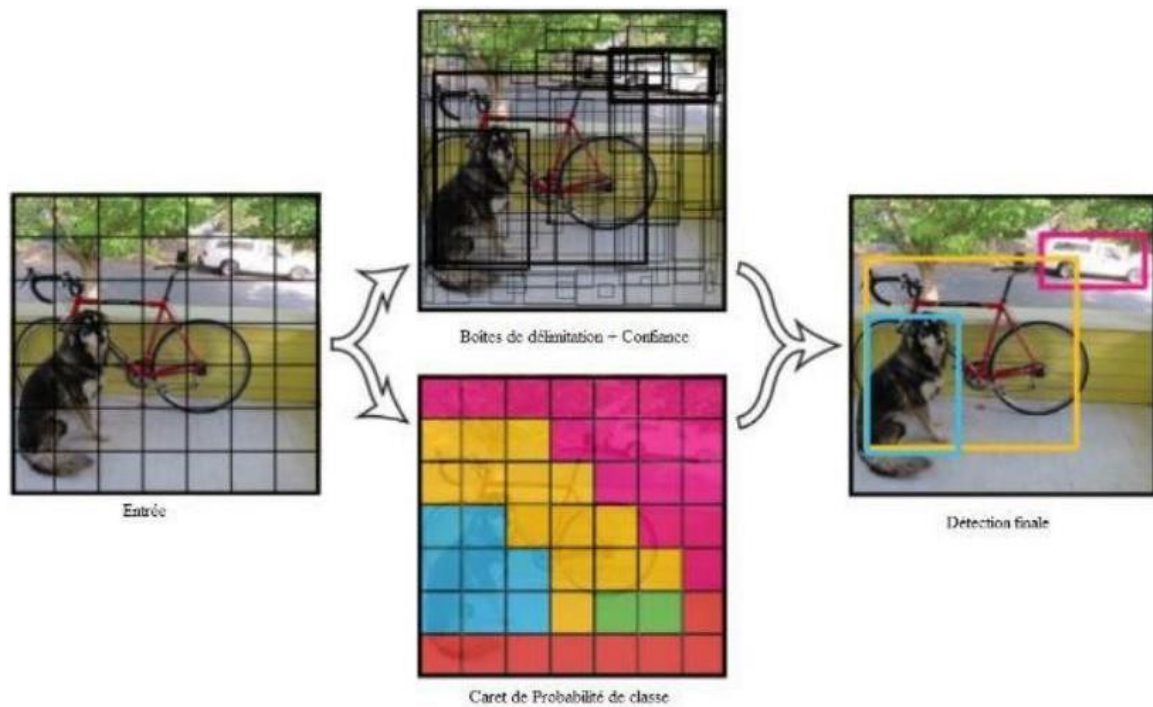


Fig2.3 Architecture de YOLO. [25]

c. Méthode de Fast R-CNN

Le même auteur de l'article précédent (R-CNN) a résolu certains des inconvénients de R-CNN pour construire un algorithme de détection d'objets plus rapide et il a été appelé Fast R-CNN. L'approche est similaire à l'algorithme R-CNN. Mais, au lieu d'alimenter les propositions de la région à la CNN, nous donnons l'image d'entrée à la CNN pour générer une carte de fonctionnalité convolutionnelle. à partir de la carte des fonctionnalités convolutionnelles, nous identifions la région des propositions et les déformons en carrés et en utilisant une couche de mise en commun RoI nous les remodelons en une taille fixe afin qu'elle puisse être introduite dans une couche entièrement connectée. à partir du vecteur de fonctionnalités RoI, nous utilisons une couche softmax pour prédire la classe de la région proposée ainsi que les valeurs de décalage pour la zone de délimitation.

La raison pour laquelle « Fast R-CNN » est plus rapide que R-CNN est parce que vous n'avez pas à alimenter 2000 propositions de région au réseau neuronal convolucionnel à chaque fois. Au lieu de cela, l'opération de convolution n'est effectuée qu'une fois par image et feature map est générée à partir de celui-ci.[27]

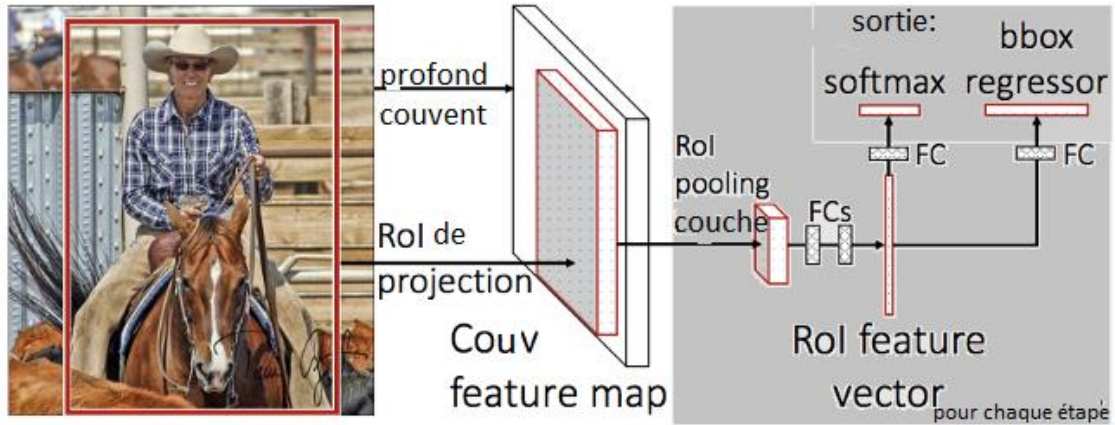


Fig2.4 Architecture de Fast R-CNN [27]

d. Méthode de Faster R-CNN

Le Faster R-CNN est une amélioration du R-CNN dans sa précision et sa rapidité à l'entraînement (Ren, He, Girshick & Sun, 2015).[28]

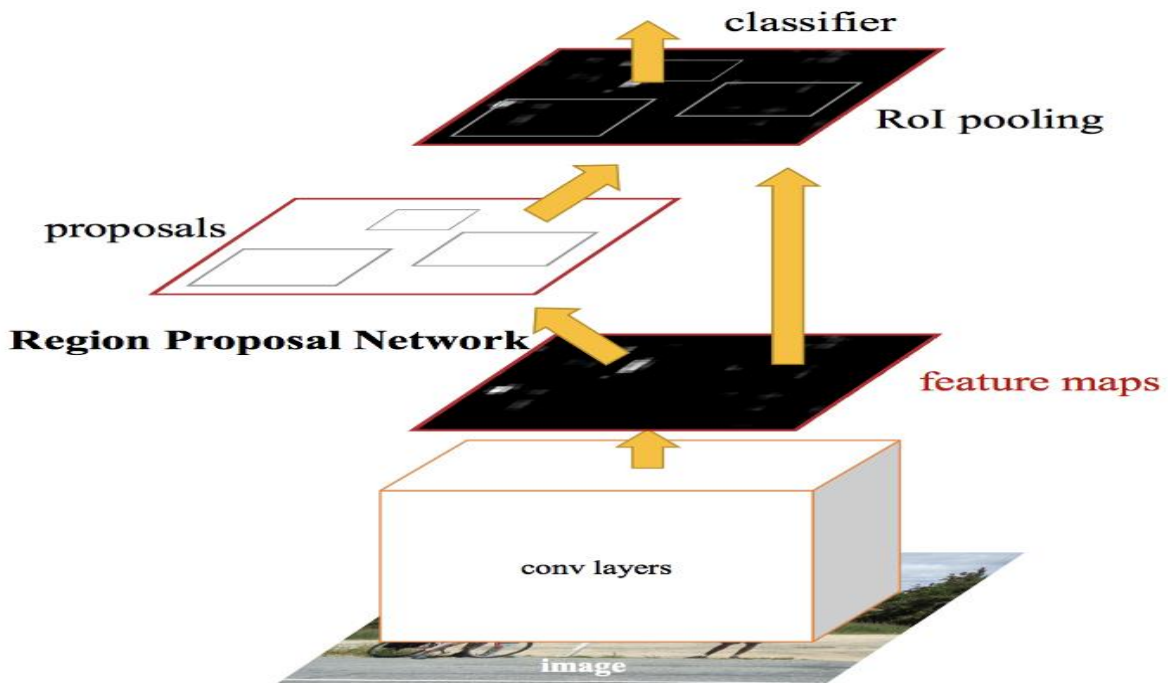


Fig2.5 Architecture de Faster R-CNN [28]

L'architecture Faster R-CNN (fig. 2.4) utilise la même carte de caractéristiques résultant des couches de convolution pour générer les régions intéressantes et pour ensuite les classifier. Le

réseau de propositions de régions utilise des fenêtres coulissantes (fig. 2.5) de tailles et ratio différents pour analyser la carte de caractéristiques en profondeur. Ces changements améliorent significativement la précision et la rapidité de l'architecture comparée au R-CNN [28].

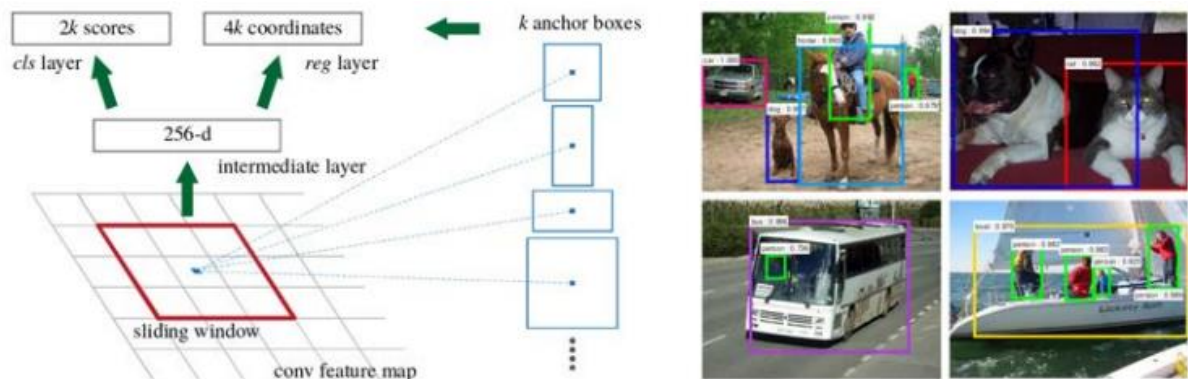


Fig2.6 Faster R-CNN : Fenêtre coulissante et résultat[28].

e. Méthode de SSD

L'article sur SSD: Single Shot MultiBox Detector (par C. Szegedy et al.) a été publié à la fin de Novembre 2016 et a atteint de nouveaux records en termes de performance et de précision pour les tâches de détection d'objets, la notation de plus de 74% mAP (moyenne précision) à 59 images par seconde sur les ensembles de données standard tels que PascalVOC et COCO. Pour mieux comprendre SSD, commençons par expliquer d'où vient le nom de cette architecture :

- **Single Shot** : cela signifie que les tâches de localisation et de classification d'objets se font en un seul passage vers l'avant du réseau
- **MultiBox**: c'est le nom d'une technique de régression de boîte de délimitation développée par Szegedy et al. (nous allons brièvement le couvrir sous peu)
- **Détecteur** : le réseau est un détecteur d'objets qui classe également les objets détectés.[29]
- **Architecture**

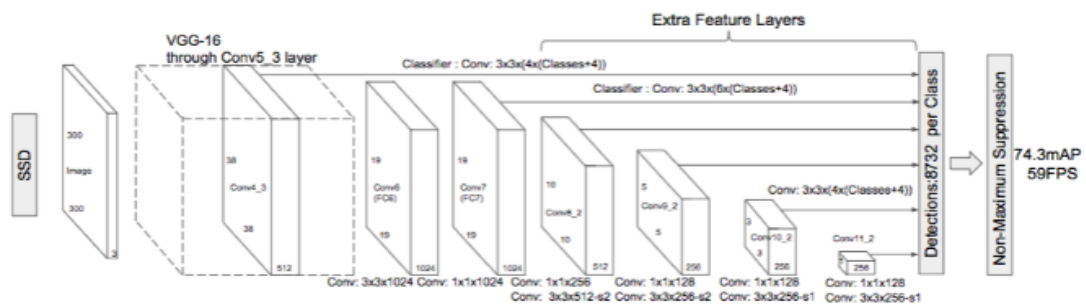


Fig2.7 Architecture of Single Shot MultiBox detector (input is 300x300x3) [29]

f. Mask R-CNN

Le Mask R-CNN est l'amélioration suivante après le Faster R-CNN. Le Mask R-CNN change la sortie du modèle final. En effet, l'architecture Faster R-CNN permet de localiser des objets distincts avec une boîte englobante. L'architecture Mask R-CNN permet, quant à elle, de segmenter chaque instance d'un objet avec un masque sémantique. Cette amélioration permet donc au modèle de faire de la segmentation d'instance (fig. 2.9) [30].

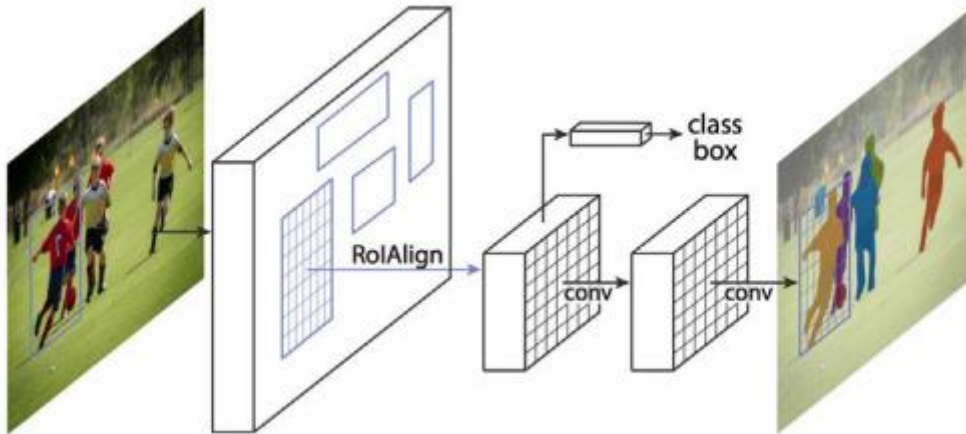


Fig2.8 Architecture Mask R-CNN [30].

Comme on peut le voir en Fig2.8, l'architecture Mask R-CNN est semblable à Faster R-CNN mais rajoute des couches en parallèle de la classification. Cette approche est différente d'autres architectures de segmentation d'instance qui classifiaient les résultats par rapport au masque généré. Cette approche en parallèle permet à chaque type d'objet de prédire leur masque sans être en concurrence avec les autres classes du modèle [30]

g. Segmentation d'instance

La segmentation d'instance est un problème de vision ordinateur. La segmentation d'instance cherche à identifier chaque instance, ou exemplaire, d'une classe recherchée à l'intérieur des données présentées. En vision d'ordinateur, la segmentation d'instance cherche à classer chaque pixel d'une image à une instance d'un objet[31]

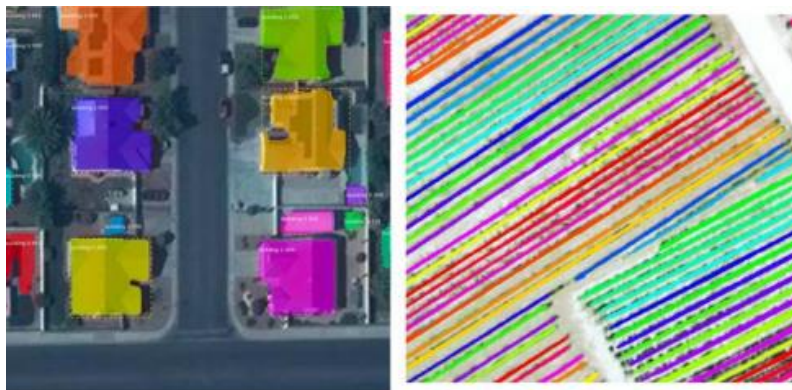


Fig2.9 Exemple de segmentation d'instance. A gauche, de bâtiments [31], à droite de vigne.

En **Fig2.9**, chaque bâtiment ou ligne de vigne est séparé avec un masque de prédiction. Les couleurs représentent chacune une instance différente. Cela permet de facilement identifier et visualiser chaque instance.[31]

3. La reconnaissance d'objets dans les images

3.1 Introduction

La reconnaissance d'image est un étape primordial pour la mise en œuvre de plusieurs application actuelles qui nécessite une interprétation de haut niveau d'image, dans ce chapitre on va présenter les approches et les méthodes principales de reconnaissance d'objet en général et évaluer leur applicabilité a notre problème, dans une deuxième étape on va parler des images satellites et finalement nous allons expliquer nos choix de méthodes pour le développement de notre système de reconnaissance d'objet cartographique cependant le principal problème de la reconnaissance d'image se résume sur le fait qu'elle ne peut pas décider si un objet spécifique ou un objet appartenant a une catégorie d'objet prédéfini existe dans une image.

3.2 Les principaux domaines d'application

En plus du problème global de la vision par ordinateur, la reconnaissance d'objets dans une grande variété de catégories est une étape importante pour de nombreuses applications spécifiques. Nous pouvons citer, entre autres, quelques exemples d'applications pour illustrer les difficultés de la recherche sur ce problème.

- La vidéo-surveillance : la reconnaissance et la détection automatique des personnes, des visages et des objets d'intérêt sur la très grande quantité d'images vidéo utilisées pour la surveillance.
- La recherche dans des bases d'images numériques : les systèmes actuels effectuent en général des requêtes à partir des mots clés associés aux images. On voudrait exploiter le contenu visuel des bases d'images sans avoir besoin de les indexer textuellement.
- La navigation des robots mobiles et des véhicules : on a besoin de reconnaître les obstacles, les objets divers dans l'environnement ou les autres véhicules pour la navigation automatique.

Pour pouvoir évaluer les différentes méthodes et pour pouvoir les comparer, il est nécessaire de créer des plateformes communes de test, nous voulons mentionner les bases d'images publiques les plus connues et les catégories d'objets répertoriés les plus communes:

- Bases d'images FERET, CMU/MIT, MIT-CBCL, Caltech, Yale : les visages ;
- Bases d'images ETHZ, Caltech, UIUC : les voitures, les motocyclettes ;
- Bases d'images Pascal, CMU : les bicyclettes ;

- Bases d'images Graz 02, Pascal : les personnes, les vaches ;
- Base d'images Weizmann : les chevaux ;[32]

3.3 Les défis majeurs de la reconnaissance d'objets

La principale difficulté de la reconnaissance d'objet est la variation d'un objet ou d'un objet dans la même catégorie sur différentes images. Les sources de variations peuvent être regroupées en sept catégories principales :

- Les variations de point de vue : les objets réels sont des objets physiques tridimensionnels. Leur apparence varie considérablement selon le point de vue.
- Les variations d'illumination : en fonction de changements de l'environnement et des conditions d'acquisition d'une image, un objet peut subir des variations d'illumination importantes
- Les occlusions : Dans l'image, les objets peuvent être partiellement déguisés par d'autres objets ou leurs propres composants.
- Les variations d'échelle : les objets d'une même catégorie peuvent avoir des tailles physiques différentes ou les images d'un même objet peuvent avoir des tailles différentes en fonction de la résolution de l'image
- Les déformations : la plupart des objets réels n'ont pas une forme rigide. Particulièrement les objets articulés subissent généralement des déformations de leurs composants.
- Complexité du fond : en général, les objets de l'image ne sont pas isolés de leur environnement. Dans un contexte complexe, il est difficile de faire la distinction entre les zones appartenant à un objet et un arrière-plan.[33]

3.4 Extraction des descripteurs

L'utilisation directe de l'image pour la reconnaissance n'est pas envisageable parce qu'elle contient une grande quantité d'information impertinente, la première étape de presque toutes les méthodes de reconnaissance est d'extraire un ensemble de descripteurs qui représentent les propriétés discriminantes de l'objet cible. Les descripteurs sont un ensemble de nombres scalaires générés pour décrire un objet, Un objet ne peut pas être reconstruit en général à partir des descripteurs, mais les descripteurs pour des objets distincts doivent avoir des valeurs suffisamment différentes pour les discriminer, et les descripteurs pour les objets similaires doivent avoir des valeurs proches, l'objectif est d'en sélectionner un nombre moins important que le nombre de pixels dans l'image. Pour une reconnaissance robuste il est souhaitable que les descripteurs soient

On peut distinguer deux types de descripteurs :

- Les descripteurs globaux sont obtenus à partir de l'image entière. Par contre,
- Les descripteurs locaux caractérisent un ensemble de régions locales significatives de l'image qui sont riches en information

Ci-dessous, nous présentons les principaux types de descripteurs et les méthodes utilisées pour les extraire

i. Les points d'intérêt

Les points d'intérêt sont des régions de l'image riches en termes de contenu de l'information locale et stables sous des transformations affines et des variations d'illumination. Ils sont des indicateurs des régions susceptibles de contenir un objet, et en même temps des parties importantes de l'objet.

a) Détection des coins

Le détecteur des coins de Harris et Stephens est l'un des plus populaires des détecteurs de points d'intérêt. Un coin, l'intersection de deux contours, a un contenu informatif important. Sur l'image les coins correspondent aux pixels qui ont des changements d'intensité élevés dans toutes les directions sur un voisinage local, L'idée principale est donc de trouver les points à partir desquels un petit déplacement donne un changement d'intensité élevée, La fig 2.10 montre les coins détectés sur une image satellitaire.[34]

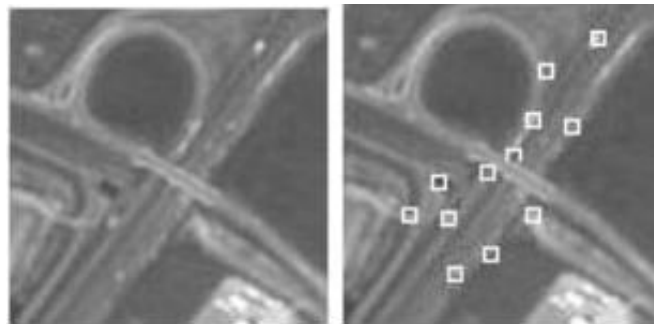


Fig2.10 Les coins détectés sur une image satellitaire par le détecteur de Harris et Stephen [34]

b) Les descripteurs SIFT

Schmid et Mohr [35] ont proposé de calculer un descripteur local pour chaque point d'intérêt. Le descripteur local est calculé à l'aide des invariants différentiels de l'intensité des pixels [36] sur un voisinage local autour du point d'intérêt. La **Fig 2.4** illustre l'approche utilisée. L'association des descripteurs locaux aux points d'intérêt rend possible un nombre d'appariement correct élevé entre deux images qui contiennent le même objet. Cela assure la robustesse de la reconnaissance d'objets.

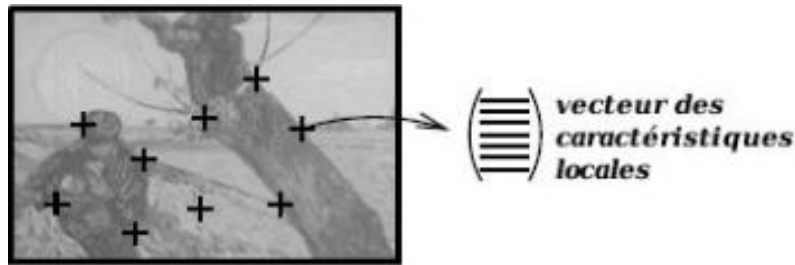


Fig2.11 Les points d'intérêt et les descripteurs locaux, dans Schmid et Mohr [35]

ii. Les descripteurs contours

Les contours d'un objet donnent suffisamment d'information sur sa forme. D'après la définition du dictionnaire Le Petit Robert [37] le mot "forme" signifie :

Forme : Ensemble des contours d'un objet, d'un être, résultant de la structure de ses parties.

L'œil humain est capable de reconnaître un objet à partir des fragments de ses contours

(Fig2.12)

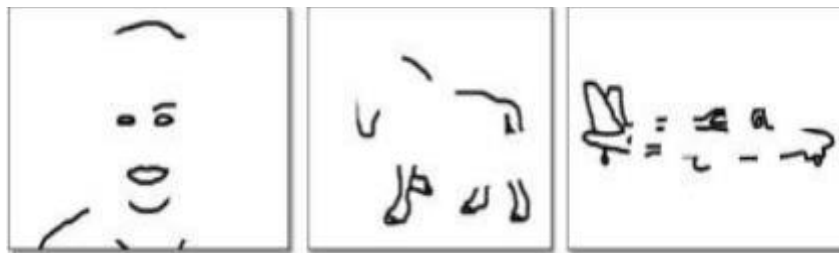


Fig2.12 – Les fragments de contours de quelques objets courants, dans Shotton et al. [38].

Les descripteurs obtenus à partir des contours ont l'avantage d'être invariants aux changements d'intensité, de texture et de couleur de l'image. Les descripteurs contours sont des candidats idéaux pour représenter les propriétés structurelles des objets

a) Les fragments de contours

Shotton et al. [38] propose d'utiliser des fragments de contours comme descripteurs. A partir d'un petit nombre d'images d'objets segmentées on construit un dictionnaire des fragments. Le dictionnaire est constitué des fragments des contours extérieurs de l'objet et des positions des fragments par rapport au centre de l'objet

b) Les primitives géométriques

Une alternative pour obtenir des descripteurs à partir des contours est de rechercher des formes qui ont des propriétés géométriques bien définies. Les objets construits par l'homme sont en général constitués de composants qui ont des formes géométriques simples, contrairement aux objets naturels qui ont des formes plus irrégulières.[39]

iii. Les descripteurs régions

Le but de ces approches est d'identifier les zones qui appartiennent à l'objet et ont une forme. On peut ensuite représenter les régions détectées par un ensemble de descripteurs de formes.[33]

a) La segmentation de l'image

La segmentation est la partition d'une image en zones homogènes. Elle a été depuis toujours un des principaux problèmes de la recherche en vision par ordinateur et de nombreux algorithmes de segmentation ont été développés.

Nous voulons donner une taxonomie générale des méthodes de segmentation. Pour une présentation détaillée des méthodes de segmentation on peut consulter [40].

On effectue en général la segmentation en utilisant quatre approches principales :

- Les approches globales utilisent l'histogramme des intensités des pixels et appliquent un seuillage pour classifier les pixels de l'image.
- Les méthodes par modèle ou motif (Template) recherchent dans l'image des régions qui correspondent à des motifs prédéfinis. [41].
- Les approches frontières se servent des contours de l'objet pour détecter une région. L'extraction des contours est suivie des algorithmes de fermeture des contours par la relaxation des contours, recherche dans un graphe des contours.
- Les méthodes de clustering, divisent l'image en régions en regroupant les pixels voisins qui ont des niveaux d'intensités proches ou qui satisfont d'autres critères numériques.[33]

b) La segmentation multi-échelle

Nous nous intéressons particulièrement aux méthodes de clustering qui donnent la possibilité de segmenter une image en plusieurs niveaux d'échelles : en haut de l'échelle, on obtient une segmentation grossière avec peu de régions et en bas de l'échelle une segmentation détaillée avec un grand nombre de régions. On ne peut pas parler d'une échelle unique de segmentation qui représente une image. Les régions obtenues aux différentes échelles correspondent aux objets ayant des tailles plus petites ou plus grandes.

Les approches ascendantes utilisent une sur-segmentation de l'image et fusionnent itérativement les régions d'après une métrique de similarité. On utilise le plus souvent la similarité radiométrique des pixels comme métrique. On peut également utiliser la texture, la couleur, ou les propriétés géométriques des régions [42].

La **Fig2.13** illustre un exemple de segmentation repose sur l'algorithme Watershed.[43]

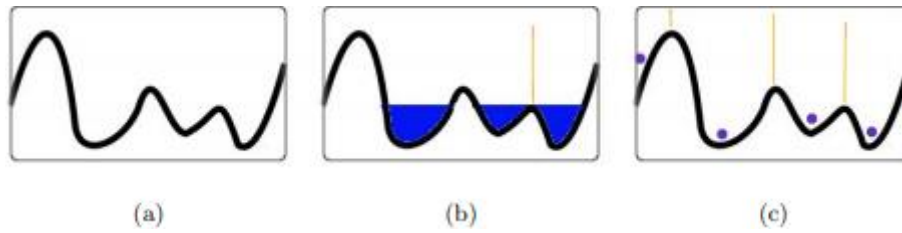


Fig2.13 – Illustration de l'algorithme Watershed sur un signal 1D. (a). L'image, (b). L'immersion et la première ligne de partage, (c). Segmentation finale, on obtient 4 régions séparées par 3 lignes de partage.[33]

iv. Les descripteurs de formes

Une fois l'image segmentée, les zones résultantes peuvent être représentées par des descripteurs de formes. Une analyse très détaillée des descripteurs de formes. est donnée à Zhang et Lu[44]. Nous présentons la forme comme une zone ou ses contours pour extraire les descripteurs [45]

a) Les descripteurs obtenus par une approche région

Pour calculer ces descripteurs la forme est considérée comme l'ensemble des pixels qui constituent sa région interne.

Les descripteurs géométriques simples :

- L'aire, le périmètre ;
- La compacité ou la circularité : le rapport du carré du périmètre à l'aire de la forme (Perimètre $2 \cdot 4 \cdot \pi \cdot \text{Aire}$).;
- La rectangularité : c'est le rapport de l'aire de la forme à l'aire du plus petit rectangle englobant ;
- L'excentricité : la mesure de la circularité d'une ellipse. Pour une ellipse c'est le rapport de la distance entre les foyers à la longueur de l'axe majeur.;
- L'élongation : le rapport de la longueur à la largeur du plus petit rectangle englobant ;
- Les axes principaux : l'axe majeur et l'axe mineur de l'ellipse qui a les mêmes moments d'ordre deux que la forme ;
- L'orientation : la direction de l'axe majeur.

Les descripteurs topologiques : Deux formes sont topologiquement identiques si on peut obtenir une en déformant l'autre par des transformations continues. Les descripteurs topologiques de base sont le nombre de composants connexes et les trous.

L'enveloppe convexe : Une région R est convexe si pour chaque paire de points p_1, p_2 de R le segment $[p_1 p_2]$ est dans R. L'enveloppe convexe est la plus petite région convexe qui contient la forme.

Les profils : On peut obtenir une signature de l'objet en le projetant sur l'axe horizontal ou vertical et en comptant le nombre de pixels sur chaque colonne ou ligne. [46].

Les moments : La fonction d'intensité de la forme peut être représentée comme une densité de probabilité d'un variable aléatoire bi-dimensionnelle:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Les moments centraux sont des moments géométriques invariants à la translation. On les obtient en utilisant comme origine le centre de gravité de la forme (x_c, y_c) . , on calcule les moments centraux normalisés qui sont invariants à l'échelle et à la translation. Hu [47] a défini 7 moments invariants à la rotation à partir des combinaisons des moments centraux normalisés jusqu' à l'ordre trois. Les moments de Zernike [48] sont construits en utilisant une famille de polynômes complexes

Le squelette : Le squelette de l'objet est un ensemble de segments connexes d'épaisseur un pixel obtenu par l'amincissement d'une forme. On l'appelle également l'axe médian, l'ensemble des points qui ont la même distance minimale aux frontières de la forme pour au moins deux points de frontière distincts.

b) Les descripteurs obtenus par une approche contours

Ce sont des descripteurs obtenus à partir des contours externes de l'objet.

Les descripteurs géométriques simples :

- Le périmètre ;
- La courbure : le contour est représenté par l'angle de la tangente à chaque point de contour. On peut ensuite calculer la dérivée des courbures sur chaque point de contour ou on peut détecter les courbures importantes du contour comme descripteur;
- L'énergie de flexion (bending) : la somme des carrées des courbures à chaque point de contour ;
- La distribution des cordes : une corde est un segment de droite qui joint deux points de contours.

Le codage de chaîne : A partir d'un point de contour initial, on traverse le contour en codant les points consécutifs par la direction du déplacement. Les chiffres de 0 à 7 sont utilisés pour coder les 8 directions principales.

Approximation polygonale : Les contours de la forme peuvent être représentés par des segments de droites obtenus par une approximation polygonale.

Shape contextes : C'est une façon de représenter le contour par rapport à un point de contour choisi. On prend n points sur le contour. Pour chaque point p_i on construit les vecteurs qui joignent p_i à ces points. Le Shape contexte de p_i est l'histogramme des coordonnées relatives des $n - 1$ points.[33]

4. La Reconstruction 3D

4.1 Introduction :

Dans la vision par ordinateur et l'infographie, la reconstruction 3D est le processus de capture de la forme et de l'apparence d'objets réels. Ce processus peut être réalisé soit par des méthodes actives ou passives. Si le modèle est autorisé à changer sa forme dans le temps, cela est appelé reconstruction non rigide ou spatio-temporelle[49]

4.2 Domaines d'applications de la reconstruction 3D :

Les applications pouvant bénéficier de la reconstruction 3D, sont toutes les applications usuelles de la synthèse d'images : CAO 1 , Réalité Virtuelle2 , Réalité Augmentée, entraînement, simulation, commerce, loisirs, compression de données, et beaucoup de recherches sont en cours dans ces sens [50].

4.3 Reconstruction 3D à partir d'une image :

La reconstruction d'objets 3D à partir d'images individuelles a de vastes applications dans le multimédia. La plupart des méthodes connexes existantes ne récupèrent que des objets 3D bruts et les objets doivent souvent être interconnectés. Dans cet article, nous proposons une nouvelle méthode qui utilise un ensemble de grilles de référence auxiliaires pour reconstruire précisément des objets 3D à partir d'une seule image non calibrée. Dans notre système, l'utilisateur dessine d'abord les dessins en ligne des objets. Ensuite, la longueur focale initiale f de la caméra est calculée avec une méthode d'étalonnage, puis la longueur focale initiale est affinée par une grille de référence. Avec le f raffiné, un environnement de mesure de position 3D est construit, et un système de coordonnées du monde est défini par l'utilisateur. Après cela, un ensemble de grilles de référence sont utilisés pour trouver les emplacements 3D précis des points d'objet et les cadres métalliques des objets sont récupérés automatiquement. Enfin, le système génère les surfaces et rend les objets 3D complets. Outre la modélisation 3D précise, notre méthode de reconstruction ne nécessite pas que les objets d'une scène soient interconnectés. Un ensemble d'exemples sont fournis pour démontrer la capacité de manipuler des objets polyèdres complexes et des surfaces courbes dans un même cadre.[51]

4.4 Reconstruction 3D à partir de plusieurs images :

Est la création de modèles en trois dimensions à partir d'un ensemble d'images. Il est le processus inverse de l'obtention d'images en 2D à partir de scènes 3D.

L'essence d'une image est une projection à partir d'une scène 3D sur un plan 2D, au cours du processus dont la profondeur est perdue. Le point 3D correspondant à un point d'image spécifique est contraint d'être sur la ligne de visée. à partir d'une seule image, il est impossible de déterminer quel point cette ligne correspond au point d'image. Si deux images sont disponibles, la position d'un point 3D se trouve l'intersection des deux rayons de projection. Ce processus est appelé triangulation. La clé de ce processus est les relations entre plusieurs vues qui véhiculent les informations que les ensembles de points correspondants doivent contenir une certaine structure et que cette structure est liée à la pose et l'étalonnage de l'appareil.

Dans les dernières décennies, il y a une demande importante pour le contenu 3D pour l'infographie, la réalité virtuelle et de la communication, ce qui déclenche un changement d'orientation pour les besoins. De nombreux systèmes existants pour la construction de modèles 3D sont construits autour d'un matériel spécialisé (par exemple des plates - formes stéréo) entraînant un coût élevé, ce qui ne peut pas satisfaire à l'exigence de ses nouvelles applications. Cet écart stimule l'utilisation des moyens d'imagerie numérique (comme un appareil photo). La loi de Moore nous dit aussi que plus de travail peut être fait dans le logiciel. Une première méthode a été proposée par Tomasi et Kanade. Ils ont utilisé une approche factorisation affines pour extraire des séquences d'images 3D. Toutefois, l'hypothèse de projection orthographiques est une limitation importante de ce système [52]

4.5 Reconstruction 3D basé sur l'apprentissage profond :

[1] présentent une solution de reconstruction d'objet dite Mesh RCNN basée sur l'utilisation de Mask RCNN avec une branche de prédiction de maillage qui produit des maillages avec une structure topologique variable en commençant par prédiction de représentations grossières de voxels qui sont converties aux maillages et raffinés avec un réseau de convolution graphique fonctionnant sur les sommets et les arêtes du maillage.

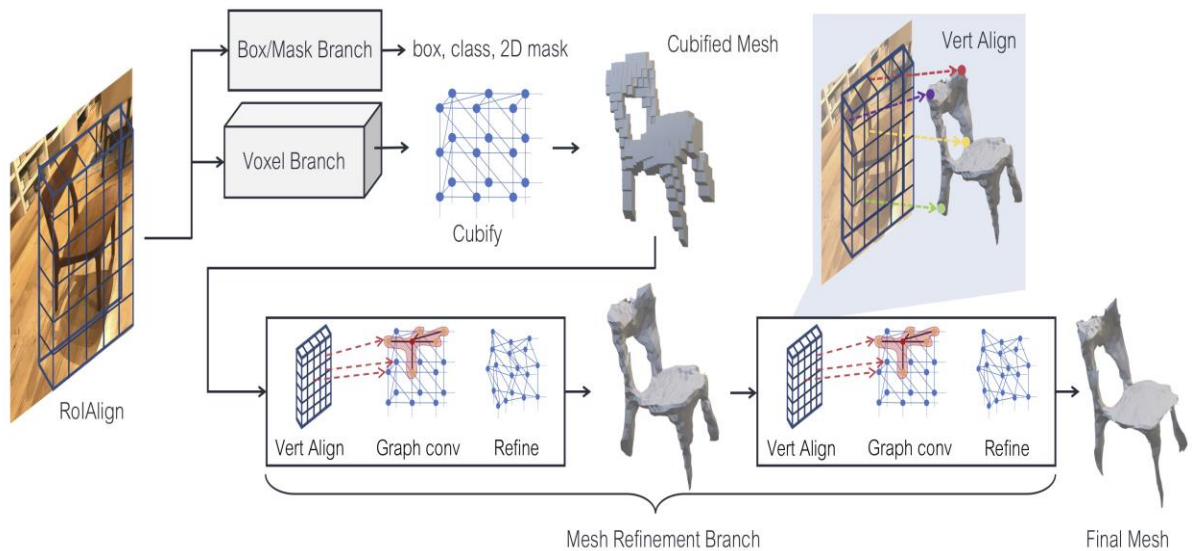


Fig2.14 Architecture du modèle Mesh RCNN

[53] proposent une nouvelle méthode qui utilise un ensemble de grilles de référence auxiliaires pour reconstruire avec précision des objets 3D à partir d'une image unique non calibrée. Son principe consiste dans un premier temps de tracer d'abord la ligne dessin des objets. Ensuite, la distance focale initiale de la caméra est calculé avec une méthode d'étalonnage, un ensemble de grilles de référence est utilisé pour trouver les emplacements 3D précis des points de l'objet enfin, le système génère le surfaces et effectue un rendu des objets 3D complets.

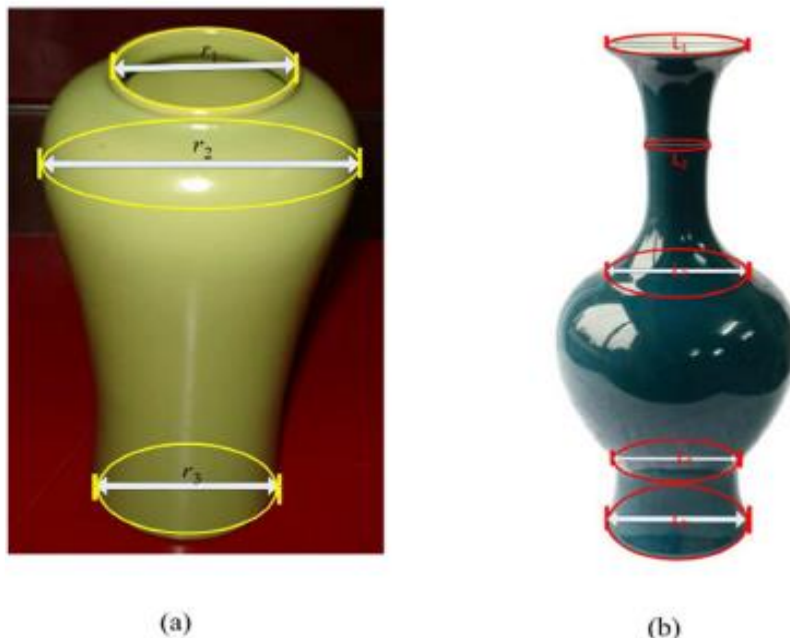


Fig2.15 mesure de paramètres avec la grille de référence.

[61] proposent une architecture de réseau de neurones convolutifs et une technique d'apprentissage de la préparation des données qui fournissent une prédiction du modèle voxel

d'une scène 3D avec plusieurs objets. Pour l'apprentissage et l'évaluation de CNN, un ensemble de données a été collecté et annoté. Il contient des séquences d'images de plusieurs scènes et des images de profondeur correspondantes et des modèles 3D de ces scènes. La séquence d'images sert de données primaires utilisées pour la reconstruction 3D de la scène par la technique SfM. Le traitement de la structure à partir du mouvement entraîne des modèles 3D de surface de tous les objets de la scène, ainsi que les positions et l'orientation de la caméra pour chaque image d'une séquence. Ensuite, le modèle 3D de surface est transformé en modèle 3D voxel et segmenté en objets séparés. Une architecture de réseau antagoniste générative conditionnelle a été développée pour la reconstruction 3D par image unique. Sa partie générative traduit une image couleur d'entrée en un modèle de voxel de sortie. La partie discriminante distingue la sortie correcte de la fausse sortie. Les deux parties sont entraînées simultanément sur l'ensemble de données préparé. L'évaluation de la partie test de l'ensemble de données préparé a démontré la capacité de prédire des modèles 3D de scènes complexes précédemment non observées contenant plusieurs objets. L'architecture de réseau neuronal proposée offre une capacité de généralisation élevée et une résolution améliorée des modèles 3D de voxel prédits.

5. Conclusion

Nous avons présenté dans ce chapitre la détection d'objets et les différentes méthodes de détection et la reconnaissance 3D et enfin la reconstruction 3D. Nous montrerons dans le chapitre suivant (Implémentation) notre approche qui nous avons utilisé pour la détection et la reconstruction des objets 3D basée sur les réseaux de neurones Mask R-CNN.

Chapitre III :

Implémentation

1. INTRODUCTION

Dans ce chapitre, nous présentons l'architecture de notre modèle, également nous détaillerons chaque partie de ce dernier.

2. Notre Modèle

2.1 L'Architecture

L'objectif de notre travail est de créer un modèle capable de reconstruire un objet à partir d'une image 2D et cela en respectant la contrainte temps réel. En effet afin d'aboutir à l'objectif de notre travail, on a suivi l'architecture suivante :

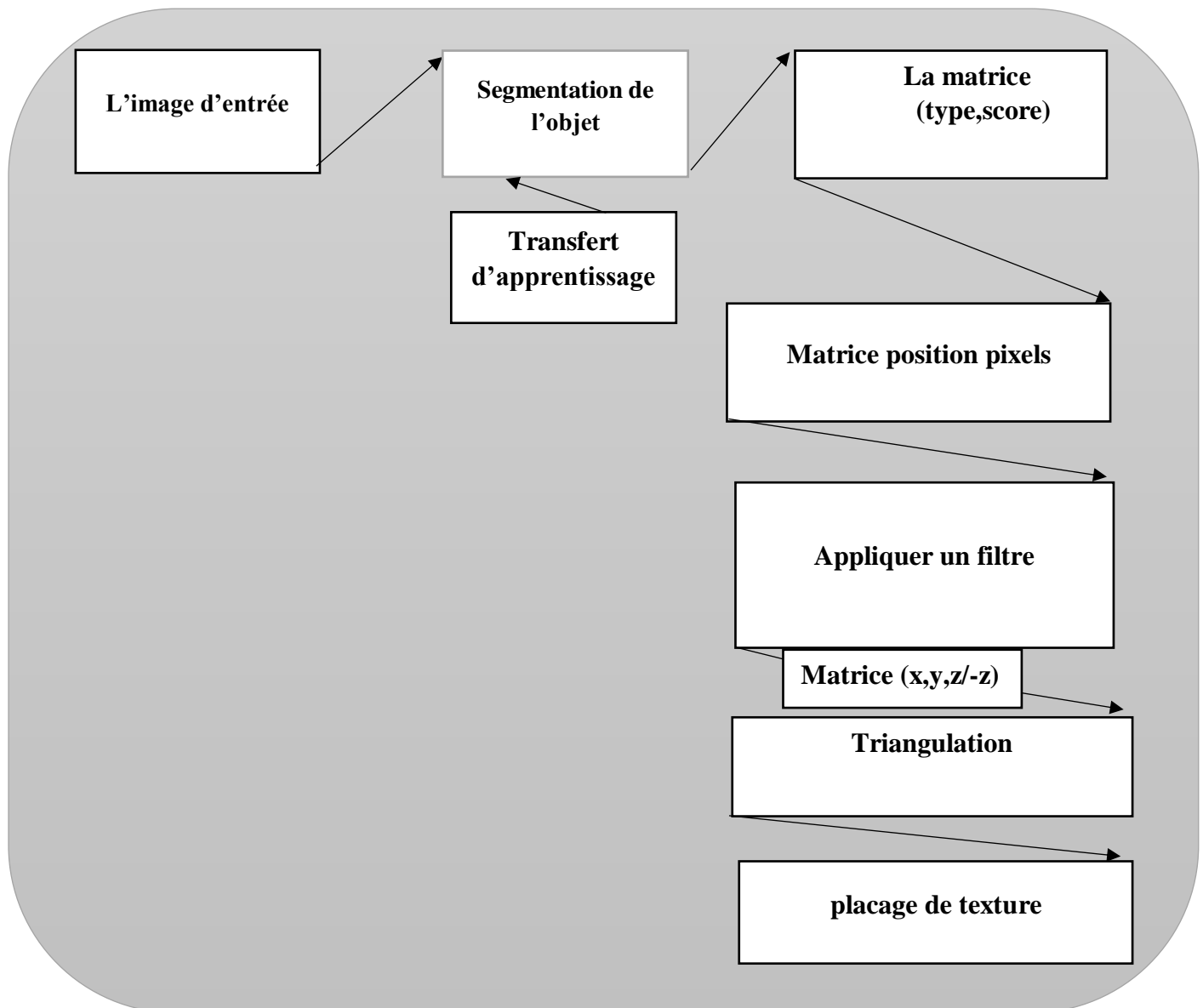


Fig3.1 L'architecture de notre méthode.

2.2 L'image d'entrée

L'image d'entrée : l'image d'entrée de notre modèle est une image 2D RGB, notre modèle opère seulement sur les objets dont la géométrie est symétrique, la figure ci-dessous présente quelques objets des images d'entrée.



Fig3.2 Image d'entrée contient plusieurs objets

2.3 La Segmentation de l'objet

Son principe consiste à classifier chaque pixel en fonction de la classe de l'objet auquel il appartient, en effet cette étape a pour objectif de localiser les pixels correspondants à notre objet dans l'image d'entrée.

Dans ce stade, nous avons utilisé la Bibliothèque Facebook qui a été proposée dans [55] pour effectuer une segmentation d'instance cela afin de ne pas fusionner tous les objets de la même classe et avoir une segmentation précise de chaque objet de la même classe. En effet cette bibliothèque est fondée sur l'architecture Mask-RCNN

Cette étape aura comme sortie un rectangle qui entoure l'objet et une probabilité d'estimation et également un Mask de pixels correspondant à cet objet.

La figure suivante présente l'architecture du masque R.CNN

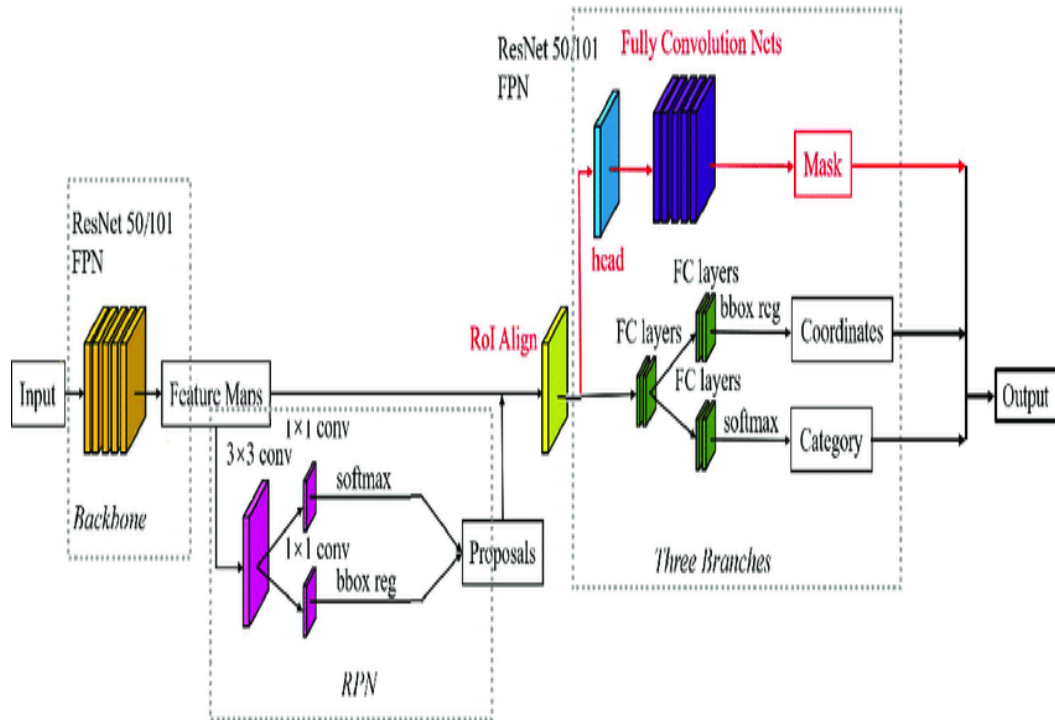


Fig3.3 Architecture masque R-CNN [56]

Par exemple, la figure ci-dessous montre une segmentation d'instance d'un objet bouteille.



Fig3.4 la segmentation de l'objet bouteille

2.4 La matrice (type,score)

L'exécution de la segmentation d'image va nous donner une table qui contient des informations sur les différents objets qui existent dans l'image d'entrée qui sont leurs classes ainsi le

pourcentage de reconnaissance de l'objet plus le cadrage de l'objet .et la matrice true or false de chaque objet et sa matrice qui a les mêmes dimensions de l'image d'entrée. Elle contient deux valeurs true or false : true l'objet cible, false : le fond de l'image.

```
Boxes : Boxes(tensor([[233.8538, 97.9705, 331.5380, 388.7406]]))
Scores : tensor(0.9932)
Classes : tensor(39) bottle
Masks : tensor([[False, False, False, ..., False, False, False],
                [False, False, False, ..., False, False, False],
                [False, False, False, ..., False, False, False],
                ...,
                [False, False, False, ..., False, False, False],
                [False, False, False, ..., False, False, False],
                [False, False, False, ..., False, False, False]])
```

Fig3.5 matrice contient des informations sur l'objet (score et true or false de l'objet cible..)

2.5 Matrice position pixels :

Faire correspondre chaque valeur true une matrice par l'indice X et Y de chaque pixel

```
tensor([[False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, True, True,
        True, True, True, True, True, True, True, True, True, True,
        True, True, True, True, True, True, True, True, True, True,
        True, True, True, True, True, True, True, True, True, True,
        True, True, True, True, True, True, True, True, True, True,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False])
```

Fig3.6 matrice true or false de l'objet cible

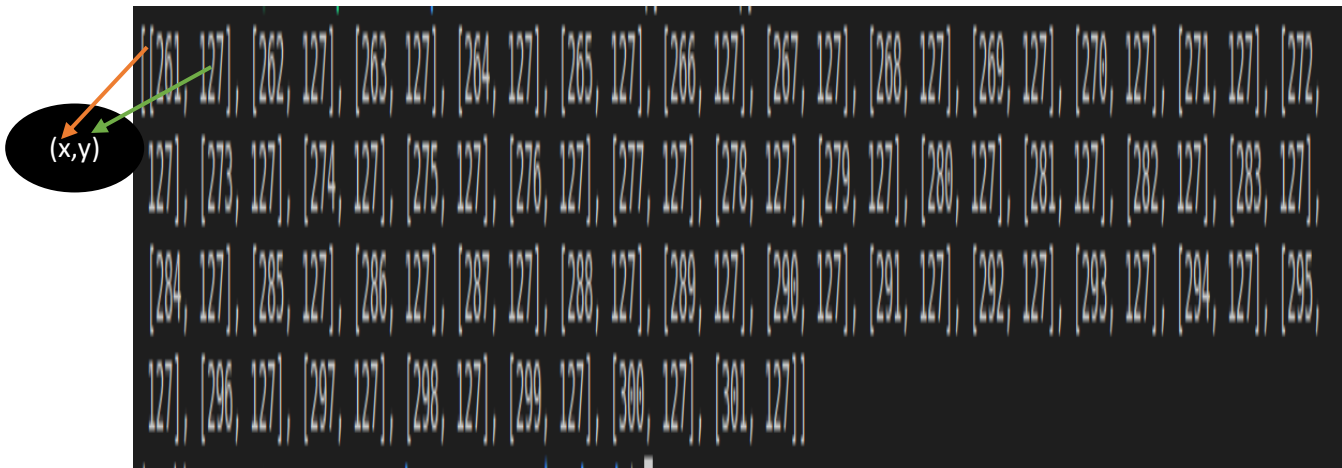


Fig3.7 matrice de X et Y de chaque pixel de l'objet cible

2.6 Appliquer un filtre :

Après plusieurs testes, nous avons effectué une étude de comparaison entre ces derniers, et comme résultat, nous avons trouvé que le filtre (22*30) est le plus efficace pour générer les points de reconstruction 3D, pour plusieurs raisons :

- Ce filtre quel que soit la taille d'objet.
- Ce filtre peut-être utiliser pour des objets de plusieurs formes symétriques.
- Ce filtre garde la même forme d'objet à étudier.

Maintenant, nous allons expliquer l'utilité d'utiliser ce filtre :

- On calcule la taille de l'axe Y (objet cible) après on divise par 30 et en approximant vers l'unité. Puis, On sauvegarde le résultat dans un variable a, après on sélectionne la 1 ère ligne Y(0) et Nous prenons le premier élément X(0) et le dernier élément X(22) et on calcule la distance entre ci deux derniers. Le résultat a été divisé sur 22 puis sauvegardé dans un variable b, On suppose que le premier point a Z(0)=0 et le dernier point a Z(22)=0.
- Les autre éléments de X se calculent comme suit :
 $X(n-1)+b=X(n)$ en même temp on va calculer la valeur de Z de cette point [X(1),Y(0)] La méthode de calcul de Z (La loi de calcul des distances entre deux points dans l'espace)
- On applique la même méthode pour le reste des points on s'arrête dans la point [X(21),Y(0)]
- Le résultat de cette étape est un tableau qui contient des points [(X(0),Y(0),Z(0))...(X(22),Y(0),Z(22))]
- Ces points que nous avons conclu dans cette étape présente la 1 ère ligne de façade Avant de l'objet
- Pour l'extraction de la 1 ère ligne de la façade arrière on supprime le premier et le dernier point de la façade présidente et le reste des points, nous multiplions leur (z *-1).
- La concaténation des deux tableaux précédents nous donne la première ligne d'objet 3D.

- Les éléments de Y se calculent comme suit : $Y(n-1)+a=Y(n)$
- Pour le reste des lignes d'objets on applique la même méthode.

2.7 Triangulation

- Dans cette étape nous allons lier chaque 3 points sous forme d'un triangle
- La liaison des points est dépendante des indices de chaque point, en appliquant cette opération sur tous les points.
- Nous allons utiliser le résultat d'étape de filtrage précédente (matrice 44*31 éléments)
- La liaison des points va être comme suit :

(i) est l'indice de la ligne et (j) est l'indice de la colonne

Les points de 1^{er} triangle (i,j),(i+1,j),(i+1,j+1)

Les points de 2^{émé} triangle (i,j),(i,j+1),(i+1,j+1)

...

Cas spécial j=43

Triangle (i,j),(i+1,j),(i+1,0)

Triangle (i,j),(i,0),(i+1,0)

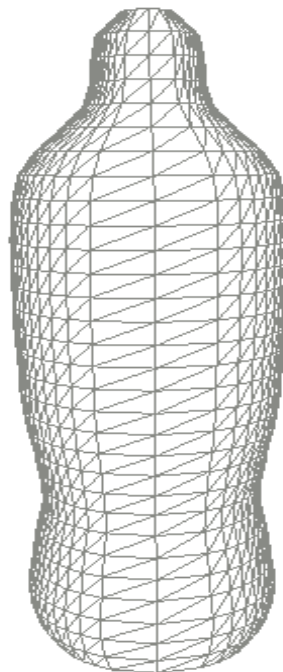


Fig3.8 la Triangulation de l'objet cible (Bouteille) sans texture

2.8 Placage de texture :

❖ Dans cette étape :

Nous comptons les variables entre 0 et 1 ($X(n)$ divisé par la dimension de X de la photo et la même chose pour Y)

Dans ce stade, nous avons utilisé la bibliothèque « three.js », qui va faire la texturage (Dans notre cas l'objet est une bouteille)

Et puis nous envoyons à la Bibliothèque « three.js » seulement les deux variables (X et Y) et la photo original puis la Bibliothèque va donner une image 3D .

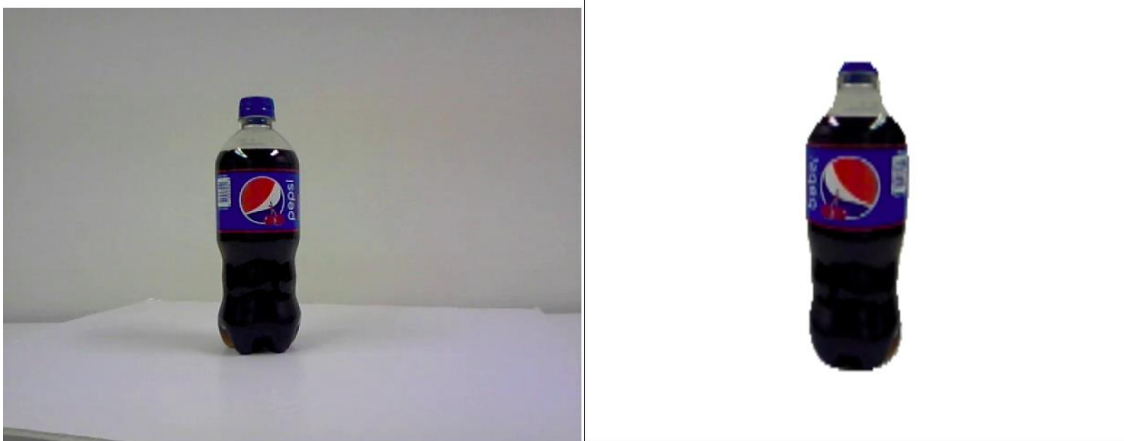


Fig3.9 Placage de texture de l'objet cible (Bouteille)

3. Conclusion

Nous avons présenté dans ce chapitre l'architecture de notre méthode pour la reconstruction des objets 3D à partir d'une image 2D, Nous avons détaillé chaque partie de notre méthode, notre méthode consiste dans un premier temps d'effectuer une segmentation d'instance cela permet de localiser les pixels de l'objet à partir d'un Mask de pixels. En effet l'utilisation d'un filtre sur matrice de sortie à permet de récupérer la dimension de profondeur, notre méthode simple à mettre en œuvre est efficace pour les objets dont la forme est symétrique. Dans le chapitre suivant nous présentons les résultats obtenus par notre méthode.

**Chapitre IV :
Résultat**

1. Introduction

Dans ce chapitre, nous allons présenter dans la section qui suit la configuration du matériel ainsi que les bibliothèques et le langage utilisés, puis nous allons présenter les résultats obtenus par notre méthode.

2. LA CONFIGURATION DU MATERIEL UTILISE

Afin de réaliser ce projet, nous avons utilisé un ensemble de matériels dont les principales caractéristiques sont les suivantes :

- Processeur: Intel(R) Core (TM) i3-4005U(1.7GHz .3MB L3 Cache)
- RAM : 4 GO.
- Système d'exploitation : Windows 10 64bits.

3. L'ENVIRONNEMENT DE TRAVAIL



3.1 Python

Python est un langage de programmation puissant de haut niveau, à la fois facile à apprendre, il prend en charge plusieurs modèles de programmation (procédural, fonctionnel et orienté objet). Les bibliothèques (packages) de Python encouragent la modularité et la réutilisabilité des codes existants. Python et ses bibliothèques sont disponibles sans difficulté pour la majorité des plateformes et il peut être redistribué gratuitement. On estime que c'est l'un des langages de programmation les plus utilisés au monde [57]

3.2 JavaScript & NodeJS

➤ Node.js (Node)  est une plateforme de développement open source pour l'exécution du côté du serveur de code JavaScript.

Node.js est destiné à s'exécuter sur un serveur HTTP dédié et à utiliser un seul thread avec un processus à la fois. Les applications Node.js sont basées sur des événements et s'exécutent de façon asynchrone.[58]



➤ JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js ou de Deno. Il s'exécute à l'aide d'un programme spécial appelé "Moteur Javascript". C'est un langage orienté objet à prototype.[59]

3.3 Pytorch



PyTorch est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch (en) développée par Facebook.

PyTorch permet d'effectuer les calculs tensoriels nécessaires notamment pour l'apprentissage profond (*deep learning*). Ces calculs sont optimisés et effectués soit par le processeur (CPU) soit, lorsque c'est possible, par un processeur graphique (GPU) supportant CUDA. Il est issu des équipes de recherche de Facebook, et avant cela de Ronan Collobert dans l'équipe de Samy Bengio à l'IDIAP.

PyTorch est dérivé d'un logiciel antérieur, Torch, qui s'utilisait avec le langage Lua. PyTorch est indépendant de Lua et se programme en Python.

PyTorch permet de:

- manipuler des tenseurs (tableaux multidimensionnels), de les échanger facilement avec Numpy et d'effectuer des calculs efficaces sur CPU ou GPU (par exemple, des produits de matrices ou des convolutions);
- calculer des gradients pour appliquer facilement des algorithmes d'optimisation par descente de gradient. PyTorch utilise la bibliothèque autograd[60]

3.4 Open cv



Open CV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage et la société ItSeez se sont succédé au support de cette bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel.

Cette bibliothèque est distribuée sous licence BSD[61]

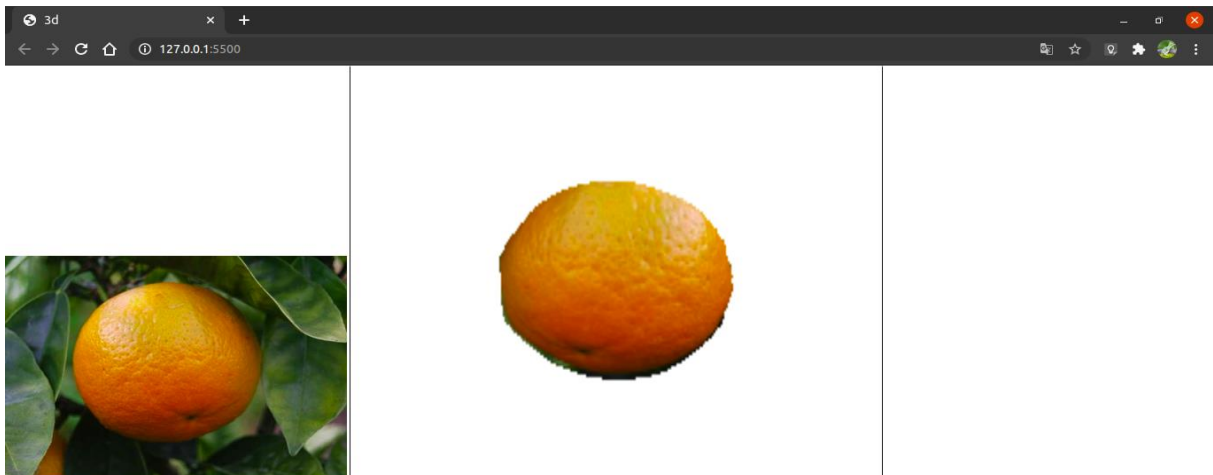
4. Résultat

Notre méthode procède en deux étapes afin de produire le résultat représenté par un objet 3D. En effet l'objet d'entrée est une image 2D cependant l'objet de sortie est un objet 3d dont l'extension est **.obj**. La figure ci-dessous montre les différentes étapes de notre méthode.



(a)

(b)



(c)

Fig4.1 Exemple d'objet orange (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture

CHAPITRE IV : Résultat

La **Fig4.2** montre un résultat d'un autre objet de la même classe avec une échelle différente et une texture différente. Cela prouve l'efficacité de notre méthode pour générer des objets 3D de différentes échelles.

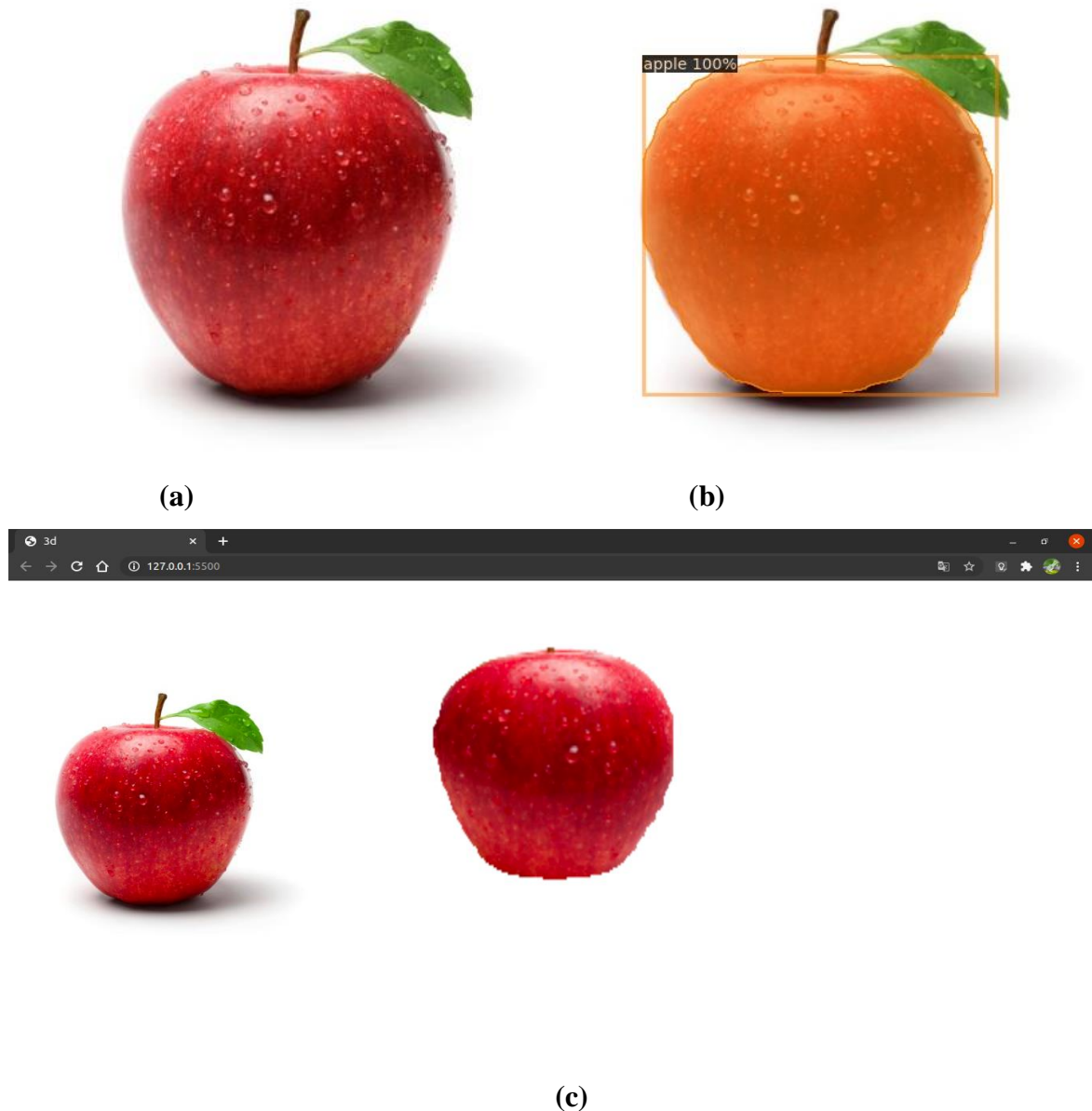
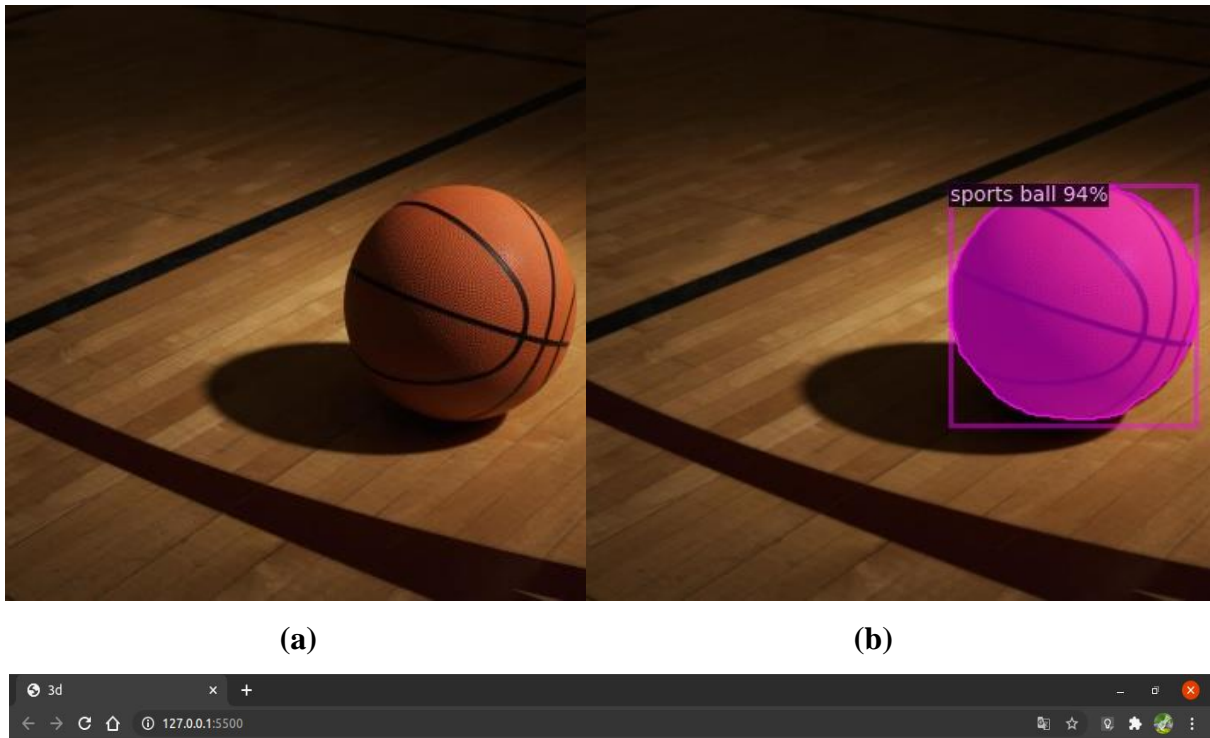


Fig4.2 Exemple d'objet pomme (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture.

CHAPITRE IV : Résultat

La **Fig4.3** montre que notre méthode capable de détecter les objets dans une image avec moins d'illumination et d'ombre dure.



(c)

Fig4.3 Exemple d'objet ballon de sport : (a) image d'entrée (b) le Mask de l'image d'entrée
(c) reconstruction géométrique et plaquage de texture.

Les figures qui suivent montre plusieurs résultats générés par notre méthode.

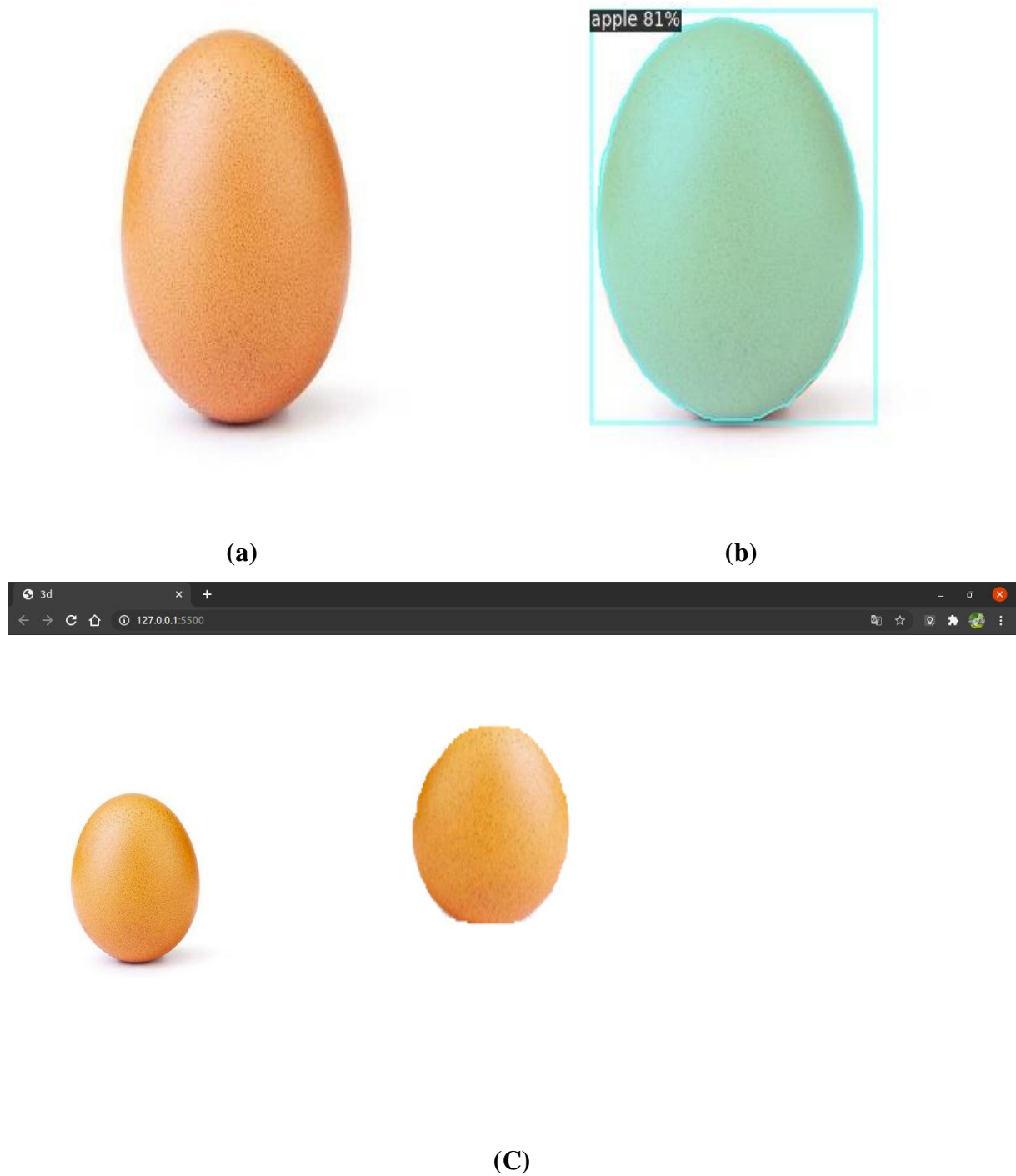
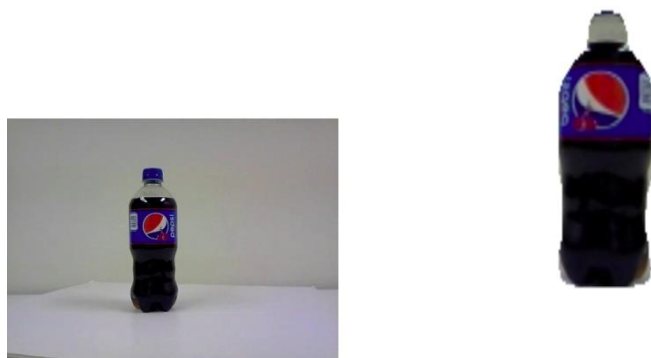
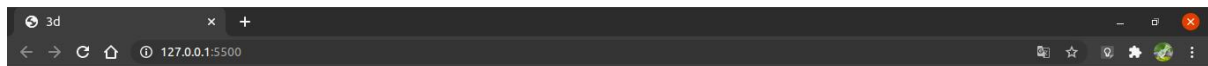


Fig4.4 Exemple d'objet œuf (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture



(a)

(b)



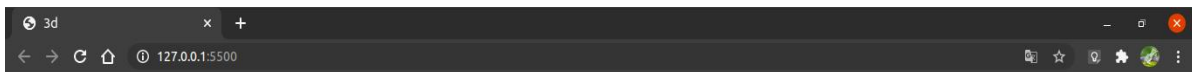
(c)

Fig4.5 Exemple d'objet bouteille (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture



(a)

(b)



(c)

Fig4.6 Exemple d'objet cup (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture



(a)

(b)



(c)

Fig4.7 Exemple d'objet Montgolfière (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture

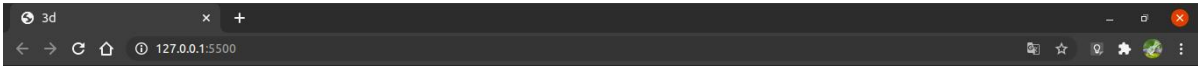
La **Fig4.8** montre la capacité de notre méthode reconstruire des objets avec textures complexes.



(a)



(b)



(c)

Fig4.8 Exemple d'objet Vase (a) image d'entrée (b) le Mask de l'image d'entrée (c) reconstruction géométrique et plaquage de texture

5. Conclusion

Durant ce chapitre nous avons présenté les résultats obtenus par notre méthode, nous constatons qu'il est possible de surmonter la complexité imposée par l'étape de modélisation des objets 3d, cela en utilisant une méthode automatique capable d'offrir un modèle approximatif.

Notre méthode traite seulement les objets dont la forme est symétrique, notre méthode produit des bons résultats en termes de détection et de reconstruction d'objets avec textures d'une qualité satisfaisante et en temps réel.

Conclusion générale et perspectives

Notre travail s'inscrit dans le cadre de reconstruction 3D des objets à partir d'une image 2D, avec l'évolution marqué par l'apprentissage profond, les méthodes à base de réseaux de neurones produisent des objets 3D avec une haute résolution, nous avons travaillé sur la reconstruction des objets qui ont une forme symétrique.

Dès le début notre principale préoccupation a été d'explorer les méthodes de reconnaissance et segmentation instantané afin d'avoir une segmentation précise de l'objet d'entrée ainsi d'accélérer la première étape de reconstruction, nous avons opté pour l'utilisation de transfer learning dans cette dernière cela permet d'avoir un mask de l'objet en temps réel, cependant pour la deuxième étape nous avons utilisé un filtre 2D afin de récupérer la profondeur de l'objet est assurer un passage d'espace 2D vers un espace 3D, notre méthode simple à mettre en œuvre et produit de bons résultats dans temps réduit ainsi elle permet de produire des objets avec et sans texture.

Pour les pistes futures nous envisageons la reconstruction de multiples objets et offrir un modèle qui prend en compte la variation de d'illumination pour avoir une reconstruction fidèle à la réalité.

Bibliographie

- [1] Gkioxari, G., Malik, J., & Johnson, J.(2019). Mesh r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (pp.9785-9795).
- [2] <http://intelligence-artificielle.weebly.com/introduction.html>
- [3]Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia.
- [4]<http://dspace.univguelma.dz:8080/xmlui/bitstream/handle/123456789/4278/PFE2019%20Lokba%20Zine%20Laabidine%20%28Tchi%20drive2%29.pdf?sequence=1&isAllowed=y>
- [5] Sabah, G. (1997). Intelligence artificielle et sciences cognitives. InCognito. Association In Cognito.
- [6]Touzet, C. (1992). Les réseaux de neurones artificiels, introduction au connexionnisme. EC2.
- [7]C.J, Gronlund. olprod, (12/07/2017) OpenLocalizationService: Présentation de Machine Learning dans le cloud Azure, "Microsoft azure,".

- [8] Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, 205-227.
- [9] Priyadharshini. (February 1, 2018). Machine Learning: What it is and Why it Matters, “simplilearn,”
- [10] Hurwitz, J. S., & Kirsch, D. (2020). *Cloud computing for dummies*. John Wiley & Sons.
- [11] Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. “O’Reilly Media, Inc.
- [12] Blum, A., & Mitchell, T. (1998, July). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92-100).
- [13] Vieux-Fort, K., Pilote, A., & Magnan, M. O. (2020). Choisir un cégep anglophone au Québec: l’expérience de jeunes francophones. *Éducation et francophonie*, 48(1), 122-143.
- [14] Alloa, E. (2018). La transparence est notre censure. *Multitudes*, (4), 56-61.
- [15] SEM, S. Intelligence artificielle 101 09 août 2018 • Catherine Dionne.
- [16] <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>
- [17] <https://www.futura-sciences.com/tech/definitions/informatique-reseau-neuronal-601>
- [18] https://thebrain.mcgill.ca/flash/capsules/pdf_articles/reseau_neurones.pdf
- [19] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press.
- [20] <https://www.aspexit.com/reseau-de-neurones-on-va-essayer-de-demystifier-un-peu-tout-ca-3/>
- [21] <https://fr.wikipedia.org/wiki/Auto-encodeur>
- [22] <https://fr.mathworks.com/discovery/deep-learning.html>
- [23] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).
- [24] Chabot, F. (2017). *Analyse fine 2D/3D de véhicules par réseaux de neurones profonds* (Doctoral dissertation).
- [25] Dore, A., Devy, M., & Herbulot, A. (2017, June). Détection d'objets en milieu naturel: application à l'arboriculture.
- [26] Redmon et al., 2015] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once : Unified, real-time object detection. *CoRR*, abs/1506.02640.

- [27] Gandhi, R. (2018). R-cnn, fast r-cnn, faster r-cnn, yolo—object detection algorithms. *Towards Data Science*, 9.
- [28] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).
- [29] Forson, E. (2017). Understanding ssd multibox-real-time object detection in deep learning. *Towards Data Science*.
- [30] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).
- [31] Mohanty, S. P. (2018). Crowdai mapping challenge 2018: Baseline with mask rcnn.
- [32]] Inglada, J. (2007). Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features. *ISPRS journal of photogrammetry and remote sensing*, 62(3), 236-248.
- [33]] Erus, G. (2008). Reconnaissance d'objets cartographiques dans les images satellitaires à haute résolution (Doctoral dissertation).
- [34] . Harris, C. G., & Stephens, M. (1998). A combined corner and edge detector 4th Alvey Vision Conf. Manchester, UK, 31 Aug.-2 Sept.
- [35] Schmid, C., & Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 19(5), 530-535.
- [36] Koenderink, J. J., & van Doorn, A. J. (1987). Representation of local geometry in the visual system. *Biological cybernetics*, 55(6), 367-375.
- [37] Déf. A.1, Dictionnaire Le Petit Robert, Société du nouveau littéré.
- [38] Shotton, J., Blake, A., & Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *IEEE transactions on pattern analysis and machine intelligence*, 30(7), 1270-1281.
- [39] Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11-15.
- [40] Image segmentation. In *ICIP03*, pages 345–348/ *Image segmentation techniques*. *CVGIP*, 29(1):100–132
- [41] Brunelli, R., & Poggiot, T. (1997). Template matching: Matched spatial filters and beyond. *Pattern Recognition*, 30(5), 751-768.
- [42] Chehata, N. (2005). Modélisation 3D de scènes urbaines à partir d'images satellitaires à très haute résolution (Doctoral dissertation).

- [43] Meyer, F., & Beucher, S. (1990). Morphological segmentation. *Journal of visual communication and image representation*, 1(1), 21-46.
- [44] Zhang, D. et Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19.
- [45]. Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques. *Pattern recognition*, 37(1), 1-19.
- [46]Kavallieratou, E., Fakotakis, N., & Kokkinakis, G. Kavallieratou, E., Fakotakis, N., & Kokkinakis, G. (2002, August). Handwritten character recognition based on structural characteristics. In *Object recognition supported by user interaction for service robots* (Vol. 3, pp. 139-142). IEEE.
- [47] Hu, M. K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2), 179-187.
- [48] Khotanzad, A., & Hong, Y. H. (1990). Invariant image recognition by Zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5), 489-497.
- [49] https://fr.qwe.wiki/wiki/3D_reconstruction
- [50] Laurent, M. (1997). Acquisition 3D au service de l'industrie. *Magazine électronique CiMax*. [50]Zou, C., Liu, J., & Liu, J. (2012, November). Precise 3d reconstruction from a single image. In *Asian Conference on Computer Vision* (pp. 271-282). Springer, Berlin, Heidelberg.
- [52] https://fr.qwe.wiki/wiki/3D_reconstruction_from_multiple_images
- [53] [Zou, C., Liu, J., & Liu, J. \(2012, November\). Precise 3d reconstruction from a single image. In Asian Conference on Computer Vision \(pp. 271-282\). Springer, Berlin, Heidelberg.](#)
- [54] [Knyaz, V. A. \(2020, April\). Machine learning for scene 3D reconstruction using a single image \(Conference Presentation\). In Optics, Photonics and Digital Technologies for Imaging Applications VI \(Vol. 11353, p. 1135321\). International Society for Optics and Photonics.](#)
- [55] Wu, Y., Kirillov, A., Massa, F., Lo, W. Y., & Girshick, R. (2019). Detectron2.
- [56]_K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [57] Van Rossum, G. (1990). The python language. See <http://www.python.org>
- [58] <https://fr.wikipedia.org/wiki/Node.js>
- [59] [Douglas Crockford](#), dans une conférence à [Yahoo!](#) [Crockford on JavaScript - Chapter 2: And Then There Was JavaScript](#) [[archive](#)], se limite à Java, Scheme et Self comme influence direct de JavaScript
- [60] <https://fr.wikipedia.org/wiki/PyTorch>

[61] <https://fr.wikipedia.org/wiki/OpenCV>