



République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la  
recherche scientifique

Université Larbi Tébessi - Tébessa

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie  
Département : Mathématiques et Informatique



Mémoire de fin d'étude  
Pour l'obtention du diplôme de **MASTER**  
Domaine : Mathématiques et Informatique  
Filière : Informatique  
Option : Systèmes d'information

Thème

**Contribution Base Intelligence Artificielle  
pour la Planification Urbaine**

Présenté Par :

Saadoud Samia

Devant le jury :

|                      |      |                          |           |
|----------------------|------|--------------------------|-----------|
| Dr. Merzoug Soltan   | MCB  | Université Larbi Tébessi | Président |
| Dr. Derdour Makhlouf | Prof | Université Larbi Tébessi | Examineur |
| Pr. Laouar Med Ridda | Prof | Université Larbi Tébessi | Encadreur |

Date de soutenance : 14/09/2020

# Résumé

La gestion des déchets accumulés est devenue une préoccupation majeure dans les zones urbaines en général et dans les municipalités algériennes en particulier, car elle peut provoquer une pollution de l'environnement par l'air, la terre et la mer et pourrait constituer une menace pour la santé humaine si elle n'est pas gérée correctement. L'une des étapes les plus importantes de la gestion des déchets est la séparation des déchets en différents composants. Ce processus se fait généralement par sélection manuelle, il est donc nécessaire de fournir un système de classification des déchets intelligent.

Dans cette mémoire, nous suggérons un modèle de classification des déchets basé sur l'apprentissage en profondeur, développé à l'aide du modèle de réseau de neurones convolutif (CNN) qui est utilisé pour classer les déchets en différents types tels que le verre, le plastique, etc. Le modèle atteint une précision de prédiction de 92% dans l'ensemble de données de test. La séparation des déchets sera plus rapide et plus intelligente et contribuera à atténuer certains problèmes de tri et de recyclage.

**Les mots clés :** Zones urbaines, Déchets accumulés, Apprentissage en profondeur, CNN, Système intelligent de classification des déchets.

# Abstract

Managing accumulated waste has become a major concern in urban areas in general and in Algerian municipalities in particular, as it can cause environmental pollution by air, land and sea and could pose a threat to human health if not managed properly. One of the most important steps in waste management is the separation of waste into different components. This process is usually done by manual selection, so it is necessary to provide an intelligent waste classification system.

In this thesis, we suggest a model of waste classification based on deep learning, developed using the convolutional neural network model (CNN) which is used to classify waste into different types such as glass, plastic, etc. The model achieves 92% prediction accuracy in the test dataset. Waste separation will be faster and smarter and will help alleviate some sorting and recycling problems.

**Keywords:** Urban areas , Accumulated waste , Deep learning , CNN , Intelligent waste classification system.

## ملخص

اصبحت ادارة النفايات المتراكمة مصدر قلق كبير في المناطق الحضرية عامة و البلديات الجزائرية خاصة ، فقد يتسبب في تلوث البيئة جوا و برا و بحرا ويمكن أن يشكل خطرا على صحة الإنسان إذا لم يتم إدارته بشكل صحيح . واحدة من أهم الخطوات في إدارة النفايات هي فصل النفايات إلى مكونات مختلفة وعادة ما تتم هذه العملية عن طريق الانتقاء اليدوي لذلك من الضروري توفير نظام ذكي لتصنيف النفايات . في هذه المذكرة نقترح نموذج تصنيف النفايات القائم على التعلم العميق ، تم تطويره باستخدام نموذج الشبكة العصبية الالتفافية الذي يس تخدم لتصنيف النفايات إلى أنواع مختلفة مثل الزجاج والبلاستيك و غيرها . يحقق النموذج دقة تنبؤ بنسبة 92 ٪ في مجموعة بيانات الاختبار. ستكون عملية فصل النفايات أسرع وأذكى و تساعد في التخفيف من بعض مشكلات الفرز و إعادة تدوير.

**الكلمات المفتاحية:** المناطق الحضرية ، النفايات المتراكمة ، التعلم العميق ، الشبكة العصبية الالتفافية ، نظام تصنيف النفايات الذكي

# Remerciements

*je remercie ALLAH de m' avoir donné la santé et le courage afin de pouvoir réussir ce travail.*

*Ce travail est l'aboutissement d'un long cheminement au cours duquel j'ai bénéficié d'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.*

*J'exprime mon grande gratitude à mon encadreur **Pr.LAOUAR MED RIDDA**, d' avoir accepté de M'encadrer durant mon travail et pour ses précieux conseils et ses orientations.*

*Je tiens aussi à remercier tous les membres de jury : **Pr.Derdour Makhluf et Dr.Merzoug Soltan**, pour leur disponibilité et acceptation d'examiner et de rapporter mon travail.*

*Je remercie ainsi Dr. Bendhib Issam le chef de spécialité Système d'information ainsi que l'ensemble du corps enseignant, je garde un très bon souvenir de vous, j'apprécie toujours vos qualités humaines, votre savoir-faire et vos compétences scientifiques.*

*Oussama, mon cher mari, je te remercie de tout mon cœur d'être mon ange gardien. Tu es toujours là, à mes côtés, pour me soutenir et pour m'encourager à avancer dans ce que je fais.*

*Merci pour ta patience et ton amour.*

# ***DEDICACE***

*A mon mari oussama*

*et mes enfants raouf, arwa,*

*ghaza et merieme djoud.*

*Saadoud Samia .*

# Table des matières

|  |          |
|--|----------|
| Introduction Générale .....  | 2        |
| <b>LE CONTEXTE DU PROJET URBAIN &amp;PROBLEMATIQUE .....</b>       | <b>4</b> |
| 1.1 Introduction .....   | 5        |
| 1.2 Principes de la planification urbaine.....                     | 5        |
| 1.3 Les objectifs de la planification urbaine.....                 | 7        |
| 1.4 Les principes fondamentaux de la planification urbaine.....    | 7        |
| 1.5 L'évaluation dans le cadre de la planification urbaine .....   | 8        |
| 1.5.1 Notion du projet urbain .....                                | 8        |
| 1.6 L'environnement durable.....                                   | 10       |
| 1.7 La gestion de déchets.....                                     | 11       |
| 1.7.1 Définition .....   | 11       |
| 1.7.2 Nomenclature des déchets.....                                | 11       |
| 1.8 Origine de la production des déchets .....                     | 12       |
| 1.9 Les Caractéristiques des déchets .....                         | 12       |
| 1.10 Classification des déchets .....                              | 12       |
| 1.11 Système de gestion des déchets.....                           | 13       |
| 1.12 Les déchets solides urbains.....                              | 14       |
| 1.13 La problématique .....  | 16       |
| 1.14 Notre objective .....   | 17       |
| 1.15 Conclusion .....  | 17       |
| <br>   |          |
| Etat de l'art .....  | 18       |
| 2.1 Introduction .....   | 19       |
| 2.2 Apprentissage automatique.....                                 | 20       |
| 2.2.1. Classification .....  | 20       |
| 2.2.1.1. Types de classification .....                             | 21       |
| 2.2.2. Fonction d'erreur et optimisation des hyperparamètres ..... | 21       |

|                          |   |           |
|--------------------------|---|-----------|
| <b>2.3</b>               | <b>Apprentissage profond .....</b>  | <b>21</b> |
| <b>2.3.1.</b>            | <b>Fonctionnement de l'apprentissage profond .....</b>                      | <b>22</b> |
| <b>2.4</b>               | <b>Domaine d'application de l'apprentissage .....</b>                       | <b>23</b> |
| <b>2.5</b>               | <b>Architectures de réseaux de neurones profonds .....</b>                  | <b>24</b> |
| <b>2.5.1.</b>            | <b>Les réseaux de neurones convolutifs (CNN) .....</b>                      | <b>24</b> |
| <b>2.5.1.1.</b>          | <b>L'opération de convolution .....</b>                                     | <b>25</b> |
| <b>2.5.1.2.</b>          | <b>Les couches de CNN .....</b>   | <b>26</b> |
| <b>2.5.1.3.</b>          | <b>Les paramètres de CNN.....</b>   | <b>28</b> |
| <b>2.5.2.</b>            | <b>R-CNN .....</b>  | <b>28</b> |
| <b>2.5.3.</b>            | <b>Faster R-CNN .....</b>   | <b>29</b> |
| <b>2.5.4.</b>            | <b>Segmentation d'instance .....</b>  | <b>30</b> |
| <b>2.5.5.</b>            | <b>Mask R-CNN .....</b>   | <b>31</b> |
| <b>2.5.6.</b>            | <b>ResNet .....</b>   | <b>31</b> |
| <b>2.5.7.</b>            | <b>Vgg Net .....</b>  | <b>32</b> |
| <b>2.5.8.</b>            | <b>Les réseaux de neurones récurrents (RNN) .....</b>                       | <b>34</b> |
| <b>2.5.8.1.</b>          | <b>Application RNN .....</b>  | <b>35</b> |
| <b>2.5.9.</b>            | <b>Apprentissage par renforcement.....</b>                                  | <b>35</b> |
| <b>2.6</b>               | <b>L'apprentissage automatique &amp; la classification des déchets.....</b> | <b>36</b> |
| <b>2.6.1.</b>            | <b>Les approches traditionnelles.....</b>                                   | <b>36</b> |
| <b>2.6.2.</b>            | <b>Les approches intelligentes .....</b>                                    | <b>37</b> |
| <b>2.7</b>               | <b>Conclusion.....</b>  | <b>39</b> |
| <br>                     |   |           |
| <b>Contribution.....</b> |   | <b>40</b> |
| <b>3.1.</b>              | <b>Introduction .....</b>   | <b>41</b> |
| <b>3.2.</b>              | <b>Problème étudiée .....</b>   | <b>41</b> |
| <b>3.3.</b>              | <b>Architecture générale de l'application .....</b>                         | <b>41</b> |
| <b>3.3.1.</b>            | <b>Prétraitement.....</b>   | <b>43</b> |
| <b>3.4.</b>              | <b>Méthodologie.....</b>  | <b>43</b> |
| <b>3.5.</b>              | <b>Architecture des nôtres réseaux proposés .....</b>                       | <b>44</b> |
| <b>3.5.1.</b>            | <b>Model 1.....</b>   | <b>44</b> |



|        |  |           |
|--------|--|-----------|
| 3.5.2. | Modèle 2.....                                  | 48        |
| 3.5.3. | Modèle 3.....                                  | 54        |
| 3.6.   | Conclusion.....                                | 57        |
|        |  |           |
|        | <b>Implémentation &amp; Discussion.....</b>    | <b>58</b> |
| 4.1.   | Introduction .....                             | 59        |
| 4.2.   | Les outils et bibliothèques Utilisés .....     | 59        |
| 4.2.1. | Environnement et technologies logicielles..... | 59        |
| 4.2.2. | Langage utilisé.....                           | 61        |
| 4.2.3. | Les bibliographies utilisées.....              | 61        |
| 4.3.   | Datasets .....                                 | 63        |
| 4.3.1. | Premier dataset.....                           | 64        |
| 4.3.2. | Deuxième Dataset .....                         | 65        |
| 4.3.3. | Etiqueter les image .....                      | 67        |
| 4.4.   | Processus des modèles .....                    | 68        |
| 4.4.1. | Étapes impliquées .....                        | 68        |
| 4.5.   | Expérimentation des résultats.....             | 70        |
| 4.6.   | Synthèse .....                                 | 77        |
| 4.7.   | Implémentation d'interface graphique.....      | 78        |
| 4.8.   | Conclusion.....                                | 82        |
|        |  |           |
|        | <b>Conclusion Générale.....</b>                | <b>84</b> |

## Liste des figures

|   |    |
|---|----|
| FIGURE 1.1. Quelques composantes du système urbain.   | 06 |
| FIGURE 1.2. Principes de la planification urbaine   | 07 |
| FIGURE 1.3. Processus de conduite d'un projet urbain .  | 09 |
| FIGURE 1.4. Types de pollution générée par les déchets solides [13]                                 | 15 |
| FIGURE 2.1. Affiliation de Deep Learning (relation entre IA et ML et DL)                            | 19 |
| FIGURE 2.2. Exemple de classification avec machine Learning   | 20 |
| FIGURE 2.3. Exemple de fonctionnement de Deep Learning  | 23 |
| FIGURE 2.4. Domaines d'applications de l'apprentissage  | 23 |
| FIGURE 2.5. Architecture du CNN   | 25 |
| FIGURE 2.6. Schéma du parcours de la fenêtre de filtre sur l'image.                                 | 25 |
| FIGURE 2.7. Convolution avec une argument maximal.  | 26 |
| FIGURE 2.8. Pooling avec un filtre 2x2 et un pas de 2.  | 27 |
| FIGURE 2.9. Architecture de R-CNN   | 28 |
| FIGURE 2.10. Architecture Faster R-CNN [19]   | 29 |
| FIGURE 2.11. Faster R-CNN : Fenêtre coulissante et résultat [19]                                    | 30 |
| FIGURE 2.12. Exemple de segmentation d'instance [20]  | 30 |
| FIGURE 2.13. Architecture Mask R-CNN [21]   | 31 |
| FIGURE 2.14. Une visualisation de l'architecture VGG [25]   | 32 |
| FIGURE 2.15. Réseaux convolutifs très profonds pour la reconnaissance d'image à grande échelle [25] | 33 |

|  |    |
|--|----|
| FIGURE 2.16. Architecture de RNN   | 34 |
| FIGURE 2.17. Exemple d'un agent utilise l'apprentissage par renforcement | 35 |
| FIGURE 2.18. Les 6 classes de data set TrashNet                          | 38 |
| FIGURE 3.1. Architecture générale de l'application                       | 42 |
| FIGURE 3.2. Diagramme de la méthodologie                                 | 44 |
| FIGURE 3.3. Architecture de CNN  | 45 |
| FIGURE 3.4. Architecture première modèle CNN                             | 48 |
| FIGURE 3.5. Division des données   | 49 |
| FIGURE 3.6. Architecture du deuxième modèle resnet                       | 52 |
| FIGURE 3.7 La valeur de Learning rate                                    | 53 |
| FIGURE 3.8. Architecture du troisième modèle Mask-RCNN                   | 56 |
| FIGURE 4.1.Echantillon de donnes de TrashNet [33]                        | 65 |
| FIGURE 4.2.Echantillon du deuxième data set [37]                         | 66 |
| FIGURE 4.3.Choix de fichier dans labelImg                                | 67 |
| FIGURE 4.4.Création de box dans labelImg                                 | 68 |
| FIGURE4.5.Le processus proposé   | 70 |
| FIGURE4.6.Précision et Erreur pour le modèle 01                          | 71 |
| FIGURE4.7.Exemple de test pour le modèle 01                              | 72 |
| FIGURE4.8.Exemple de test pour le modèle 03                              | 77 |
| FIGURE4.9.Interface principale de l'application                          | 79 |
| FIGURE4.10.Sélection d'image   | 80 |
| FIGURE 4.11. Le fichier des images                                       | 81 |
| FIGURE 4.12. Affichage de choix  | 81 |
| FIGURE 4.13. Résultat final  | 82 |



Introduction

Générale

## 1. Introduction

Le recyclage est nécessaire en milieu urbain. Le processus actuel de tri et de recyclage des déchets nécessite des installations de recyclage pour le tri manuel et une série de grands filtres pour séparer les objets plus spécifiques.

Les consommateurs peuvent également être confus quant à la façon de déterminer la bonne façon d'éliminer une grande variété de matériaux utilisés dans les emballages

L'élimination des déchets solides est devenue un très gros problème, et les déchets solides comprennent le papier, le bois, le plastique, le métal, le verre, etc. Les méthodes traditionnelles de gestion des déchets sont la mise en décharge et l'incinération, et elles ont des résultats inefficaces, coûteux et polluants.

## 2. Objective

Notre objectif est de réduire le travail manuel et de trouver un moyen intelligent de trier les ordures. Cela rendrait les centres de tri des déchets plus efficaces et aiderait à réduire les déchets, car les employés ne trient pas toujours tout avec une précision correcte. Afin de simuler le flux de matériaux dans une usine de recyclage ou si quelqu'un a pris une photo d'un matériau pour l'identifier, notre suggestion consiste à recevoir des photos d'une chose et à la classer dans le type de matériau de recyclage. L'entrée de la première contribution est constituée d'images qui ont une forme sur un fond blanc propre. Ensuite, nous utilisons l'algorithme CNN pour classer l'image en six catégories de déchets. Pour la deuxième contribution est pour le but de classer des multiple objets dans une même image . Ainsi, nous pouvons prédire la catégorie de déchets à laquelle appartient l'objet. Cela aura des impacts environnementaux positifs ainsi que des impacts économiques bénéfiques.

### **Le travail réalisé dans cette mémoire consiste à :**

- ✓ Faire un état de l'art sur les systèmes de gestion des déchets solides basés sur la classification d'images des déchets.
- ✓ Proposé un système automatique basé sur l'intelligence artificielle pour la classification des

déchets.

- ✓ Évaluer les performances du système proposé.

### **3. La structure du memoire**

#### **1. Le premier chapitre : Contexte de travaille & Problématique**

- Présente le contexte de projet urbain, de gestion des déchets et la problématique de travaille.

#### **2. Le second chapitre : Etat de l'art**

- Un état de l'art sur l'apprentissage automatiques ainsi que les travaux d'apprentissage automatique à la gestion des dèches

#### **3. Le troisième chapitre : Contribution**

- Introduit notre contribution

#### **4. Le dernier chapitre : Implémentation & Evaluation**

- C'est une implémentation de notre proposition, les résultats et la discussion





# Chapitre 1

***Le Contexte Du Projet Urbain &  
Problématique***

## 1.1 Introduction

Les milieux urbains constituent une dimension fondamentale dans le monde et la conception de ces milieux devient une tâche très difficile à cause des besoins très divers auxquels l'aménagement des surfaces doit répondre tels que : l'agriculture, la protection de l'environnement, la vie sociale, les activités économiques et financières, etc. En effet, cet aménagement est assuré par la discipline nommée : la planification urbaine. La planification urbaine est l'ensemble des outils et des moyens permettant la mise en œuvre des politiques sectorielles et la rationalisation de la gestion urbaine à travers la définition d'objectifs, de principes de développement et de projets d'aménagement.

Dans ce chapitre, nous allons présenter un aperçu sur le contexte de la planification urbaine en mettant l'accent sur ses différents aspects et objectifs afin de présenter notre problématique.

## 1.2 Principes de la planification urbaine

La planification urbaine a été définie et abordée selon différentes sources, parmi lesquelles nous citons :

- Le dictionnaire d'urbanisme [1] définit la planification urbaine :

*« Ensemble d'études, de démarches, voire de procédures juridiques ou financières, qui permettent aux collectivités publiques de connaître l'évolution des milieux urbains, de définir des hypothèses d'aménagement concernant à la fois l'ampleur, la nature et la localisation des développements urbains et des espaces à protéger, puis d'intervenir dans la mise en œuvre des options retenues ».*

- D'après [2] la planification urbaine est :

*« Les plans, les institutions, les pratiques et les techniques qui cherchent à organiser la ville. La planification part de la demande sociale et non de la quête d'une quelconque ville idéale. Elle est apparue comme moyen d'empêcher l'anarchie urbanistique, de réaliser l'harmonie entre les besoins en logements, en emplois, en services, en circuits de distribution et en infrastructures de circulation. Elle traduit le passage de l'urbanisme d'autrefois à l'aménagement de l'espace de la ville de demain ».*

- Finalement, selon Nigel Taylor [3]

*« La planification urbaine n'est pas, au sens strict, une science (pas même une science sociale). C'est plutôt une forme d'action sociale, mue par certaines valeurs morales, politiques et esthétiques dans le but de donner forme à l'environnement physique. C'est pour cette raison que ce type d'action sociale peut se retrouver sous de multiples formes ».*

La planification urbaine couvre un ensemble de secteurs physiques et sociologiques, nous citons la démographie, l'emploi, l'habitat, l'utilisation des sols, les services publics, le budget, l'environnement, le transport... etc. La figure 1.1 illustre l'interaction entre ces différents secteurs.

Robert Laurini [4] a mentionné que ces activités de planification ont les quatre qualités suivantes en commun :

**1. La planification est orientée vers l'avenir :**

Les décisions prises dans le processus de planification sont généralement faites pour affecter une condition future dans l'environnement.

**2. La planification concerne l'évaluation des solutions pour résoudre les problèmes :**

Cela est profondément ancré dans les théories de la planification rationnelle qui sous-tendent la pratique de la planification actuelle.

**3. La planification est politique :**

Chaque décision de planification publique a lieu dans un contexte politique. Il est important de réaliser que la majorité des activités de planification impliquent l'utilisation ou la réglementation des terrains sous une certaine forme.

**4. La planification a une responsabilité particulière pour représenter les besoins des minorités :**

Les personnes handicapées, pauvres et les groupes sous-représentés. Les planificateurs doivent prêter une attention particulière aux besoins de ces groupes dans le cadre de leur code professionnel de déontologie.

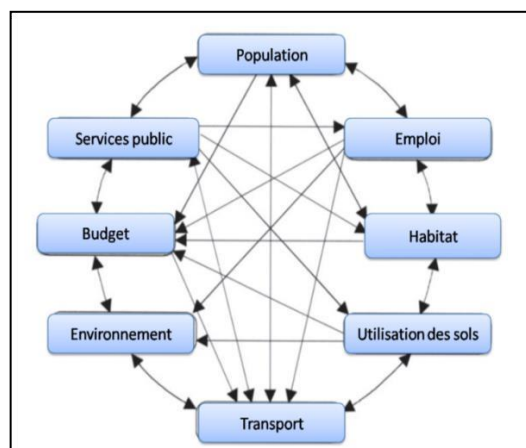


FIGURE 1.1. Quelques composantes du système urbain[4].

### 1.3 Les objectifs de la planification urbaine

La réalisation des objectifs de la planification urbaine est essentiellement une tâche politique. Les objectifs de la planification territoriale urbaine sont résumés comme suit [5] :

- La planification urbaine doit coordonner les différentes fonctions du sol.
- La planification urbaine doit coordonner les affectations, arbitrer les conflits d'utilisation, etc.
- La planification territoriale urbaine ne permet pas uniquement de légaliser les zones à bâtir, mais elle propose aussi une approche globale d'urbanisation, de transport, d'environnement, etc.

### 1.4 Les principes fondamentaux de la planification urbaine

La planification urbaine s'appuie sur trois grands principes (voir figure 1.2) :

#### 1. La coordination

Une des principales missions de la planification territoriale urbaine, elle consiste à coordonner les différentes demandes formulées en matière d'affectation des sols compte tenu de ses propres objectifs et des exigences des politiques sectorielles.

#### 2. La coopération

La coopération une action collective par laquelle des sujets contribuent à un même résultat. En ce sens, les processus optant pour un mode de fonctionnement coopératif attendent en retour une minimisation des risques et une réduction de l'incertitude.

#### 3. La participation

La participation est un moyen indispensable pour la validation d'un projet en planification urbaine. Elle permet l'intégration des différents acteurs (individus ou groupes) pour aider le décideur à établir un choix et améliorer les procédures décisionnelles.

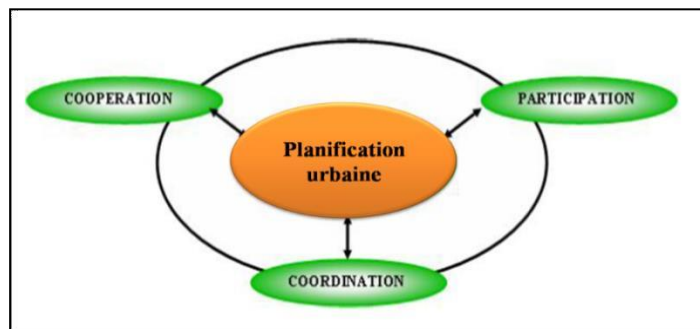


FIGURE 1.2. Principes de la planification urbaine [6].

## 1.5 L'évaluation dans le cadre de la planification urbaine

### 1.5.1 Notion du projet urbain

La notion de « projet urbain » apparaît dans les années 1970 comme une réponse à de nombreux bouleversements [7]. Selon [8], le projet urbain « organise un territoire afin d'en améliorer l'usage, la qualité, le fonctionnement, la dynamique économique et culturelle et les relations sociales. Il doit assurer à tous l'accessibilité à l'espace public, à l'habitat, aux équipements, aux transports ; se préoccuper de la qualité des espaces publics, de l'architecture, des paysages, de l'environnement naturel, de la mise en valeur du patrimoine ; servir les enjeux du développement durable, avec une utilisation économe de l'espace tout en garantissant le fonctionnement des infrastructures, ainsi que des réseaux de transports et de distribution.»

#### 1.5.1.1 Types de projets urbains

Il existe trois types de projets urbains :

- **Le projet urbain politique** : regroupe les projets dits « de territoire », « de ville », « de développement » ; il couvre tout le territoire communal ou intercommunal.
- **Le projet urbain opérationnel** : est considéré comme des opérations urbaines d'une certaine ampleur, durant au moins une dizaine d'années, généralement multifonctionnelles, associant des acteurs privés.
- **Le projet urbain architectural** : est réservé à l'approche du « design urbain », en lien avec sa réhabilitation faite par les architectes-urbanistes. Il est centré sur un bâtiment, ou un ensemble de bâtiments.

#### 1.5.1.2 Acteurs du projet urbain

Il existe plusieurs groupes qui ont un rôle important pour l'avancement du projet urbain :

- 1) **Les acteurs concernés** : qui ont un rôle dans la marche du projet urbain. On distingue :
  - **Les décideurs** : il s'agit des élus concernés, ainsi que des chefs de service des administrations ayant un pouvoir décisionnel sur le projet urbain. Leur rôle est de donner une orientation au projet urbain, et de mettre à disposition les ressources nécessaires.
  - **Les opérationnels** : ce sont les acteurs en charge de la gestion concrète du projet : le chef de projet, les collaborateurs des administrations impliquées dans la structure opérationnelle. Leur rôle est de mener à bien le projet urbain, en réalisant les objectifs fixés par les décideurs.

2) **Les acteurs intéressés** : qui s'impliquent plus ou moins ponctuellement sur un aspect ou un autre du projet urbain. On distingue :

- **Les mandataires** : sont des professionnels qui ont une mission sur un aspect ou l'autre du projet urbain : consultants stratégiques, architectes, sociologues et animateurs des démarches participatives.
- **Les associations** : ce sont des interlocuteurs clés pour la gestion des projets. Non seulement elles amènent de précieuses connaissances sur le contexte local, mais elles proposent le plus souvent un regard pointu et complémentaire sur des thématiques particulières.

3) **Les acteurs touchés** : sont ceux que le projet urbain affecte directement. On distingue :

- **Les propriétaires** : il peut s'agir de simples propriétaires privés souhaitant valoriser leur parcelle, ou de propriétaires institutionnels tels que les caisses de pension et les assurances.
- **Les habitants** : les habitants ne constituent pas un groupe d'acteurs homogène. Ils se distinguent par leur attitude, par leur niveau de participation, par les enjeux qu'ils défendent (privés, collectifs, sociaux, environnementaux,...).

### 1.5.1.3 Principales étapes du projet urbain

Le déroulement d'un projet urbain s'articule autour d'un processus d'étapes séquentielles et complémentaires [9] (Figure 1.3) :

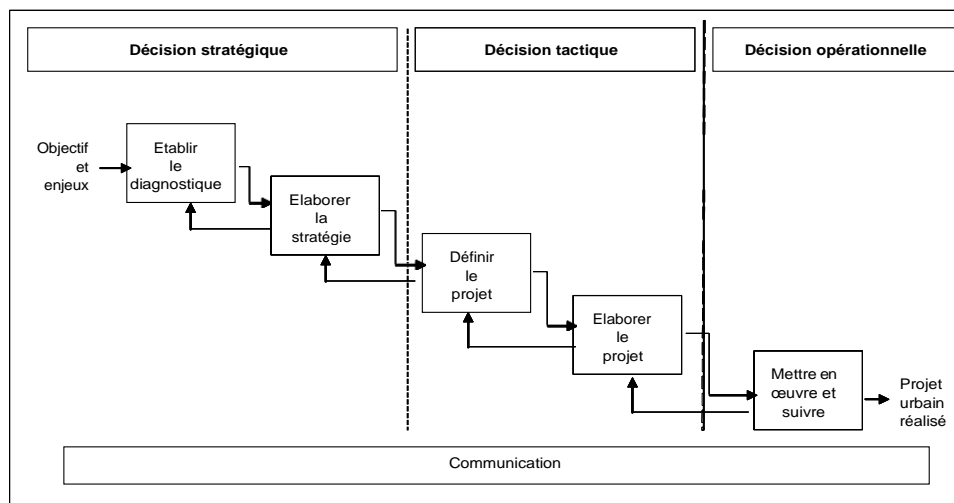


FIGURE 1.3. Processus de conduite d'un projet urbain [9].

- **Etablir le diagnostic** : pour préciser les réflexions menées en matière d'urbanisme, la consultation des différents acteurs et les études et les données existantes. Il est nécessaire pour définir le contexte, la problématique et le périmètre d'intervention.

- **Elaborer la stratégie** : c'est l'étape la plus difficile à franchir et la plus sensible politiquement. En effet elle oblige les responsables politiques à s'avancer, à se positionner vis-à-vis d'un avenir par définition incertain.
- **Etablir le projet** : c'est l'étape la plus classique, elle consiste à mettre en forme (textes, documents graphiques, ...) la stratégie définie à l'étape précédente en termes de grandes politiques d'aménagement et de développement économique, social, environnemental, ...
- **Elaborer le projet** : à la suite du diagnostic, les objectifs et les grandes lignes du projet urbain se formalisent dans le cadre d'un plan de référence. Ce document est constitué de plans synthétiques qui traduisent les grandes options de développement du projet urbain.
- **Mettre en œuvre et suivre** : dès la conception du cahier des charges. Celle-ci va de la collecte des données jusqu'à la définition précise et concrète des différents secteurs étudiés et leur programmation dans le temps.
- **Communication du projet** : cette étape est essentielle, mais souvent négligée. Elle sert à communiquer le projet aux entreprises, institutions, associations et habitants ; autrement dit à tous les acteurs du projet urbain.

## 1.6 L'environnement durable

A l'heure actuelle, la qualité environnementale de la plupart des villes reste à un bas niveau, la situation est très sévère et inquiétante due à une pollution de l'air, de l'eau, du bruit et des ordures. Dans un certain sens, la capacité environnementale des villes atteint ses limites, au-delà desquelles elle ne peut plus soutenir le développement économique. Le problème environnemental est devenu l'un des goulots d'étranglement du développement économique et social des villes :

- 1) **La pollution de l'eau en milieu urbain** : 50% des eaux souterraines en milieu urbain sont polluées, et 86% des eaux de cours en ville ne sont pas de la qualité tolérable.
- 2) **La pollution atmosphérique** : la pluie acide tombe sur la plupart des régions urbaine ; presque toutes les villes algériennes sont frappées à la fois par les entreprises polluées et l'utilisation des ressources polluées ; la présence des particules fines est plus que tolérable.
- 3) **Les déchets sont catastrophiques** : deux tiers de villes sont entourées par les ordures de la vie quotidienne, entassées en banlieue sans avoir été traitées et qui sont devenues une deuxième source de pollution.

- 4) **Les nuisances acoustiques** : les résultats de la supervision sur la plupart des villes montrent que plus de deux tiers de la population citadine vivent dans un environnement plus ou moins bruyant.
- 5) **Le bruit** : La circulation trop intense en milieu urbain à cause de grande nombre de voiture et la circulation intense.

La principale cause de ces problèmes est l'homme qui ne cesse de s'accroître chaque année et qui laisse de plus en plus son empreinte sur l'environnement pour satisfaire ses besoins et pour améliorer sa condition de vie.

Les stratégies pour développer l'environnement urbain :

- Inciter toute la société à l'aménagement de l'environnement
- Optimiser l'utilisation circulaire des ressources
- Eviter un cycle vicieux de la pauvreté et des problèmes environnementaux

Tous les points ci-dessus sont indirectement contenus dans le processus de gestion des déchets. Ce processus est introduit dans la partie suivante.

## 1.7 La gestion de déchets

### 1.7.1 Définition

Dans le langage courant, le terme déchets désigne ordure, immondice, ou tout autre résidu rejeté parce qu'il n'est plus consommable ou utilisable et donc n'a plus de valeur.

La Convention de Bâle sur le contrôle des mouvements de déchets dangereux définit les déchets comme les « *substances ou objets qu'on élimine, qu'on a l'intention d'éliminer ou qu'on est tenu d'éliminer en vertu des dispositions du droit national* » [10].

### 1.7.2 Nomenclature des déchets

Tous les déchets sont identifiés par un code à six chiffres comprenant :

- Sa catégorie d'origine (1<sup>er</sup> et 2<sup>ème</sup> chiffres),
- Son regroupement intermédiaire (et 3<sup>ème</sup> et 4<sup>ème</sup> chiffres),
- Sa désignation (et 5<sup>ème</sup> et 6<sup>ème</sup> chiffres).

Un astérisque (\*) est ajouté pour distinguer les déchets dangereux.



## 1.8 Origine de la production des déchets

La production des déchets est inéluctable pour les raisons suivantes :

- **Biologiques** : tout cycle de vie produit des métabolites ;
- **Économiques** : les produits en une durée de vie limitée ;
- **Technologiques** : tout procédé industriel conduit à la production de déchet ;
- **Écologiques** : les activités de la dépollution (eau, air) génèrent inévitablement d'autres Déchets qui nécessiteront une gestion spécifique ;
- **Accidentelles** : l'inévitable dysfonctionnement des systèmes de production et de consommation sont eux aussi à l'origine de déchets.
- **Chimiques** : toute réaction chimique est régie par le principe de la conservation de la matière et dès que veut obtenir un produit à partir de deux autres on en produira un quatrième

## 1.9 Les Caractéristiques des déchets

On caractérise les déchets par quatre paramètres essentiels suivants :

- **La densité** : La connaissance de la densité est d'une grande importance pour le choix des moyennes décollectes et de stockage.
- **Le degré d'humidité** : Les ordures renferment une suffisante quantité d'eau variant en fonction des saisons et le milieu environnemental
- **Le pouvoir calorifique** : Le pouvoir calorifique est défini comme la quantité de chaleur dégagée par la combustion de l'unité de poids en ordures brutes.
- **Le rapport des teneurs en carbone et azote** : Le rapport C/N a été choisi comme critère de qualité des produits obtenus par le compostage des déchets.

## 1.10 Classification des déchets

Les déchets peuvent être solides, liquides ou gazeux. Ils peuvent être dangereux ou non, ils peuvent être aussi classés selon leur provenance ou leur composition et leur propriété physique et chimique. On peut classer les déchets selon plusieurs critères : En fonction de l'activité à l'origine du déchet, en fonction de sa nature ainsi que de son mode d'élimination :

- **Déchets organiques.** Ce sont les déchets issus de déchets organiques. Ils sont générés principalement dans des résidences, des restaurants et des établissements commerciaux travaillant avec des produits alimentaires.

- **Déchets recyclables.** Ce sont tous les déchets qui peuvent être utilisés dans le processus de transformation en d'autres éléments ou dans la fabrication de matières premières. Elle est générée dans les résidences, les entreprises doit être séparée de manière à ce que les équipes de collecte sélective se rassemblent puis se chargent de la transformation finale.
- **Déchets industriels.** Ce sont les résidus, principalement solides, provenant du processus de production dans les industries. Il est généralement composé de restes de matières premières destinées au recyclage ou à la réutilisation dans le processus industriel.
- **Déchets hospitaliers.** Ce sont les déchets produits dans les hôpitaux et les cliniques médicales qui peuvent présenter une contamination et transmettre des maladies aux personnes qui entrent en contact avec eux. Il doit être traité conformément aux normes établies, avec tous les soins possibles.
- **Déchets commerciaux.** C'est celui produit par les établissements commerciaux, tels que les magasins de vêtements, les jouets et les appareils ménagers. Ces déchets sont presque entièrement destinés au recyclage.
- **Déchets verts.** C'est le matériau qui résulte principalement de l'élagage des arbres, des branches, des troncs, des écorces et des feuilles qui tombent dans les rues. Parce qu'il s'agit de matière organique, il pourrait être utilisé pour le compostage et la production d'engrais organique.
- **Déchets électroniques.** Il s'agit des déchets générés par la mise au rebut de produits électroniques grand public qui ne fonctionnent plus ou sont devenus obsolètes. Pour l'élimination, il existe des lieux appropriés, tels que des sociétés et des coopératives actives dans le domaine du recyclage.
- **Déchets nucléaires.** C'est celui qui est généré principalement par les centrales nucléaires. Il s'agit d'un déchet extrêmement dangereux, car il s'agit d'un élément radioactif et doit être traité conformément à des normes de sécurité strictes.

## 1.11 Système de gestion des déchets :

La gestion des déchets est l'ensemble des opérations et moyens mis en œuvre pour limiter, valoriser ou éliminer les déchets. Les différentes opérations de gestion sont :

- 1) **La prévention :** La prévention des déchets consiste à prévenir ou réduire la production et la nocivité des déchets. La prévention inclut toutes les étapes du cycle de vie d'un produit avant qu'il soit considéré comme un déchet. La prévention quantitative concerne la diminution de la masse et du volume des déchets, la prévention qualitative concerne la nocivité des déchets.

- 2) **Le réemploi**, il englobe la réutilisation et la régénération et le recyclage. C'est, en quelque sorte, prolonger la durée de vie du produit avant qu'il ne devienne un déchet.
- 3) **La collecte** : La collecte désigne l'ensemble des opérations qui consistent à regrouper les déchets, depuis leurs sources de production puis à les transporter jusqu'aux centres de traitement [10].
- 4) **Le transit et le transport des déchets** : La station de transit ou centre de transfert reçoit les flux de déchets ménagers et permet leur stockage dans une fosse, des conteneurs ou sur une aire. Les déchets, éventuellement compactés sont ensuite acheminés, par un mode de transport vers une installation de traitement ou de stockage.
- 5) **Le tri** : Opération de séparation des différents flux de déchets par les producteurs. Le tri sélectif consiste à récupérer séparément les déchets selon leur nature. Il est possible de faire le tri à la source de création de déchet ou de collecter les déchets ensemble et d'effectuer le tri ensuite dans un centre de tri [11].
- 6) **Le traitement** : Ensemble de procédés visant à transformer les déchets pour notamment en réduire dans des conditions contrôlées leur potentiel polluant initial, la quantité ou le volume, et assurer leur recyclage ou leur valorisation.

## 1.12 Les déchets solides urbains

### 1.12.1 Définition

Le dictionnaire de l'Environnement définit un déchet solide comme un déchet qui n'est pas à l'état liquide. (Rushbrook et Pugh 1999) ont précisé que le terme déchet solide peut se référer au déchet municipal qui contient sept catégories : résidentiel , commercial, institutionnel, déchets de nettoyage des voies publiques, déchets de construction et de démolition, déchets hospitaliers, déchets industriels.

### 1.12.2 Type de déchets solides

Selon leur origine, les déchets solides peuvent être de types différents, tels [12]

- Déchets résidentiels.
- Industrielle
- Institutionnel
- Construction et démolition
- Services municipaux

### 1.12.3 Caractéristiques des déchets solides

- **Corrosif** : contenant des acides capables de corroder des conteneurs pour l'esprit
- **Allumable** : peuvent provoquer des incendies dans certaines conditions(ex huiles usagées)
- **Réactifs** : ils provoquent des explosions, des vapeurs toxiques lorsqu'ils sont chauffés.
- **Toxicité** : déchets nocifs ou mortels ingérés ou absorbés [12]

### 1.12.4 Impacts des déchets solides

#### 1) Impacts sur la santé humaine

- Hépatites épidémiques et sériques.
- Conjonctivites épidémiques.
- Proéminence de la tuberculose.
- Effets multiples des substances radioactives.
- Intoxications aux produits dangereux.
- Maladies de contact de la peau et des muqueuses.

#### 2) Impacts sur l'environnement

Les déchets solides ont un impact environnemental sévère qui se manifeste par la (Figure 1.4) :

- Altération de la qualité de l'air (gaz, fumées et poussières) ;
- Altération des sols et des paysages par des polluants chimiques ;
- Pollution des ressources en eau par les infiltrats et les eaux usées.

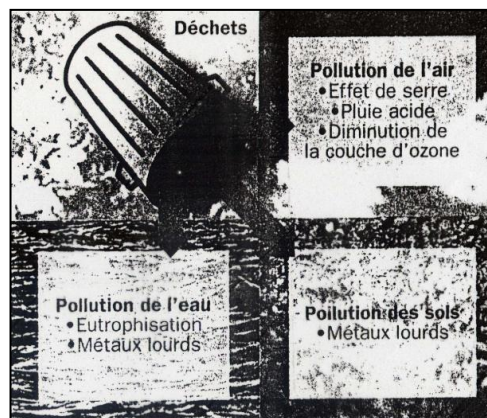


FIGURE 1.4. Types de pollution générée par les déchets solides [13]

### **1.12.5 Les Obstacles empêchant le bon tri des déchets solides**

- Manque d'incitations ;
- Absence de technologie ;
- Informations inadéquates ;
- Manque de sensibilisation ;
- Dominance du secteur non organisé ;
- Problèmes de financement et de durabilité ;
- Comportement irresponsable du consommateur ;
- Manque de coordination entre les différentes autorités.
- Capacité limitée des agences gouvernementales ;

## **1.13 La problématique**

Des millions de personnes dans le monde trient manuellement les ordures, parfois triées selon les normes de sécurité au travail, parfois ces normes sont inexistantes et dangereuses pour l'homme.

Les déchets sont constitués de nombreux composants, dont certains contiennent des substances toxiques qui ont un impact négatif sur la santé humaine et l'environnement, s'ils ne sont pas manipulés correctement. Ces risques sont souvent causés par un dépistage et un traitement incorrect. Les éléments dangereux présents dans les déchets solides comprennent les métaux ferreux et non ferreux, les plastiques, le verre, les circuits imprimés, le caoutchouc, les retardateurs de flamme et d'autres éléments. La présence de ces éléments encourage le secteur informel à poursuivre des méthodes d'extraction non scientifiques telles que la combustion et les bains acides. Les travailleurs sans instruction et non qualifiés impliqués dans la séparation et l'élimination des déchets mettent leur santé en danger en raison des mauvaises pratiques suivies au cours de ce processus.

La gestion des déchets fait référence au processus d'élimination des déchets tout en respectant l'environnement. La première étape consiste à collecter les déchets auprès des consommateurs, puis à les trier.

Pour assurer le bon fonctionnement du processus de trie, nous suggérons que l'introduction de l'intelligence artificielle soit nécessaire pour faciliter les efforts de tri des déchets afin de réduire les déchets et la pollution. Cela aura non seulement des impacts environnementaux positifs, mais aussi des impacts économiques bénéfiques. De plus, notre système jouit d'un grand attrait social en ajoutant la valeur du savoir et du catalyseur social dans l'élimination et l'élimination des déchets.

## 1.14 Notre objective

Cette étude vise à comprendre comment les déchets sont triés pour "progresser vers une économie circulaire où les déchets sont valorisés comme ressources et réutilisés". Identifier les opportunités pour les entreprises de trier leurs déchets de manière plus intelligente. Le tri intelligent contribuera à augmenter la valeur des déchets en tant que ressource et à réduire les dommages à l'environnement.

Nous proposons une solution de tri intelligent des déchets basée sur une technologie d'apprentissage automatique qui a déjà prouvé son efficacité dans le classement d'images. L'application de cette technologie au secteur des déchets permettra un tri fiable et sûr des déchets. Les utilisateurs pourront trier les images de déchets via une interface utilisateur avec différents niveaux de fonctionnalité et d'accès.

## 1.15 Conclusion

Le domaine de la planification urbaine est un domaine très riche et vaste. Il inclut de nombreuses et différentes problématiques telles que La protection de l'environnement. L'informatisation de ces problèmes apparait une tâche très intéressante à cause de nombreux obstacles qui empêchent l'élimination appropriée des déchets. Dans le cadre d'une vision intégrée du développement durable, il convient d'appliquer les technologies les plus récentes afin de réduire ces obstacles et d'appliquer le principe de l'économie économique, l'application des principes de responsabilité et la maîtrise de la bonne gestion des déchets de manière crédible.

De ce fait, le développement d'une application en planification urbaine pour résoudre la problématique d'évaluation des projets urbains nécessitera l'intégration de différentes disciplines, à savoir l'automatisation et l'intelligence artificielle.

Le chapitre suivant introduit la technologie d'apprentissage automatique et un état de l'art sur les différentes solutions de gestion de déchets



# Chapitre 2

*Etat de l'Art*



## 2.1 Introduction

Beaucoup d'entre nous sont incapables de faire la distinction entre intelligence artificielle (IA), apprentissage automatique (ML) et apprentissage profond (DL), mais si les trois termes sont souvent utilisés de manière interchangeable, ils ne font pas référence aux mêmes choses. Voici une image qui tente de visualiser la distinction ou bien la relation entre eux :

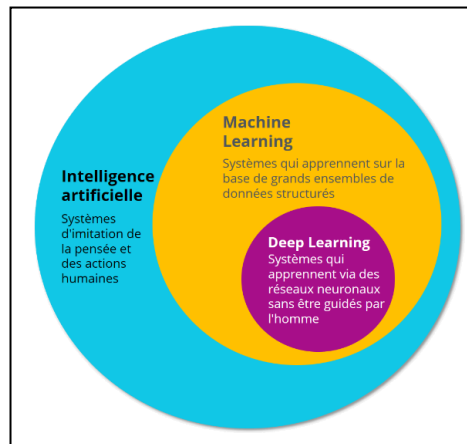


FIGURE 2.1. Affiliation de Deep Learning (relation entre IA et ML et Deep Learning)[14]

L'apprentissage automatique se concentre uniquement sur la résolution des problèmes concrets. Il faut également quelques idées de l'intelligence artificielle. L'apprentissage automatique passe par les réseaux de neurones conçus pour imiter les capacités de prise de décision de l'homme. Les outils et techniques de ML sont les deux sous-ensembles étroits centrés uniquement sur l'apprentissage en profondeur. Nous devons l'appliquer pour résoudre tout problème qui nécessite une réflexion, qu'il soit humain ou artificiel. Tout réseau de neurones profonds comprendra trois types de couches : la couche d'entrée, la couche cachée et la couche de sorties

Nous utilisons un algorithme machine pour analyser les données, en tirer des leçons et prendre des décisions éclairées en fonction de ce qu'elles ont appris. Fondamentalement, l'apprentissage en profondeur est utilisé en couches pour créer un « réseau de neurones artificiel » capable d'apprendre et de prendre des décisions intelligentes par lui-même. Nous pouvons dire que l'apprentissage en profondeur est un sous-domaine de l'apprentissage automatique.

Différentes architectures de réseau de neurone sont utilisées dans DL, chacune avec ses propres avantages et inconvénients. Les réseaux de neurones convolutionnels ou CNN sont une extension des MLP permettant de répondre efficacement aux principaux défauts des MLP. Ils sont conçus pour extraire automatiquement les caractéristiques des images et aussi des texte (1D) d'entrée, sont invariants à de légères distorsions de l'image, et implémentent la notion de partage des poids permettant de réduire considérablement le nombre de paramètres du réseau.

## 2.2 Apprentissage automatique

L'apprentissage automatique (machine Learning) est le champ de l'IA qui permet à une machine d'apprendre, d'améliorer progressivement ses performances sur une tâche spécifique en se basant sur des données, le tout sans être explicitement programmé pour résoudre cette tâche. [14]

### 2.2.1. Classification

La classification est une méthode qui consiste à diviser un grand groupe de données en un certain nombre de catégories souhaitées et distinctes, dans lesquelles on peut attribuer une classification à chaque catégorie.

En d'autres termes, la classification appartient à la catégorie de l'apprentissage supervisé, qui présente l'avantage de fournir les objectifs des données d'entrée. Les applications de classification sont présentées dans de nombreux domaines, notamment l'accréditation de crédit, le diagnostic médical, le marketing ciblé, etc.

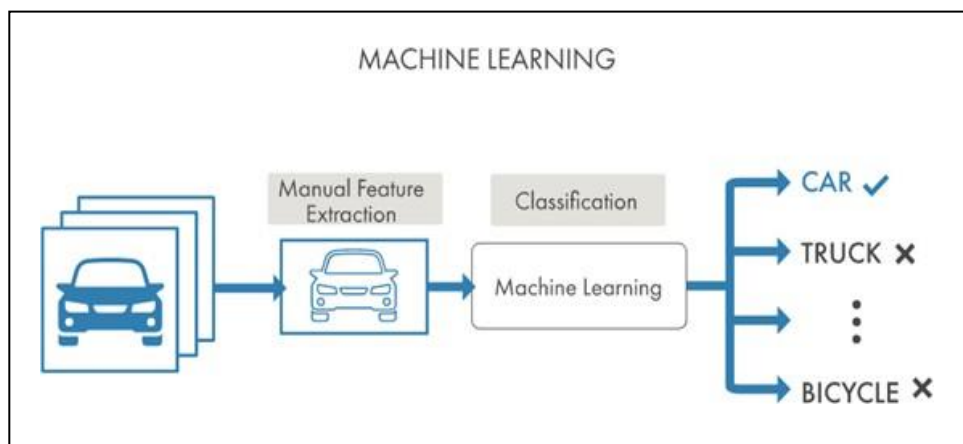


FIGURE 2.2. Exemple de classification avec machine Learning [14]

Comme son nom l'indique, la classification est obtenue en classant les objets dans des sous-catégories à partir d'un groupe général, mais la machine peut-elle distinguer les personnes ou les animaux à l'aide des images, ou peut-elle classer les textes en fonction de leur contenu ? Oui, la machine peut effectuer toutes les classifications à l'aide des données et les algorithmes appropriés, et donner des résultats précis en peu de temps. Donc elle est un exemple de reconnaissance de forme.

Mais dans machine Learning, la classification donner un problème d'identification une nouvelle observation à quel groupe de catégories (sous-populations) appartient, sur la base d'un ensemble de données de formation contenant des observations et dont les catégories sont connues.

### 2.2.1.1. Types de classification

#### 1. Classification binaire

Cette classification est suivie si nous devons diviser et classer les données disponibles en deux catégories différentes (en identifiant le groupe auquel chacune appartient) et la base de la règle de classification. Inclut les contextes nécessitant de décider si l'élément a une propriété spécifique ou une classification binaire typique, par exemple, si une personne est malade et que l'on souhaite diagnostiquer le type de maladie, "x" s'il est infecté ou non.

#### 2. Classification multi classe

Ici, le nombre de classes devrait être supérieur à deux, en fonction du large éventail de données disponibles. Dans l'apprentissage automatique, la classification de classes multiples ou multi-frontières est un problème de classification dans l'une ou plusieurs des trois catégories, en dépit de l'existence d'algorithmes de classification autorisant l'utilisation de plus de deux classes.

### 2.2.2. Fonction d'erreur et optimisation des hyperparamètres

Les hyperparamètres sont essentiels dans l'apprentissage automatique, car elles sont générées souvent des modèles avec des performances très différentes.

Le réglage des hyperparamètres d'apprentissage automatique est une tâche fastidieuse, mais cruciale, car la performance d'un algorithme peut dépendre fortement du choix des hyperparamètres, il est effectué à l'aide de méthodes automatisées qui visent à trouver les hyperparamètres optimaux en moins de temps, en utilisant une recherche éclairée, sans aucun effort manuel au-delà de la configuration initiale.[20]

## 2.3 Apprentissage profond

L'apprentissage en profondeur (Deep Learning) est un type d'apprentissage automatique (machine Learning) qui vise à former et à enseigner à l'ordinateur afin d'exercer des fonctions humaines telles que la distinction des objets visuels et l'identification des sons et des images. Au lieu d'organiser des données, l'apprentissage en profondeur définit ses propres paramètres de base, ce qui permet à la machine d'apprendre par elle-même, et l'intérêt actuel pour l'apprentissage en profondeur est en partie dû au bourdonnement qui entoure l'intelligence artificielle. Les techniques d'apprentissage en profondeur ont amélioré la capacité de classer, reconnaître, détecter, etc.

### 2.3.1. Fonctionnement de l'apprentissage profond

L'apprentissage en profondeur change votre façon de penser aux problèmes que vous résolvez avec l'analyse. Cela va de dire à l'ordinateur comment résoudre un problème, puis de le former à la résolution du problème lui-même.

Une approche classique de l'analyse consiste à utiliser les données disponibles pour concevoir des entités afin d'en déduire de nouvelles variables, puis sélectionner un modèle analytique et enfin estimer les paramètres (ou les inconnues) de ce modèle. Ces techniques peuvent donner des systèmes prédictifs qui ne se généralisent pas bien, car leur exhaustivité et leur exactitude dépendent de la qualité du modèle et de ses caractéristiques. L'ajout de plus de données nécessite de tout recommencer. La nouvelle approche basée sur l'apprentissage en profondeur consiste à remplacer la formulation et la spécification du modèle par des caractérisations hiérarchiques (ou couches) qui apprennent à reconnaître les caractéristiques latentes des données à partir des régularités des couches. Le changement de paradigme avec l'apprentissage en profondeur passe de l'ingénierie des fonctionnalités à la représentation des fonctionnalités.

Pour mieux comprendre comment le faire, prenons un exemple simple : nous voulons identifier un voiture parmi un grand groupe de voitures qui sont filmés, quelle que soit la manière dont ils sont photographiés.

À cette fin, une vaste collection d'images différentes de voitures et d'images d'autres objets doit être assemblée afin de former le réseau de neurones artificiels à la pratique de l'apprentissage en profondeur. Ces images sont converties en données et transmises au réseau après que le poids soit attribué à différents éléments par les cellules nerveuses artificielles. Les informations sont ensuite collectées par la dernière couche de neurones pour déterminer s'il s'agit d'une voiture ou non.

Ensuite, le réseau de neurones artificiels comparera ce résultat à la réponse fournie. S'il y a correspondance, le réseau enregistrera ce succès, car il l'utilisera dans d'autres communes. Si le résultat est négatif, le réseau reconnaîtra le déséquilibre et ajustera le poids des différents neurones pour corriger son erreur. Le processus se fait des milliers de fois jusqu'à ce que l'image soit reconnue et appelée ce type d'apprentissage par "apprentissage supervisé".

La seconde méthode est l'apprentissage non supervisé, qui repose sur des données non classifiées, c'est-à-dire que le réseau doit identifier des modèles dans les ensembles de données. Pour apprendre par eux-mêmes que tous les éléments de l'image peuvent être pertinents.

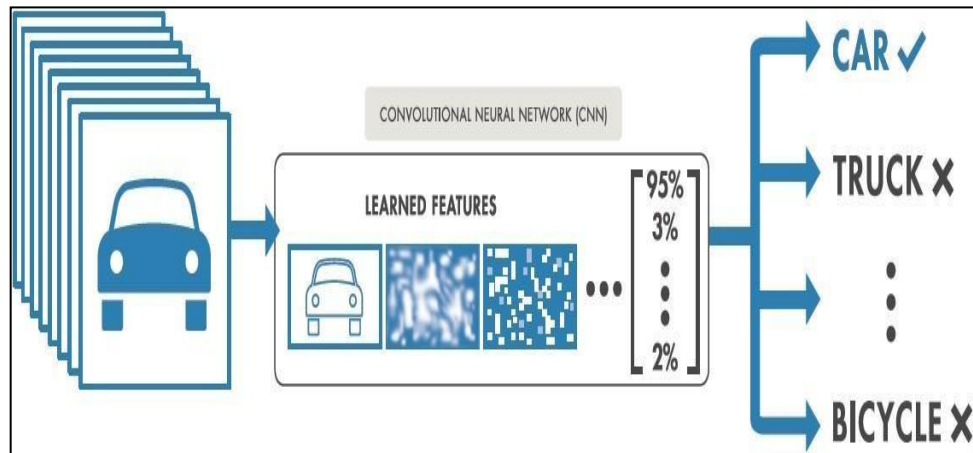


FIGURE 2.3. Exemple de fonctionnement de Deep Learning [14]

## 2.4 Domaine d'application de l'apprentissage

La figure suivante résume les différents domaines de l'apprentissage automatique :

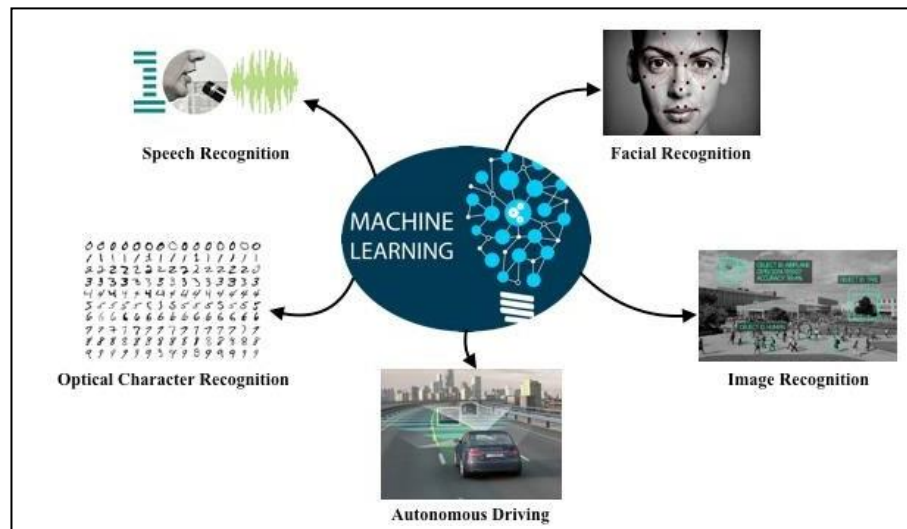


FIGURE 2.4. Domaines d'applications de l'apprentissage [14]

### 1. Reconnaissance d'image :

Il est utilisé pour identifier des objets, des personnes, des lieux, des images numériques, etc. Le cas d'utilisation populaire de la reconnaissance d'image et de la détection de visage est la suggestion de marquage automatique des amis.

### 2. Reconnaissance vocale

La reconnaissance vocale est un processus de conversion d'instructions vocales en texte, et elle est également connue sous le nom de "parole en texte". À l'heure actuelle, les algorithmes d'apprentissage automatique sont largement utilisés par diverses applications de reconnaissance

vocale. L'assistant Google, Siri, Cortana et Alexa utilisent la technologie de reconnaissance vocale pour suivre les instructions vocales.

### **3. Voitures autonomes :**

L'apprentissage automatique joue un rôle important dans les voitures autonomes. Tesla, l'entreprise de fabrication de voitures la plus populaire, travaille sur la voiture autonome. Il utilise une méthode d'apprentissage non supervisée pour former les modèles de voitures à détecter les personnes et les objets pendant la conduite.

### **4. Détection de fraude en ligne :**

L'apprentissage automatique rend notre transaction en ligne sûre et sécurisée en détectant les transactions frauduleuses. Chaque fois que nous effectuons une transaction en ligne, une transaction frauduleuse peut avoir lieu de différentes manières, telles que de faux comptes, de faux identifiants et de voler de l'argent au milieu d'une transaction.

### **5. Diagnostic médical :**

En science médicale, l'apprentissage automatique est utilisé pour le diagnostic des maladies. Avec cela, la technologie médicale se développe très rapidement et est capable de construire des modèles 3D qui peuvent prédire la position exacte des lésions dans le cerveau.

## **2.5 Architectures de réseaux de neurones profonds**

Les deux architectures les plus utilisées en réseaux de neurones profonds sont les réseaux de neurones convolutifs et Les réseaux de neurones récurrents :

### **2.5.1. Les réseaux de neurones convolutifs (CNN)**

Le réseau de neurones convolutifs (CNN) est un type de réseau de neurones artificiels, il est largement utilisé dans le domaine de l'analyse d'images et la reconnaissance vidéo. CNN présente l'avantage d'utiliser des filtres de noyau (kernel) et mettent en œuvre du pooling.

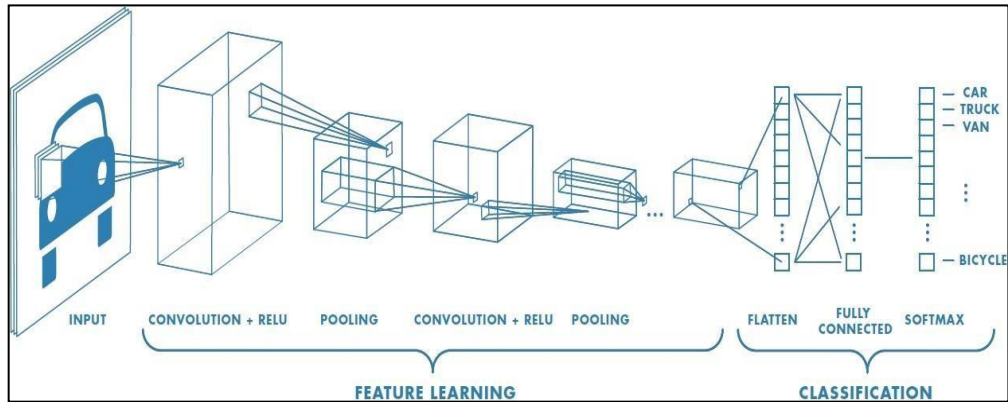


FIGURE 2.5. Architecture du CNN [18]

### 2.5.1.1. L'opération de convolution

La convolution agit comme un filtrage. On définit une taille de fenêtre qui va se balader à travers toute l'image (rappelez-vous qu'une image peut être vue comme étant un tableau). Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image puis elle va se décaler d'un certain nombre de cases (c'est ce que l'on appelle le pas) vers la droite et lorsqu'elle arrivera au bout de l'image, elle se décalera d'un pas vers le bas, ainsi de suite jusqu'à ce que le filtre ait parcouru la totalité de l'image : [18]

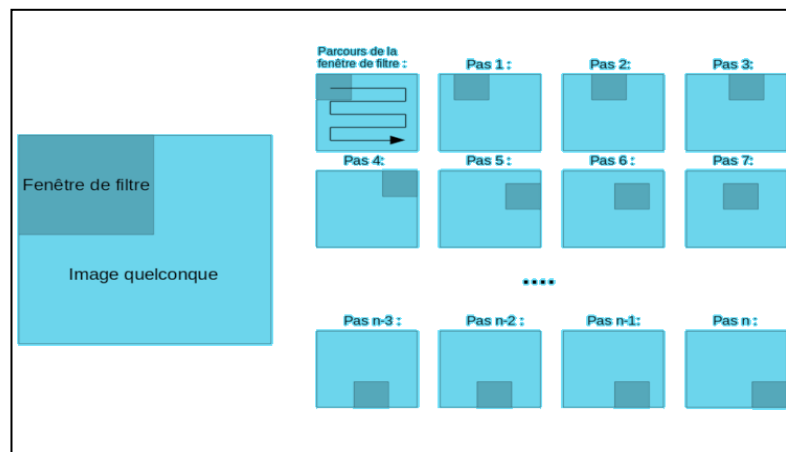


FIGURE 2.6. Schéma du parcours de la fenêtre de filtre sur l'image. [18]

Le but est de se servir des valeurs présentes dans le filtre à chaque pas. Par exemple si l'on définit une fenêtre 3 par 3, cela représentera 9 cases du tableau (9 pixels). La convolution va effectuer une opération avec ces 9 pixels. Il peut s'agir de n'importe quelle opération, par exemple on extrait la valeur la plus grande (soit le pixel avec la plus grande valeur). [6]

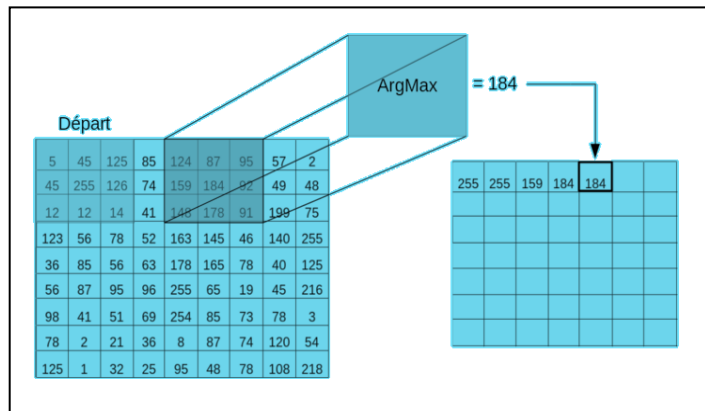


FIGURE 2.7. Convolution avec une argument maximal. [18]

On fait glisser la fenêtre en bleu et à chaque pas on récupère la valeur la plus grande parmi les 9 valeurs de pixels. [18]. La sortie de cette opération (convolution) qui s'appelle « matrice de caractéristique » à des dimensions plus petites que celle de l'image en entrée

### 2.5.1.2. Les couches de CNN

La structure de CNN consiste en un ensemble de couches de traitement indépendantes :

#### 1. Couche convolution

Cette couche est le composant principal des réseaux de neurones convolutifs et constitue leur première couche. Sa fonction est de localiser un ensemble de caractéristiques(features) dans les images qui ont été données sous forme des entrées, raison pour laquelle nous effectuons un processus appelé méta-filtrage, dans lequel nous faisons glisser la fonction d'image dans une forme de fenêtre et calculons le produit de convolution entre la feature et chaque portion de l'image balayée.

Pour mesurer le volume de la couche de convolution, nous avons besoin de 3 hyperparamètres :

1. Profondeur de la couche : Le nombre de neurones qui ont une corrélation directe avec le même champ du récepteur.
2. Le pas : Le pas contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
3. La marge (à 0) ou zéro padding : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cette marge permet de contrôler la dimension spatiale du volume de sortie. [19]



## 2. Couche de pooling (POOL)

La chose la plus commune à propos de cette couche est qu'elle est placée entre deux couches de convolution, ce qui lui permet de recevoir de nombreuses feature maps en entrée, l'opération de pooling consiste à minimiser la taille de l'image et à en préserver les propriétés importantes.

Parmi les caractéristiques connues de pooling est de donner une grande puissance de calcul, et malgré la très faible représentation (et donc la perte d'informations associées), toutes les tendances actuelles consistent à utiliser des filtres de petite taille (type  $2 \times 2$ ). Mais le risque de sur-apprentissage posait de nombreux problèmes qui permettaient très bien d'éviter la couche de pooling

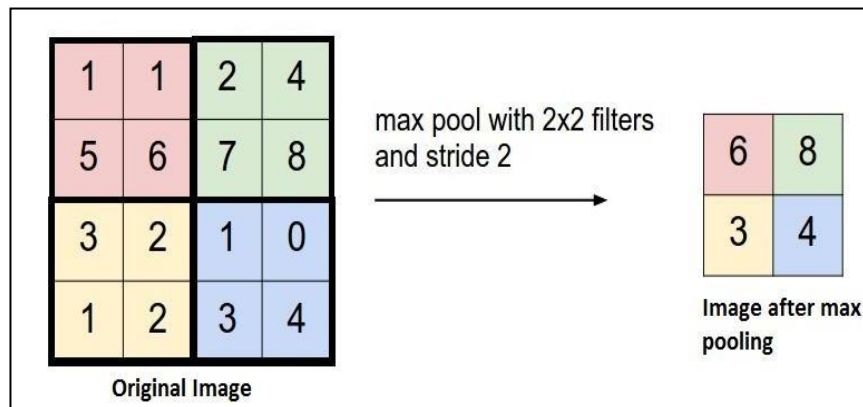


FIGURE 2.8. Pooling avec un filtre 2x2 et un pas de 2. [16]

## 3. Couche de correction (ReLU)

Les fonction d'activation tel que : Relu, tanh, sigmoid , on a but pour améliorer l'efficacité du traitement en. La fonction ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire, augmente les propriétés non linéaires de la fonction de décision elle remplace toutes les valeurs négatives reçues en entrées par des zéros.

## 4. Couche entièrement connectée

Ce type de couche reçoit un vecteur d'entrée et produit un nouveau vecteur de sortie. Pour cela, il applique une combinaison linéaire et éventuellement une fonction d'activation aux valeurs reçues en entrée.

## 5. Couche de perte (LOSS)

La dernière couche dans le réseau , spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte « Softmax » est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. [16]

### 2.5.1.3. Les paramètres de CNN

#### 1. Nombre de filtres

En raison de la réduction de la taille des images intermédiaires en fonction de la profondeur du traitement, le nombre de filtres dans les calques proches de l'entrée est réduit. Il est très important de conserver le nombre de sorties intermédiaires (c'est-à-dire de multiplier le nombre d'images intermédiaires par le nombre de positions de pixels) afin de préserver les informations d'entrée.

#### 2. Forme du filtre

Les formes de filtrage sont souvent choisies en fonction de l'ensemble de données, en raison de la grande diversité de leurs formes dans la littérature. Les meilleurs résultats sont ceux obtenus sur MNIST ( $28 \times 28$ ) dans une plage de  $5 \times 5$  dans la première couche.

#### 3. Forme du Max Pooling

Les valeurs typiques sont  $2 \times 2$ . Des volumes d'entrée très importants peuvent justifier une mise en commun de  $4 \times 4$  dans les couches inférieures. Cependant, le choix de formes plus grandes réduira considérablement la dimension du signal et peut entraîner une perte excessive d'informations.

### 2.5.2. R-CNN

Le R-CNN est une architecture de détection d'objet. Le R-CNN commence par extraire des régions intéressantes de l'image, puis il utilise ces régions comme données d'entrée pour un CNN. Cette séparation en régions permet de détecter plusieurs objets de plusieurs classes différentes dans une même image. Cette solution proposée par Girshick [17] a permis d'améliorer la précision des modèles de détection.

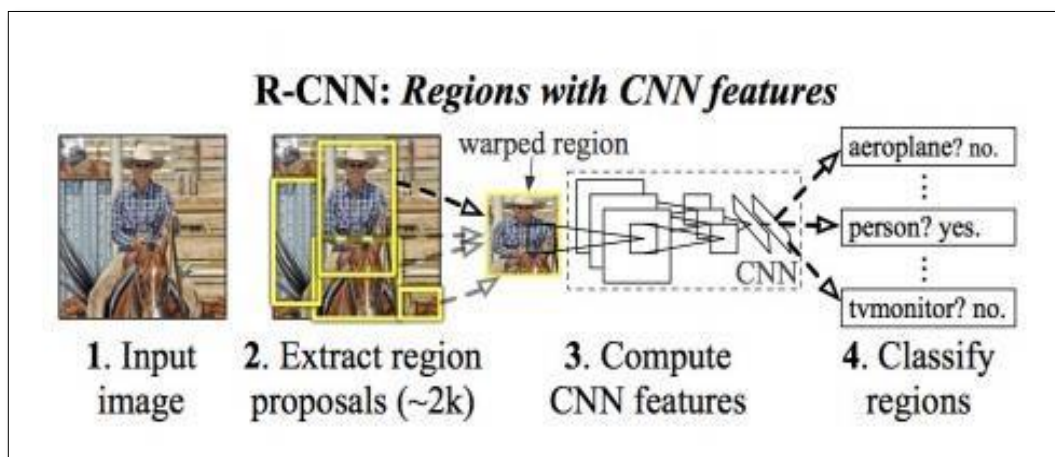


FIGURE 2.9. Architecture R-CNN [17]

Dans le R-CNN présenté en figure 2.9, les régions sont extraites grâce à une recherche sélective proposée par Uijlings [18]. Celle-ci utilise la structure de l'image et plusieurs techniques de partitionnement pour récupérer toutes les régions intéressantes possibles

### 2.5.3. Faster R-CNN

Le Faster R-CNN est une amélioration du R-CNN dans sa précision et sa rapidité à l'entraînement [19].

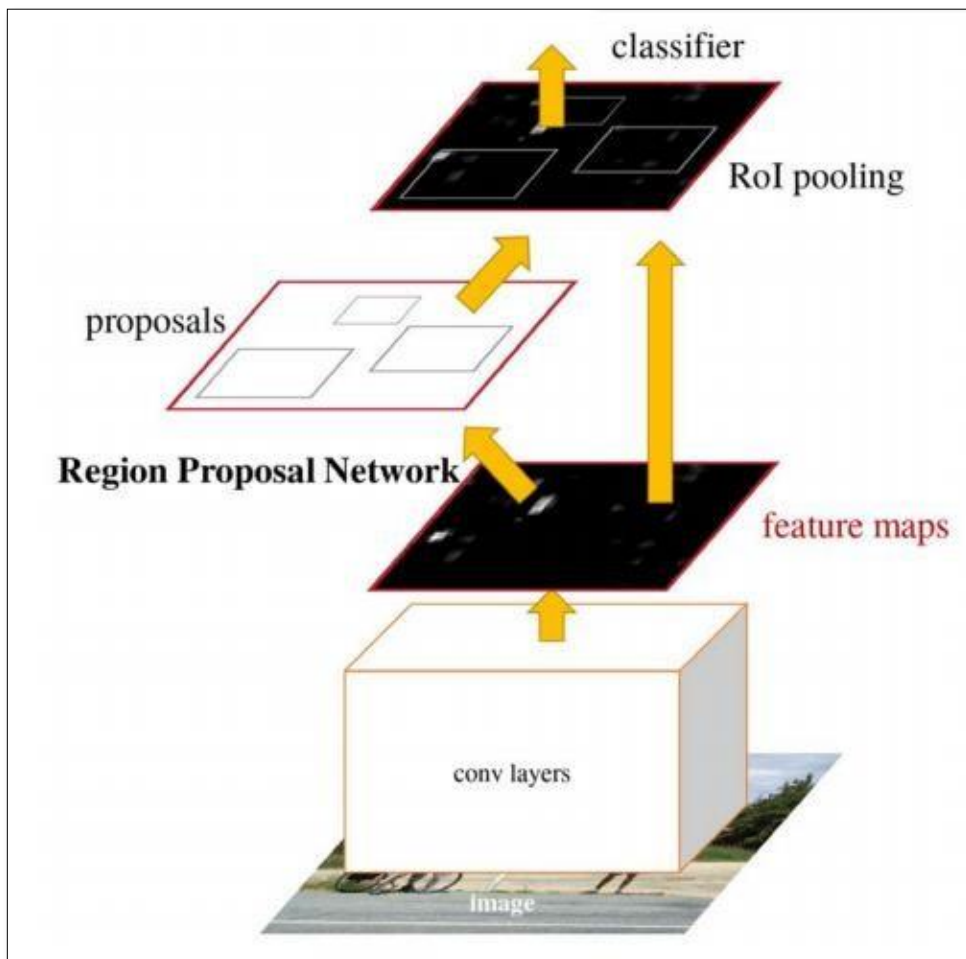


FIGURE 2.10. Architecture Faster R-CNN [19]

L'architecture Faster R-CNN (fig. 2.10) 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. 2.11) 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 [19]

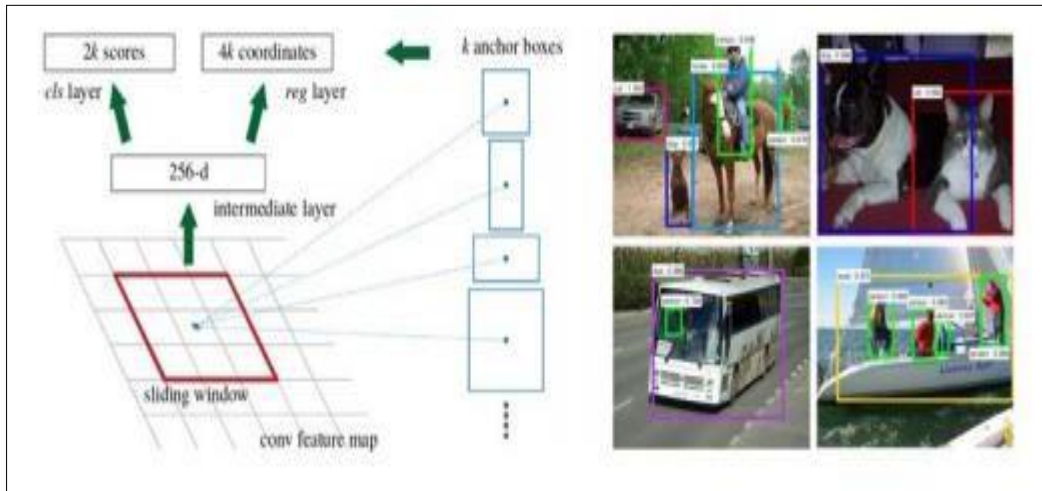


FIGURE 2.11. Faster R-CNN : Fenêtre coulissante et résultat [19]

### 2.5.4. Segmentation d'instance

La segmentation d'instance est un problème de vision ordinaire. 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 classifier chaque pixel d'une image à une instance d'un objet.

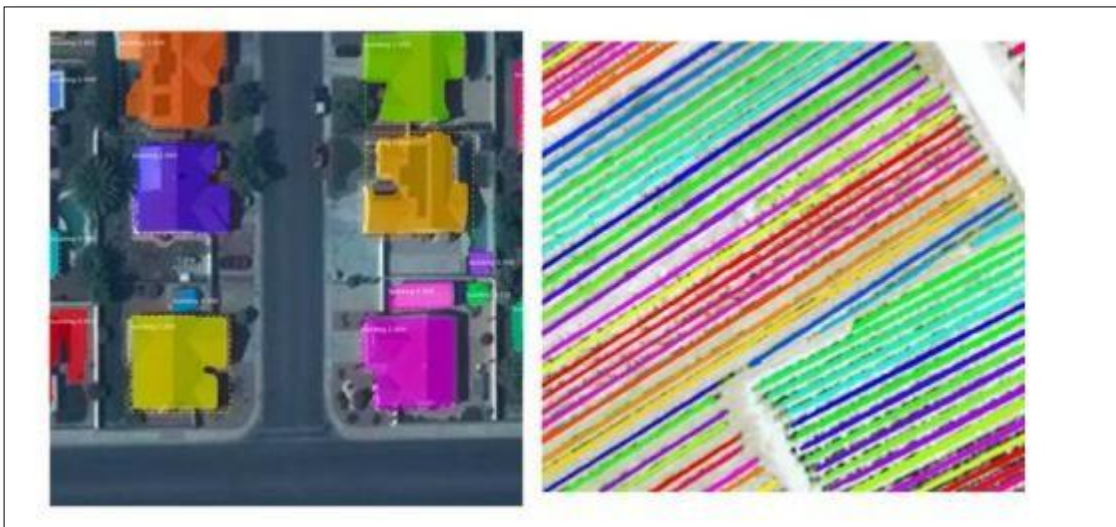


FIGURE 2.12. Exemple de segmentation d'instance [20]

En figure 2.12, 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.

### 2.5.5. 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 [21]

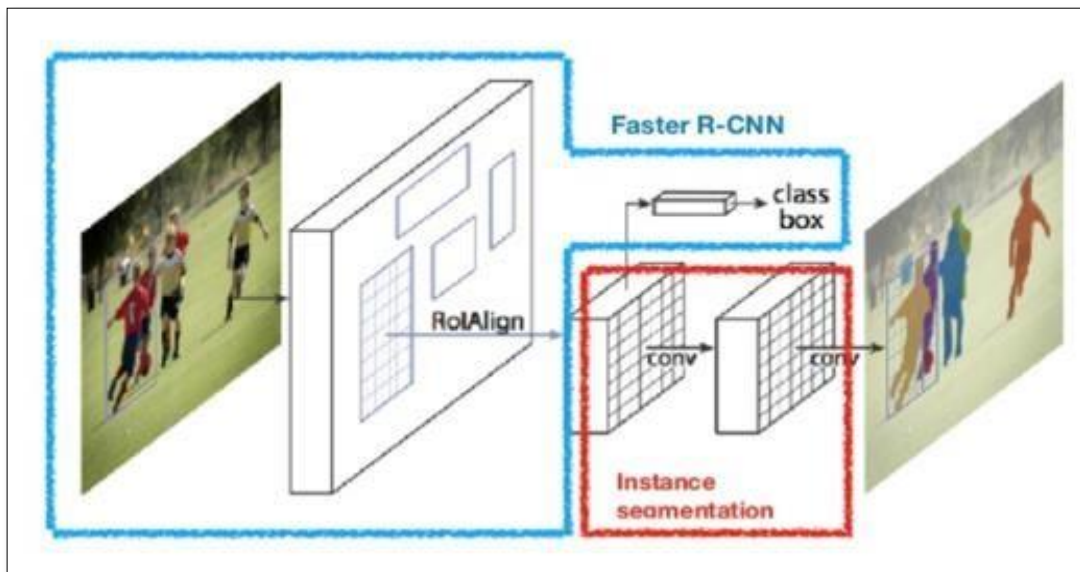


FIGURE 2.13. Architecture Mask R-CNN [21]

Comme on peut le voir en figure 2.13, 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 [21]

### 2.5.6. ResNet

Les Residual Network (ResNet) sont des réseaux de neurones qui implémentent l'apprentissage résiduel. Cette innovation permet de faciliter l'apprentissage des couches neuronales profondes. L'apprentissage résiduel découle de l'observation que les couches profondes d'un réseau de neurones avaient des difficultés à converger. L'apprentissage résiduel permet à l'information de contourner certaines couches et réduit ainsi le problème de dégradation [22]



### 2.5.7. Vgg Net

L'architecture du réseau VGG a été présentée par Simonyan et Zisserman dans leur article de 2014 [25], Very Deep Convolutional Networks for Large Scale Image Recognition.

Ce réseau se caractérise par sa simplicité, n'utilisant que  $3 \times 3$  couches convolutives empilées les unes sur les autres de plus en plus en profondeur. La réduction de la taille du volume est gérée par la mise en commun maximale. Deux couches entièrement connectées, chacune avec 4 096 nœuds, sont ensuite suivies d'un classificateur softmax (ci-dessus).

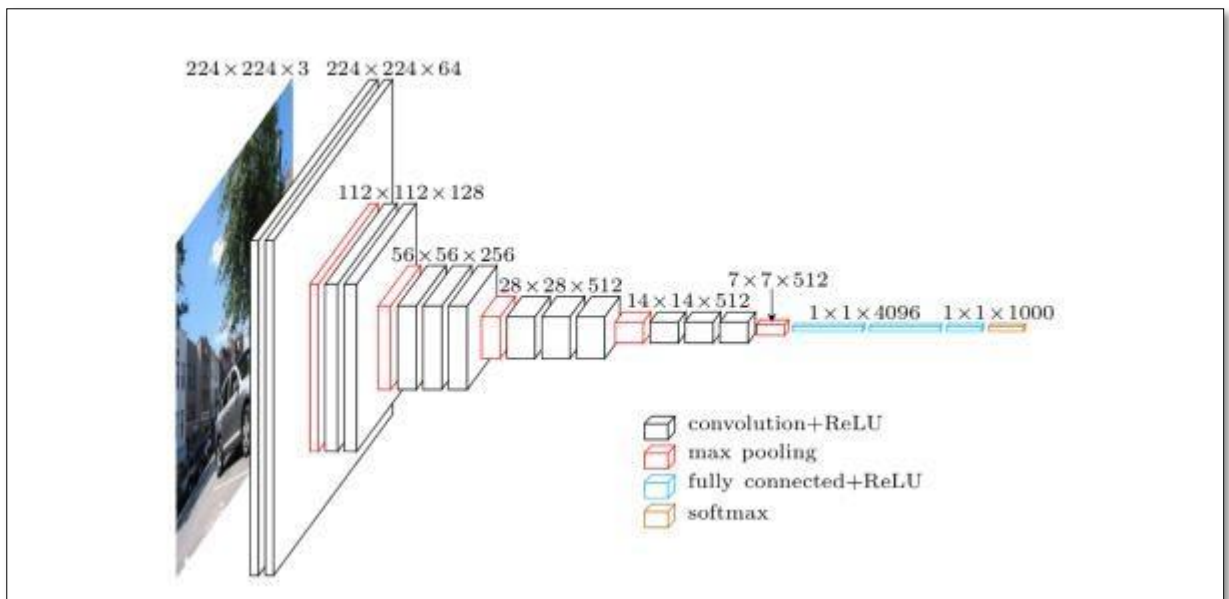


FIGURE 2.14. Une visualisation de l'architecture VGG [25]

Les «16 » et «19 » représentent le nombre de couches de poids dans le réseau (colonnes D et E de la figure 2.15 ci-dessous):

| ConvNet Configuration       |                        |                               |  |  |   |
|-----------------------------|------------------------|-------------------------------|--|--|---|
| A                           | A-LRN                  | B                             | C  | D  | E   |
| 11 weight layers            | 11 weight layers       | 13 weight layers              | 16 weight layers                           | 16 weight layers                           | 19 weight layers  |
| input (224 × 224 RGB image) |                        |                               |  |  |   |
| conv3-64                    | conv3-64<br><b>LRN</b> | conv3-64<br><b>conv3-64</b>   | conv3-64<br>conv3-64                       | conv3-64<br>conv3-64                       | conv3-64<br>conv3-64                                    |
| maxpool                     |                        |                               |  |  |   |
| conv3-128                   | conv3-128              | conv3-128<br><b>conv3-128</b> | conv3-128<br>conv3-128                     | conv3-128<br>conv3-128                     | conv3-128<br>conv3-128                                  |
| maxpool                     |                        |                               |  |  |   |
| conv3-256<br>conv3-256      | conv3-256<br>conv3-256 | conv3-256<br>conv3-256        | conv3-256<br>conv3-256<br><b>conv1-256</b> | conv3-256<br>conv3-256<br><b>conv3-256</b> | conv3-256<br>conv3-256<br>conv3-256<br><b>conv3-256</b> |
| maxpool                     |                        |                               |  |  |   |
| conv3-512<br>conv3-512      | conv3-512<br>conv3-512 | conv3-512<br>conv3-512        | conv3-512<br>conv3-512<br><b>conv1-512</b> | conv3-512<br>conv3-512<br><b>conv3-512</b> | conv3-512<br>conv3-512<br>conv3-512<br><b>conv3-512</b> |
| maxpool                     |                        |                               |  |  |   |
| conv3-512<br>conv3-512      | conv3-512<br>conv3-512 | conv3-512<br>conv3-512        | conv3-512<br>conv3-512<br><b>conv1-512</b> | conv3-512<br>conv3-512<br><b>conv3-512</b> | conv3-512<br>conv3-512<br>conv3-512<br><b>conv3-512</b> |
| maxpool                     |                        |                               |  |  |   |
| FC-4096                     |                        |                               |  |  |   |
| FC-4096                     |                        |                               |  |  |   |
| FC-1000                     |                        |                               |  |  |   |
| soft-max                    |                        |                               |  |  |   |

FIGURE 2.15. Réseaux convolutifs très profonds pour la reconnaissance d'image à grande échelle [25]

En 2014, les réseaux de 16 et 19 couches étaient considérés comme très profonds (bien que nous ayons maintenant l'architecture ResNet qui peut être formée avec succès à des profondeurs de 50 à 200 pour ImageNet et plus de 1000 pour CIFAR-10).

Simonyan et Zisserman ont trouvé l'entraînement VGG16 et VGG19 difficile (en particulier en ce qui concerne la convergence sur les réseaux plus profonds), donc afin de faciliter l'entraînement, ils ont d'abord formé des versions plus petites de VGG avec moins de couches de poids (colonnes A et C).

Les réseaux plus petits ont convergé et ont ensuite été utilisés comme initialisations pour les réseaux plus grands et plus profonds - ce processus est appelé pré-formation.

Tout en ayant un sens logique, la pré-formation est une tâche fastidieuse et fastidieuse, qui nécessite la formation d'un réseau entier avant de pouvoir servir d'initialisation pour un réseau plus profond.

### 2.5.8. Les réseaux de neurones récurrents (RNN)

Un réseau de neurones récurrents (RNN) est une classe de réseaux de neurones artificiels où les connexions entre les nœuds forment un graphe dirigé le long d'une séquence. Cela lui permet de présenter un comportement dynamique temporel pour une séquence temporelle. Contrairement aux réseaux neuronaux à anticipation, les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées. Cela les rend applicables à des tâches telles que la reconnaissance de l'écriture manuscrite ou la reconnaissance vocale connectée, non segmentées.

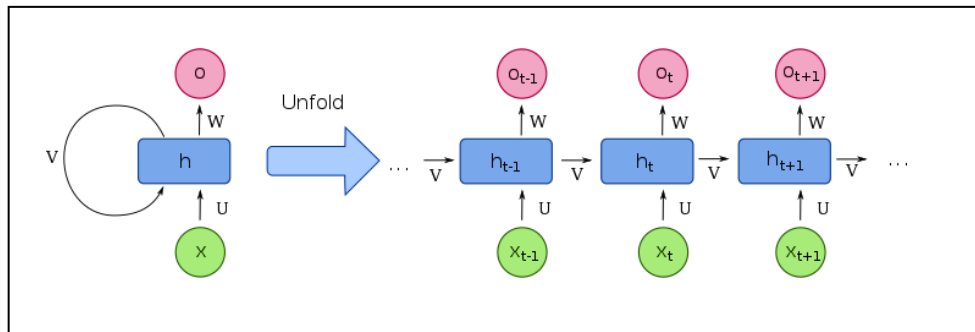


FIGURE 2.16. Architecture de RNN [23]

Le terme "réseau neuronal récurrent" est utilisé indifféremment pour désigner deux grandes classes de réseaux ayant une structure générale similaire, où l'une est impulsion finie et l'autre impulsion infinie. Les deux classes de réseaux présentent un comportement dynamique temporel. [23] Un réseau récurrent à impulsions finies est un graphe acyclique dirigé qui peut être déroulé et remplacé par un réseau neuronal strictement à anticipation, tandis qu'un réseau récurrent à impulsions infinies est un graphe cyclique dirigé qui ne peut pas être déroulé.

Les réseaux à impulsions finies et les réseaux récurrents à impulsions infinies peuvent avoir un état stocké supplémentaire et le stockage peut être sous le contrôle direct du réseau de neurones. Le stockage peut également être remplacé par un autre réseau ou graphique, s'il intègre des délais ou des boucles de retour. De tels états contrôlés sont appelés état bloqué ou mémoire bloquée et font partie de réseaux de mémoire à court terme (LSTM) et d'unités récurrentes déclenchées. [24]

Les RNN ont connu un grand succès dans les tâches d'étiquetage et de prédiction de séquences, tel que la reconnaissance de l'écriture manuscrite et la modélisation du langage. Dans la modélisation acoustique pour la reconnaissance de la parole, cependant, où les réseaux de neurones profonds (DNN) sont l'état de la technique établi, les RNN ont récemment reçu peu d'attention au-delà des tâches de reconnaissance téléphonique à petite échelle, les exceptions notables étant le travail de Robinson. [15]



### 2.5.8.1. Application RNN

Les réseaux de neurones récurrents jouent un rôle actif dans le domaine de la reconnaissance de la parole, tel qu'utilisé par Google Translation Network. Il a déjà été appliqué dans le domaine de la traduction automatique et de la commande de robot, ainsi que dans la prédiction de séries chronologiques. Il a réussi la composition musicale et l'apprentissage grammatical, à détecter la parité des protéines et à prédire la sous-cellularisation cellulaire et la prévisibilité dans le secteur des soins médicaux.

### 2.5.9. Apprentissage par renforcement

L'apprentissage par renforcement (RL pour Reinforcement Learning) est une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences successives, ce qu'il convient de faire de façon à trouver la meilleure solution.

Dans un tel problème, on dit qu'un « agent » interagit avec « l'environnement » pour trouver la solution optimale. L'apprentissage par renforcement diffère fondamentalement des problèmes supervisés et non supervisés par ce côté interactif et itératif : l'agent essaie plusieurs solutions « d'exploration », observe la réaction de l'environnement et adapte son comportement (les variables) pour trouver la meilleure stratégie (il « exploite » le résultat de ses explorations). Un des concepts clés de ce type de problèmes est l'équilibre entre ces phases d'exploration et d'exploitation. Cette méthode est particulièrement adaptée aux problèmes nécessitant un compromis entre la quête de récompenses à court terme et celle de récompenses à long terme. Parmi les exemples de problèmes traités de cette façon, on peut évoquer : apprendre à un robot à marcher en terrain difficile.

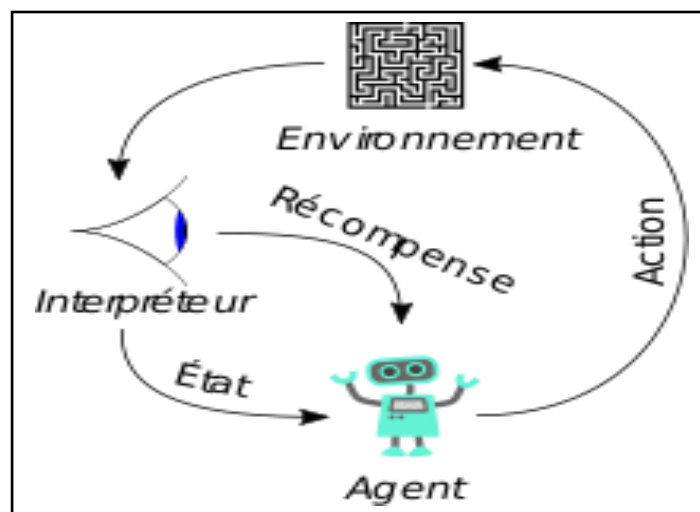


FIGURE 2.17. Exemple d'un agent utilise l'apprentissage par renforcement [15]

## 2.6 L'apprentissage automatique & la classification des déchets

Avant de parler des différents travaux d'apprentissage automatique dans la catégorisation des images de déchets, nous prendrons un aperçu des techniques traditionnelles, puis nous plongerons dans les technologies intelligentes

### 2.6.1. Les approches traditionnelles :

#### 1. Dans le but d'identifier et suivre la collecte sélective en stockant les données des propriétaires et des informations sur leurs bacs.

- ✚ Dans [26] ont proposé une méthode pour améliorer la qualité de la collecte sélective en suivant le flux de déchets d'une ville, où chaque déchet est détecté à partir d'informations stockées sur une étiquette RFID (associée aux déchets) et lors de l'étape de traitement des déchets étiquettes RFID sont lus pour fournir des informations pertinentes.
- ✚ Dans [27] a été proposé d'estimer le volume des déchets ménagers sur la base d'une analyse d'image du contenu du couvercle du conteneur ouvert à partir d'étiquettes RFID, telles que l'étiquette serait utilisée pour associer chaque bac à l'adresse de la maison qui en est propriétaire.
- ✚ Dans [28], les auteurs ont discuté des applications de la technologie RFID dans l'autogestion des produits, en mettant l'accent sur la gestion des déchets solides municipaux et ses implications environnementales, également en identifiant chaque bac à partir d'une étiquette RFID.
- ✚ Dans [29], les auteurs ont proposé un capteur RFID (pour stocker le poids de la poubelle et identifier le propriétaire de chaque poubelle) qui mesure dans chaque opération de collecte, le poids de la poubelle jetée dans le camion poubelle.
- ✚ Dans [30], les auteurs ont suivi les consommateurs en les identifiant à l'aide d'une étiquette RFID associée à leur corbeille. Les consommateurs sont également financièrement récompensés en fonction du poids de leurs emballages de recyclage ou de la quantité qu'ils ont recyclée.

- ✚ Dans [31] ont utilisé des étiquettes RFID dans leurs produits pour fournir des informations aux prestataires de services de recyclage afin d'aider à décider du traitement approprié de ce produit.

### **Critiques :**

- À ce stade, les travaux présentés s'appuient sur le problème de l'identification, du suivi et de l'analyse des déchets mis au rebut afin de stimuler le comportement responsable de la population pour une collecte sélective.
- Cependant, aucune de ces études ne visait à aider la population à éliminer correctement les ordures.

## **2. Dans le but d'aider les consommateurs à éliminer correctement les déchets,**

- ✚ Sinha et al. [32] ont développé une méthode capable de sélectionner le meilleur conteneur de recyclage compte tenu des déchets jetés par l'utilisateur. Ils ont présenté un modèle utilisant Ontology Web Language (OWL) pour trier les déchets intelligents pour un meilleur recyclage des matériaux. Ils utilisent l'ontologie OWL pour représenter des informations sur la quantité de matériaux recyclables précieux contenus dans chaque déchet

### **Critiques :**

- Les approches traditionnelles de la classification des déchets se sont concentrées sur les propriétés physiques, telles que le poids, la forme (solide, liquide, aqueux ou gazeux) et le type de processus qui a généré les déchets.
- Ces propriétés peuvent ensuite être utilisées par un algorithme d'apprentissage automatique pour classer les déchets selon les catégories définies.

### **2.6.2. Les approches intelligentes :**

De nombreux modèles différents ont été développés pour la classification des images, comme RNN, SVM, ANN, etc., mais Convolutional Neural Network, qui est un algorithme d'apprentissage automatique, a vraiment mieux performé qu'eux tous.

- ✚ En 2016, Yang et Thung [33] ont publié un ensemble de données **Trashnet** (figure 2.18) composé de six classes : verre, papier, carton, plastique, métal et poubelle en utilisant l'algorithme SVM de machine Learning, avec une transformation de caractéristique invariante d'échelle (SIFT<sup>i</sup>) et un réseau de neurones convolutifs (CNN). Leurs approches ont atteint un taux de précision de 63% et 22% pour le SVM formé et le CNN, respectivement.



FIGURE 2.18. Les 6 classes de data set TrashNet [33]

Les images de cet ensemble de données ont été prises par plusieurs appareils mobiles tels que Apple iPhone 7 Plus, Apple iPhone 5S et Apple iPhone SE. Les descriptions détaillées de cet ensemble de données seront présentées dans le troisième chapitre. Plusieurs études concernant le problème de classification des déchets utilisant l'ensemble de données Trashnet pour évaluer les approches proposées qui sont résumées dans les paragraphes suivants.

- ✚ En 2018 Aral et al. [34] ont utilisé les modèles d'apprentissage par transfert provenant de plusieurs modèles CNN bien connus pour la classification d'images, notamment Densenet<sup>ii</sup>, pour classer les déchets dans l'ensemble de données Trashnet. Dans leurs expériences, les auteurs ont utilisé 70% de l'ensemble de données Trashnet pour la formation, 13% pour la validation et 17% pour les tests. De plus, le lot et la taille d'entrée ont été sélectionnés respectivement à 8 et  $224 \times 224$ . Selon les résultats expérimentaux, un modèle d'apprentissage par transfert de DenseNet a archivé la meilleure précision qui donne 95%.

- ✚ Ensuite en 2018, Bircanoglu et al. [35] ont développé un modèle de réseau neuronal convolutionnel , pour la classification des déchets. Bien que le mdel n'ait atteint que 81% en termes de précision pour l'ensemble de données Trashnet avec 70% d'images pour la formation, 13% pour la validation et 17% pour les tests, il a réduit la complexité du temps en réduisant le nombre de paramètres de sept millions à trois millions.
- ✚ En 2019 [36] les auteurs ont évalué l'utilisation de plusieurs modèles CNN, notamment ResNet<sup>iiii</sup> pour la classification automatique des déchets. Dans cette étude, les auteurs ont utilisé 80% de l'ensemble de données Trashnet pour la formation, 10% pour la validation et les 10% restants pour les tests. Les meilleurs résultats de performance ont été obtenus en utilisant une architecture basée sur ResNet avec 88,66% en termes de précision moyenne pour l'ensemble de données Trashnet.

### **Critique :**

- Par conséquent, ces modèles (SIFT , ResNet et DenseNet ) sont des modèles légers pour plusieurs systèmes qui restreignent les périphériques matériels.

**Notre travail est similaire à celui de Yang et al. [34], à la différence près que nous avons formé les images avec d'autres techniques NN, telles que le simple réseaux CNN pour éviter les critiques citer précédents et aussi d'augmenter le taux de modèle.**

**De plus, cette étude recueille également un autre ensemble de données pour créer un modèle qui catégorise une image contenant de nombreux éléments. Cet ensemble de données contient trois types de déchets, y compris le verre, le plastique et le métal**

## **2.7 Conclusion**

Dans ce chapitre, nous avons présenté le domaine de l'intelligence artificielle et clarifier l'apprentissage automatique, y compris la classification et les implications de la relation entre l'apprentissage automatique et l'apprentissage en profondeur, qui a surmonté les méthodes traditionnelles de manipulation d'images et de texte en rendant les systèmes plus rapides et plus faciles. Aussi dans le dernier point nous avons établi un état de l'art sur les différentes approches et techniques de classification des déchets

Le chapitre qui suit, on va présenter notre contribution ainsi que notre modèles les algorithmes utilisés dans notre architecture.



---

<sup>i</sup> La transformation de caractéristique invariante à l'échelle (SIFT) est un algorithme de détection de caractéristique en vision par ordinateur pour détecter et décrire les caractéristiques locales dans les images.

<sup>ii</sup> L'architecture DenseNet est une architecture nouvelle, c'est une extension logique de ResNet.

<sup>iii</sup> Un réseau neuronal résiduel (ResNet) est un réseau neuronal artificiel (ANN) d'un type qui s'appuie sur des constructions connues des cellules pyramidales du cortex cérébral. Les réseaux de neurones résiduels le font en utilisant des connexions de saut ou des raccourcis pour sauter par-dessus certaines couches

# Chapitre 3

*Contribution*



## **1.1. Introduction**

Dans ce chapitre, nous présentons notre Système de classification de déchets. En premier temps, nous débutons par une introduction qui constitue une brève définition de notre système de classification automatique, ensuite nous présenterons les deux contributions de classification mono objet et multi objets proposées. Puis nous détaillerons les différents modules qui le composent.

## **1.2. Problème étudiée**

La méthode actuelle de séparation des déchets est la méthode de sélection manuelle, dans laquelle les gens sont employés pour séparer divers matériaux, lorsqu'ils sont vulnérables aux maladies dues aux substances nocives dans les ordures. Dans cet esprit, nous avons stimulé cela pour développer un système automatisé capable de trier les déchets pour être plus précis et gagner du temps de tri. Parmi les avantages du système : le recyclage des déchets séparés utiles et leur conversion en énergie et en carburant pour la croissance de l'économie. Le système que nous proposons dépend de l'algorithme du réseau neuronal convolutif. Avec une profondeur croissante, la reconnaissance du réseau neuronal convolutif peut être augmentée.

## **1.3. Architecture générale de l'application**

Dans cette section, nous proposons un Système de Classification des déchets Solides en temps réel. La figure suivante présente l'architecture générale de notre système de classification proposé.

Après le processus de collecte des ordures, un système de traitement d'image sur ordinateur peut être utilisé pour le diviser en différents types. La méthode que nous suggérons utilise un appareil photo numérique connecté à un ordinateur avec un programme de classification des déchets. Le programme utilisé dans ce système utilise des techniques de traitement d'image basées sur le réseau convolutionnel et le open cv pour reconnaître les choses. La caméra prend

des photos des déchets qui sont passés séparément sur la bande transporteuse et les envoie au système programmé sur l'ordinateur pour l'identification. Enfin, selon les résultats du système, les déchets sont dirigés vers des conteneurs spécifiques.

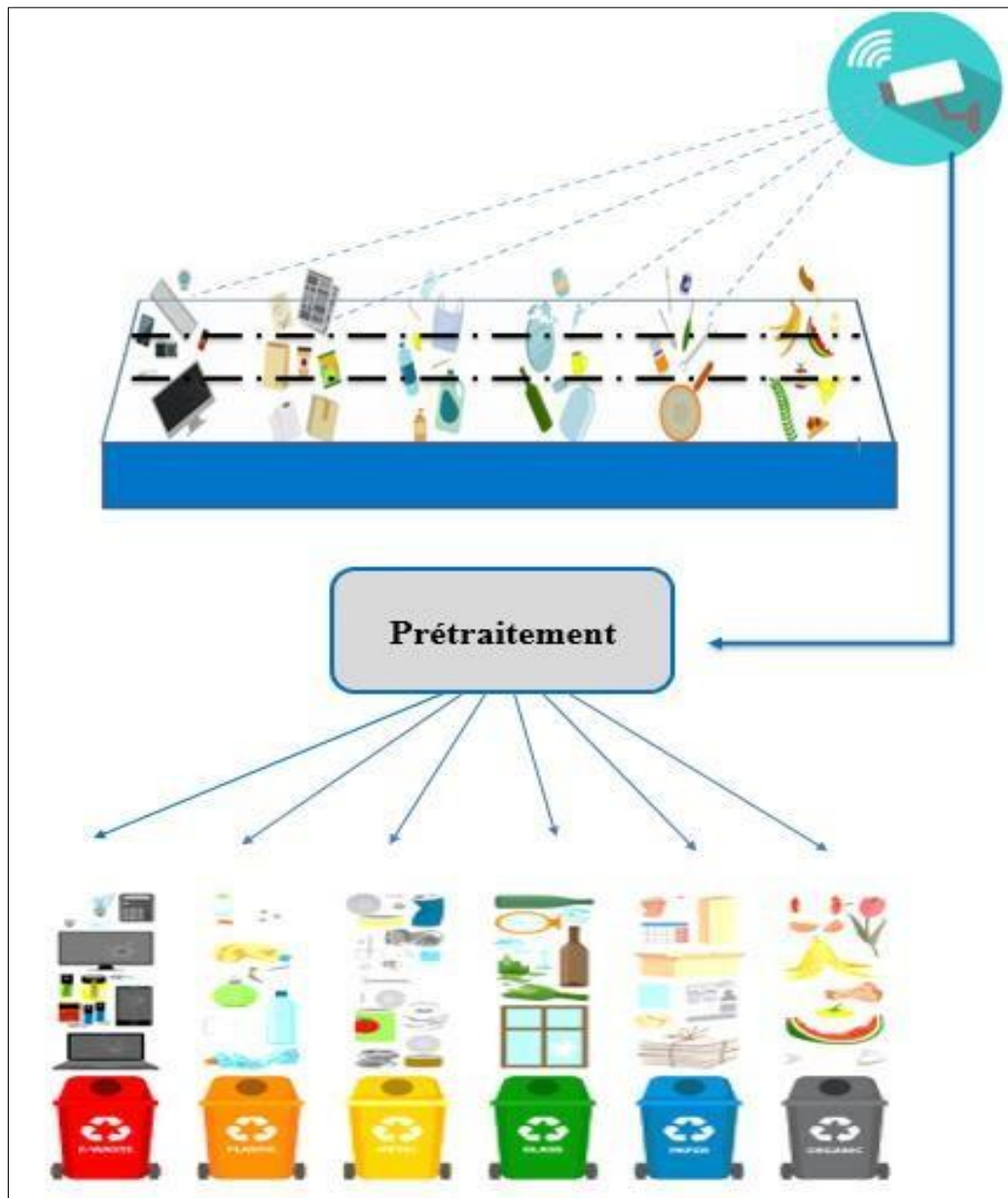


FIGURE 3.1. Architecture générale de l'application

### **1.3.1. Prétraitement :**

Dans la phase de prétraitement illustré dans la figure précédente est le cœur du programme, c'est lui qui va effectuer la tâche que nous cherchons à accomplir, les sections suivant expliquant bien cette point.

## **1.4. Méthodologie**

L'idée proposée se concentre principalement sur la classification et l'identification des déchets déversés dans les ordures. Habituellement, les déchets non triés sont déversés dans une décharge et ils sont décomposés, ce qui prend cependant des milliers d'années dans le cas des déchets non biodégradables. Ce projet propose une idée où un système informatique à lui seul peut identifier les déchets sans aucune intervention humaine sur la base du matériau du déchet, quelles que soient sa forme, sa couleur et sa taille, et les classer. Le système proposé peut apprendre par lui-même et peut donc constamment se mettre à jour en cas de nouveaux déchets. L'avantage de ce système proposé est qu'il inclurait une classification facile des déchets solides, une décomposition des déchets, des risques sanitaires réduits et un processus plus rapide qui ne nécessite qu'un investissement initial négligeable pour cela.

La figure suivante présente le principe système proposé.

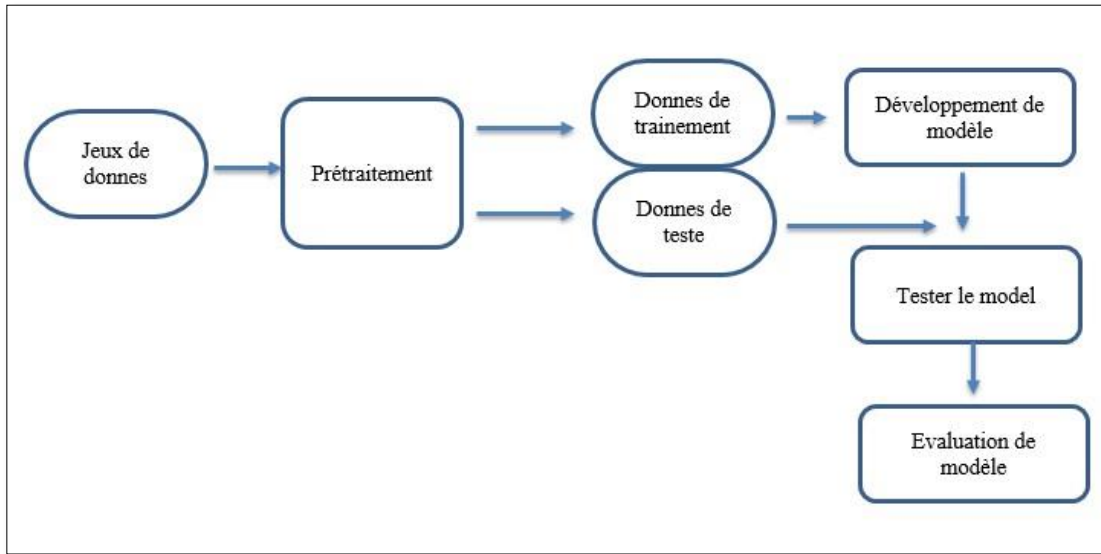


FIGURE 3.2. Diagramme de la méthodologie

La première phase consiste à collecter les données, ensuite dans la phase de prétraitement les données sont normalisées, et organisées avec la même taille et finalement divisées en deux groupes (train & test). Les données (train) sont utilisées pour développer le modèle et les données « test » sont utilisées pour tester le modèle, ensuite le modèle doit passer par la dernière phase d'évaluation.

## 1.5. Architecture des nôtres réseaux proposés

Au cours de nos expérimentations, nous avons créé trois modèles (modèle 1, modèle 2 et modèle 3) avec différentes architectures, où on a appliqué le modèle 1 sur CNN et les modèles 2 sur Resnet-34 et 3 sur Mask rcnn.

Dans ce qui suit on présente l'architecture des trois modèles :

### 1.5.1. Model 1

Le premier modèle que nous présentons dans la figure suivante c'est un modèle CNN-2D, est

composé de 3 couches de convolution et 3 couches de maxpooling et 3 couches de sortie.

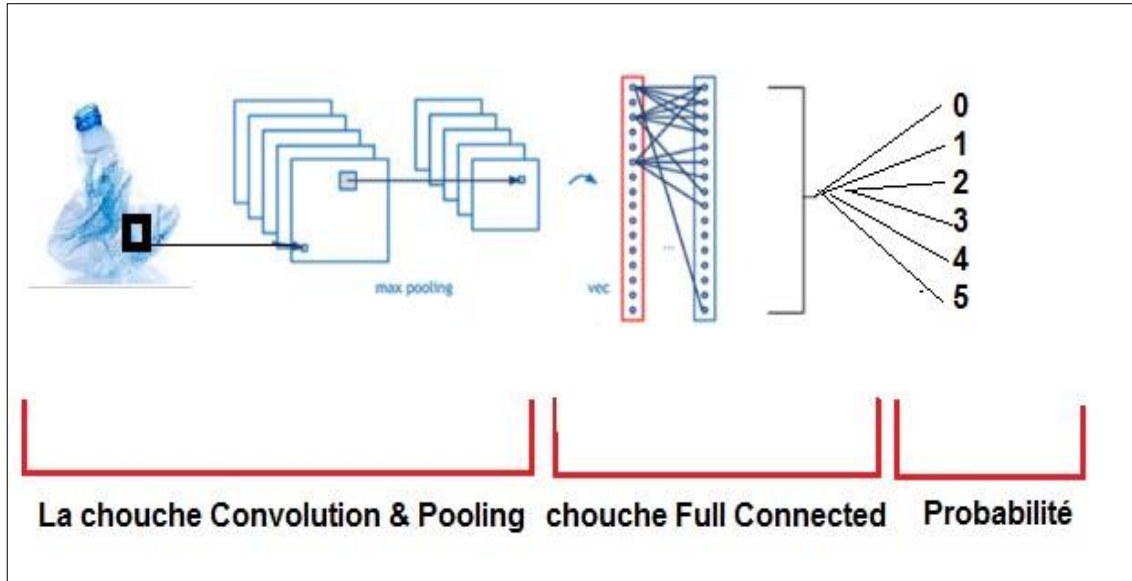


FIGURE 3.3. Les couches de CNN

La première couche c'est la couche de convolution ou Couche « CONV » : c'est là que CNN apprend, donc nous aurons certainement des matrices de poids. Pour calculer les paramètres apprenables ici, il suffit de multiplier les par la forme de largeur  $m$ , hauteur  $n$ , les filtres  $d$  de la couche précédente et de tenir compte de tous ces filtres  $k$  dans la couche actuelle. N'oubliez pas le terme de biais pour chacun des filtres. Le nombre de paramètres dans une couche CONV serait :  $((m * n * d) + 1) * k$ , ajouté 1 en raison du terme de biais pour chaque filtre. La même expression peut s'écrire comme suit :

Le nombre de paramètre = (la largeur du filtre \* la hauteur du filtre \* nombre de filtres dans la couche précédente + 1) \* nombre de filtres.

$$\text{Param} = ((3*3*3) + 1) * 32 = 896$$

Ensuite la couche POOL : Cela n'a pas de paramètres apprenables car il ne fait que calculer un nombre spécifique, aucun apprentissage impliqué, ainsi nombre de paramètres = 0. On applique Maxpooling pour réduire la taille de l'information ainsi la quantité de paramètres et de calcul.

La deuxième couche de convolution sera calculée avec la même formule de la première couche :

Le nombre de paramètre = (la largeur du filtre \* la hauteur du filtre \* nombre de filtres dans la couche précédente + 1) \* nombre de filtres.

$$\mathbf{Param = ((3*3*32) + 1) * 64 = 18496}$$

La quatrième et le sixième sont des couches de maxpooling avec le nombre de paramètres = 0, la cinquième couche c'est la dernière couche de convolution, leur paramètre calculé comme suit :

Le nombre de paramètre = (la largeur du filtre \* la hauteur du filtre \* nombre de filtres dans la couche précédente + 1) \* nombre de filtres.

$$\mathbf{Param = ((3*3*64) + 1) * 32 = 18464}$$

La huitième couche c'est une couche de sortie, cela a certainement des paramètres apprenables, en fait, par rapport aux autres couches, cette catégorie de couches a le plus grand nombre de paramètres, car, chaque neurone est connecté à tous les autres neurones, Alors, c'est le produit du nombre de neurones dans la couche actuelle c et du nombre de neurones sur la couche p précédente et comme toujours, n'oubliez pas le terme de biais. Ainsi, le nombre de paramètres

ici sont :

Le nombre de paramètre = ((couche actuelle c \* couche précédente p)  
+ 1 \* c)

$$\mathbf{Param = ((64*512) + 1*64) = 32832}$$

La deuxième couche aussi est une couche de sortie :

Le nombre de paramètre = ((couche actuelle c \* couche précédente p)  
+ 1 \* c)

$$\mathbf{Param = ((32*64) + 1*32) = 2080}$$

La dernière couche utilisée la fonction d'activation un softmax qui permet de calculer la distribution de probabilité des 6 classes.

Le nombre de paramètre = ((couche actuelle c \* couche précédente p)  
+ 1 \* c)

$$\mathbf{Param = ((6*32) + 1*6) = 198}$$

Totale paramètre =  $\Sigma Param$ .

$$\mathbf{Total param = \Sigma (896+18496+18464+32832+2080+198) = 72966}$$

```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 32, 32, 32)         896
max_pooling2d (MaxPooling2D) (None, 16, 16, 32)         0
conv2d_1 (Conv2D)            (None, 16, 16, 64)         18496
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 64)           0
conv2d_2 (Conv2D)            (None, 8, 8, 32)           18464
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 32)           0
flatten (Flatten)            (None, 512)                 0
dense (Dense)                 (None, 64)                  32832
dropout (Dropout)            (None, 64)                  0
dense_1 (Dense)               (None, 32)                  2080
dropout_1 (Dropout)          (None, 32)                  0
dense_2 (Dense)               (None, 6)                   198
-----
Total params: 72,966
Trainable params: 72,966
Non-trainable params: 0
    
```

FIGURE 3.4. Architecture première modèle CNN

## 1.5.2. Modèle 2

Dans ce modèle on utilise le même dataset du modèle précédent, on a divisé les images en dossiers d'images de train, de validation et de test avec une répartition 50 – 25 – 25.



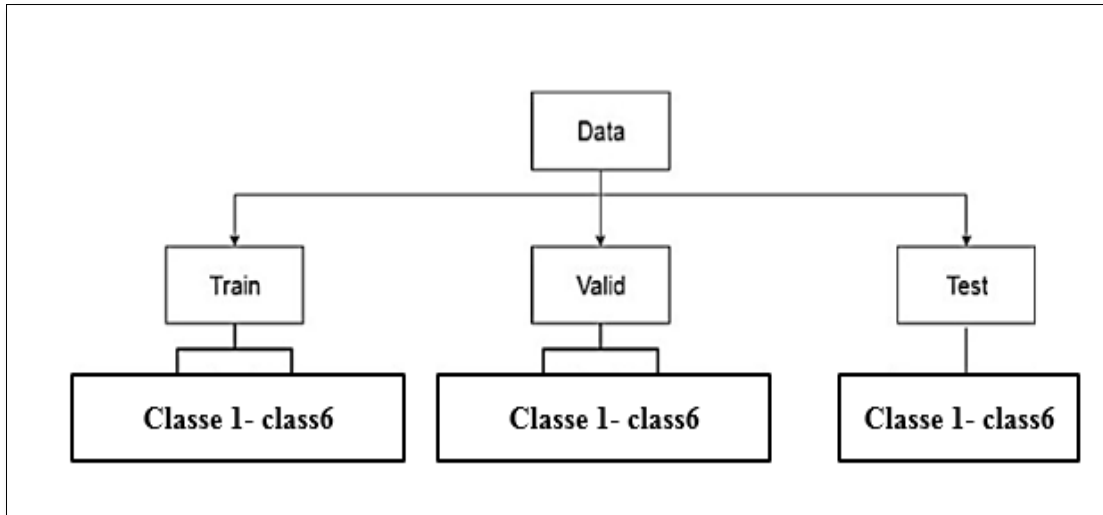


FIGURE 3.5. Division des données

### ❖ Resnet 34

Pour le training de modèle nous avons choisi le resnet 34 . Un réseau neuronal résiduel est un réseau neuronal convolutif (CNN) avec de nombreuses couches. En particulier, resnet34 est un CNN avec 34 couches qui ont été pré-entraînées sur la base de données ImageNet. Un CNN préformé fonctionnera mieux sur les nouvelles tâches de classification d'images car il a déjà appris certaines fonctionnalités visuelles et peut transférer ces connaissances (d'où un transfert d'apprentissage).

Puisqu'ils sont capables de décrire plus de complexité, les réseaux de neurones profonds devraient théoriquement mieux fonctionner que les réseaux peu profonds sur les données d'entraînement. En réalité, cependant, les réseaux de neurones profonds ont tendance à avoir des performances empiriques pires que les réseaux peu profonds.

Des réinitialisations ont été créées pour contourner ce problème à l'aide d'un hack appelé connexions de raccourci.

Lorsque des ajustements sont nécessaires, les connexions de raccourci appliquent la fonction d'identité pour transmettre des informations aux couches suivantes. Cela raccourcit le réseau neuronal lorsque cela est possible et permet aux réinitialisations d'avoir des architectures profondes et de se comporter davantage comme des réseaux neuronaux peu profonds. Le 34

dans resnet34 se réfère simplement au nombre de couches.

L'architecture de notre deuxième modèle est la suivante :

```
Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (4): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
      (1): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
      (2): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
)
```

```

(5): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (2): BasicBlock(
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (3): BasicBlock(
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
...
(6): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (2): BasicBlock(
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (3): BasicBlock(
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)

```

```

(4): BasicBlock(
  (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace)
  (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(5): BasicBlock(
  (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace)
  (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(7): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (2): BasicBlock(
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(1): Sequential(
  (0): AdaptiveConcatPool2d(
    (ap): AdaptiveAvgPool2d(output_size=1)
    (mp): AdaptiveMaxPool2d(output_size=1)
  )
  (1): Flatten()
  (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): Dropout(p=0.25)
  (4): Linear(in_features=1024, out_features=512, bias=True)
  (5): ReLU(inplace)
  (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (7): Dropout(p=0.5)
  (8): Linear(in_features=512, out_features=6, bias=True)
)
)

```

FIGURE 3.6. Architecture du deuxième modèle resnet

### ❖ Learning rate

Il faut trouver un taux d'apprentissage ( Learning rate ) pour la descente de gradient pour assurer que notre réseau neuronal converge assez rapidement sans manquer l'erreur optimale. On applique la fonction suivante :

```
model = create_cnn(data,models.resnet34,metrics=error_rate)
model.lr_find(start_lr=1e-6,end_lr=1e1)
model.recorder.plot()
```

Le chercheur de taux d'apprentissage suggère un taux d'apprentissage de  $5,13 \times 10^{-3}$ . Avec cela, nous pouvons former le modèle.

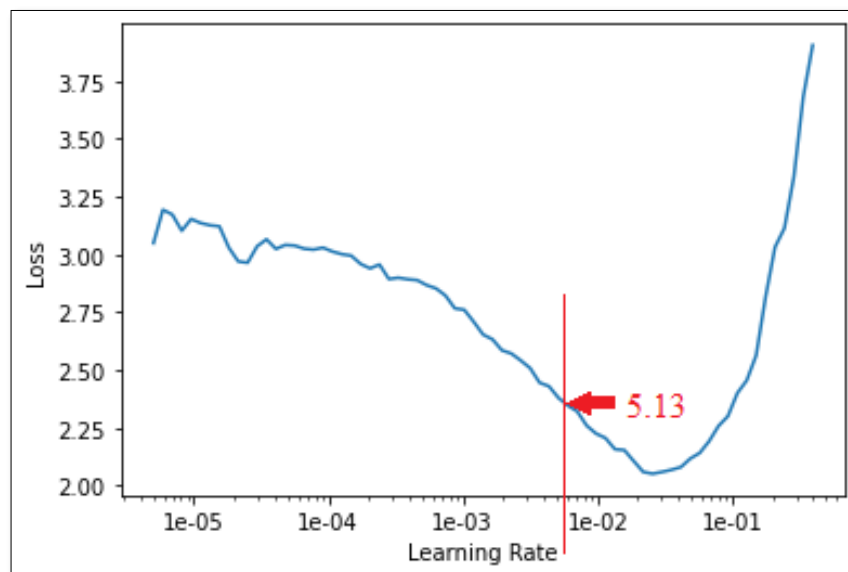


FIGURE 3.7 La valeur de Learning rate

### 1.5.3. Modèle 3

Ce modèle est différent aux deux premiers modèles, dans cette conception nous visons à atteindre le but de détecter et de classer des multiples objets situés dans une seule image, à cette raison nous proposons un modèle pour la détection d'objets recyclables à l'aide de MaskRCNN.

#### ❖ Mask RCNN

Notre choix tourne autour de Mask RCNN pour accomplir notre tâche. Mask RCNN est une extension de Faster RCNN qui identifie les régions d'intérêt et détermine les cadres de délimitation et les classifications de ces objets candidats. Mask RCNN étend cette sortie pour inclure également un masque pour chaque région d'intérêt. Ce masque nous permet d'effectuer une segmentation précise des objets en plus de la délimitation et de la classification.

Plus en détail, le modèle Mask RCNN est composé de quatre parties : « a feature extractor », « the region proposal network », « the classifier & bounding box regressor », et « the segmentation masks ». Le RCNN d'origine sur lequel ce modèle est basé à l'extracteur de caractéristiques comme hyper paramètre du modèle, et Mask RCNN ne fait pas exception. Dans le cas de Mask RCNN, l'extracteur de fonctionnalités est Resnet-101. Cet extracteur de caractéristiques à la pointe de la technologie exploite les fonctions résiduelles d'apprentissage qui font référence aux entrées de couche.

L'extracteur de caractéristiques de Mask RCNN comprend également un réseau de pyramides de caractéristiques (FPN) qui permet de détecter des objets à différentes échelles. Le FPN a une voie ascendante calcule différentes cartes d'entités avec une étape de mise à l'échelle de 2, et la voie descendante crée des entités de résolution plus élevée en sur échantillonnant des cartes d'entités plus grossières. Les deux chemins sont connectés latéralement pour fusionner les cartes de caractéristiques à la même échelle créée par les deux [6].

Une fois les caractéristiques extraites, le réseau de proposition de région (RPN), introduit dans Faster RCNN [7], scanne les boîtes d'ancrage dans l'image (plus précisément la carte des

caractéristiques de l'image) et propose soit un objet de premier plan, soit un arrière-plan pour chaque ancre boîte. Si un objet de premier plan est proposé, le RPN crée également un cadre de délimitation pour l'objet.

Un nombre réduit de propositions probables est obtenu en choisissant uniquement les propositions d'objet de premier plan avec la probabilité la plus élevée et en utilisant la suppression non max pour réduire les propositions proches les unes des autres.

Le masque RCNN se développe principalement sur le RCNN plus rapide après le RPN. Le masque RCNN prédit toujours la classe et effectue une régression de boîte englobante, mais il produit également un masque de segmentation binaire en parallèle. Le prédicteur de classe peut choisir parmi le nombre de classes prédéfinies avec une classe d'arrière-plan qui rejettera automatiquement la proposition de région si elle est sélectionnée. Le raffinement de la boîte englobante permet un léger ajustement du décalage et de l'échelle de la boîte englobante pour plus précisément l'objet. Un ajustement supplémentaire de Faster RCNN est l'étape RoIAlign qui vise à réduire les désalignements entre les régions d'intérêt et les caractéristiques qui pourraient affecter la prédiction de masques précis au pixel.

La segmentation code la disposition spatiale des objets . Cela diffère de la classification et de la boîte englobante car le masque a des valeurs de pixels correspondantes sans le réduire à des représentations vectorielles. Alors que les masques sont généralement de petite taille, les masques de vérité terrain pendant le train sont réduits tandis que les prédictions de masque sont mises à l'échelle jusqu'à la boîte englobante. Ces considérations empêchent le modèle d'être aussi intense en calcul. La fonction de perte est une somme de la perte de classe, de la perte de boîte englobante et de la perte de masque.

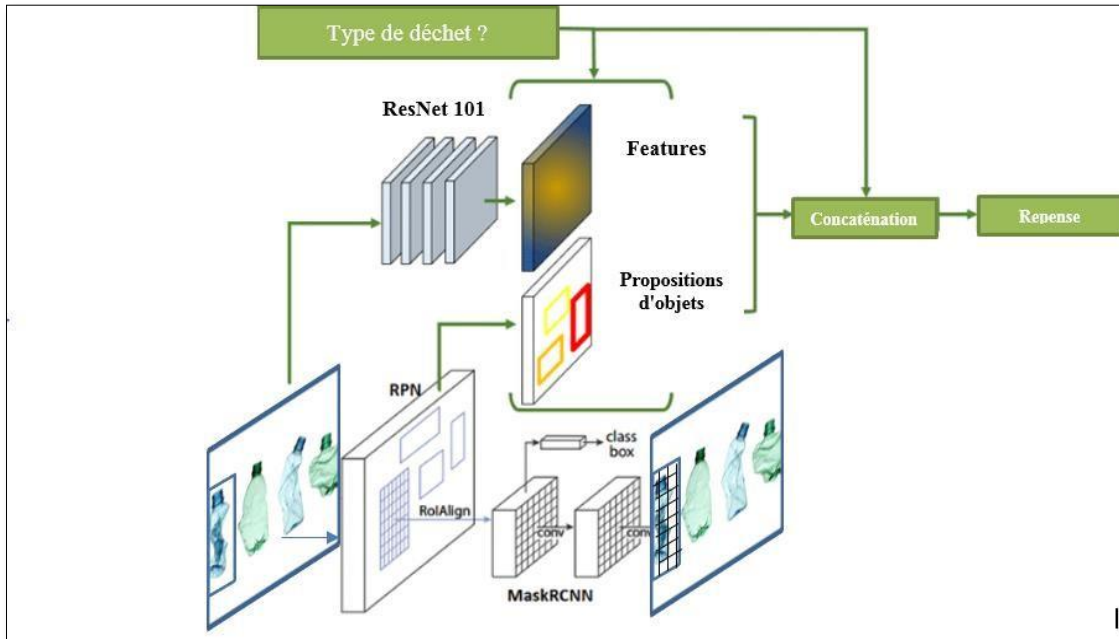


FIGURE 3.8. Architecture du troisième modèle Mask-RCNN

❖ **Apprentissage par transfère :**

L'idée de base de l'apprentissage par transfert est de prendre un modèle préformé et de lui fournir des données différentes pour accomplir une tâche autre que celle pour laquelle il a été initialement formé. Cela nous permet de tirer parti de la formation qui a déjà eu lieu et de réduire considérablement le temps que nous aurions eu pour former un modèle à partir de zéro . Dans notre cas, nous avons utilisé un modèle Mask-RCNN pré-entraîné sur le jeu de données COCO comme base pour notre modèle.

Nous avons tenté un autre type d'apprentissage par transfert pour ce modèle de « **réglage fin** » ou « **fin tuning** » . Dans ce cadre, nous avons formé toutes les couches du modèle existant avec les nouvelles données d'entraînement. Nous avons choisi ce style d'apprentissage par transfert afin de comparer et contraster les performances.



## 1.6. Conclusion

Nous avons présenté dans ce chapitre les architectures de nos modèles de classification des déchets en temps réel, en premier temps, nous avons détaillé notre première contribution de classification des images contenant un seul objet avec un arrière-plan simple et blanc on utilise le CNN et le Resnet, ainsi, dans la deuxième contribution (modèle 3) nous avons détaillé notre proposition de classification des images multi objets on se base sur le MaskRCNN. Le chapitre suivant sera le dernier chapitre qui englobe notre implémentation, résultats, synthèse et discussion.



# Chapitre 4

***Implémentation & Discussion***

## 4.1. Introduction

Dans ce dernier chapitre nous avons effectué plusieurs évaluations pour la classification des déchets. Aussi nous avons étudié plusieurs cas et nous avons fait des comparaisons.

## 4.2. Les outils et librairies Utilisés


### 4.2.1. Environnement et technologies logicielles

#### a) Ressources matérielles :

La configuration du matériel utilisé dans notre implémentation est :


- ✓ Un PC portable : Acer i3-4005U CPU @1.70GHz
- ✓ RAM :4 GO
- ✓ Disque dur de taille 500 GO
- ✓ Système d'exploitation : Windows 10 64 bits

#### b) Ressources logicielles :


|   |   |
|---|---|
|  | <p>Vue les fortes exigences de calculs et espaces nécessaires pour l'implémentation des solutions à base de « TensorFlow » et « Keras », nous avons opté pour l'utilisation de calculs en Cloud via la plateforme « Google Colab » qui offre la possibilité de calcul GPU. C'est une plateforme gratuite. Elle utilise l'environnement « Jupyter Notebook » pour les différentes routines de programmation en langage Python.</p> |
|---|---|



|   |   |
|---|---|
|    | <p><b>PyCharm</b> est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains. Il fournit l'analyse de code, un débogueur graphique, un testeur d'unité intégré, l'intégration avec des systèmes de contrôle de version (VCS) et prend en charge le développement Web avec Django ainsi que la science des données avec Anaconda.</p>                          |
|   | <p><b>Qt version 5</b> (prononcé "cute") est une boîte à outils de widgets gratuite et open-source pour créer des interfaces utilisateur graphiques ainsi que des applications multiplateformes qui s'exécutent sur diverses plates-formes logicielles et matérielles telles que Linux, Windows, macOS, Android ou des systèmes embarqués avec peu ou aucun changement dans la base de code sous-jacente tout en étant une application native avec des capacités et une vitesse native.</p> |
|  | <p><b>LabelImg</b> est un outil d'annotation d'image graphique et d'étiquettes de délimitation d'objets dans les images.</p> <p>Il est écrit en Python et utilise Qt pour son interface graphique.</p> <p>Les annotations sont enregistrées sous forme de fichiers XML au format PASCAL VOC, le format utilisé par ImageNet.</p>  |


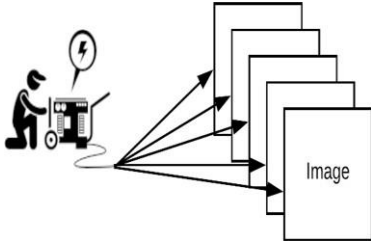
### 4.2.2. Langage utilisé :

|   |   |
|---|---|
|  The Python logo consists of two interlocking snakes, one blue and one yellow, forming a circular shape. | <p><b>Python</b> est le langage de programmation open source le plus employé par les informaticiens, il a libéré les développeurs de contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.</p> |
|---|---|

### 4.2.3. Les bibliographies utilisées

|   |   |
|---|---|
|  The OpenCV logo features three interlocking circles in red, green, and blue, with the text "OpenCV" below them. | <p><b>Open CV</b> (Open Source Computer Vision Library) est une bibliothèque de logiciels open source de vision informatique et d'apprentissage automatique. La bibliothèque contient plus de 2500 algorithmes optimisés, qui incluent un ensemble complet d'algorithmes de vision et d'apprentissage automatique. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer les actions humaines dans des vidéos.</p> |
|---|---|

|  |   |
|--|---|
|   | <p>Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, Elle a été développée pour permettre une expérimentation rapide.</p> <p><b>Avantages :</b></p> <ul style="list-style-type: none"> <li>+ Le backend par excellence pour Theano ou TensorFlow</li> <li>+ Interface haut niveau, intuitive</li> </ul> <p><b>Inconvénients</b></p> <ul style="list-style-type: none"> <li>+ Moins flexible que les autres API</li> </ul>  |
|  | <p><b>TensorFlow</b> est une bibliothèque de logiciels gratuits et open-source pour le flux de données et la programmation différentiable à travers une gamme de tâches. C'est une bibliothèque mathématique symbolique, et est également utilisée pour des applications d'apprentissage automatique telles que les réseaux de neurones</p> <p><b>Avantages</b></p> <ul style="list-style-type: none"> <li>+ Supporté par Google</li> <li>+ Une très grande communauté</li> <li>+ Le support du multi-GPU</li> </ul> <p><b>Inconvénients</b></p> <ul style="list-style-type: none"> <li>+ Plus lent que les autres framework dans de nombreux benchmarks, bien que Tensorflow se rattrape.</li> </ul> |

|  |   |
|--|---|
|   | <p><b>SKLearn Scikit-learn</b> (anciennement scikits.learn) est une bibliothèque d'apprentissage automatique de logiciels libres pour le langage de programmation Python. Il présente divers algorithmes de classification, de régression et de regroupement, notamment des machines à vecteurs de support, des forêts aléatoires, l'amélioration du gradient, k-means et DBSCAN, et est conçu pour interopérer avec les bibliothèques numériques et scientifiques Python NumPy et SciPy.</p> |
|  | <p><b>ImageDataGenerator</b> : Cette fonction génère des lots de données d'image de tenseurs avec une augmentation des données en temps réel. Les données seront bouclées (par lots). Keras possède cette classe ImageDataGenerator qui permet aux utilisateurs d'agrandir l'image à la volée très facilement.</p> <p>Utilisé si votre statut d'utilisation n'est pas très courant et que vous ne disposez que de quelques photos dans votre jeu de données d'entraînement</p>                |

### 4.3. Datasets

Nous avons basé notre travail sur la dataset « **TrashNet** » utilisée dans les travaux connexes du chapitre précédent afin de comparer notre modèle avec leur modèle, en plus nous avons utilisé une autre dataset pour atteindre le but de classification multi objets.



### 4.3.1. Premier dataset

Trash Net [33] c'est le premier ensemble de données utilisée. Les photos ont été prises en plaçant l'objet sur un tableau blanc et en utilisant la lumière du soleil et / ou l'éclairage de la pièce. Les images ont été redimensionnées à 512 x 384, ce qui peut être modifié dans `data / constants.py` (leur redimensionnement implique de passer par l'étape 1 en cours d'utilisation). Les appareils utilisés sont Apple iPhone 7 Plus, Apple iPhone 5S et Apple iPhone SE.

La taille de l'ensemble de données d'origine dépasse environ 3,5 Go. L'ensemble de données comprend six catégories : **verre, papier, carton, plastique, métal et déchets**. L'ensemble de données comprend actuellement 2527 photos :

| <b>Catégorie</b> | <b>Nb d'image</b> |
|------------------|-------------------|
| Verres           | 501               |
| Papiers          | 594               |
| Cartons          | 403               |
| Plastique        | 482               |
| Minéraux         | 410               |
| Poubelle         | 137               |
| <b>Total</b>     | <b>2527</b>       |

TABLEAU 4.1. Les différentes catégories du TrashNet

Un échantillon de mélange de différentes catégories de cette dataset est présenté dans la figure suivante :



FIGURE 4.1. Echantillon de données de TrashNet [33]

**Nous remarquons à partir de la photo précédente que tous les image en la même taille (512\*512), c'est pour cela nous avons ignoré l'étape de prétraitement de données.**

### 4.3.2. Deuxième Dataset

Comme le jeu de données précédent ne contenait pas d'images avec beaucoup d'objets, nous avons dû rechercher un nouvel ensemble de données pour atteindre l'objectif de classer une image contenant plusieurs types de déchets. Le jeu de données que nous avons utilisé remonte à [37]. Il s'agit d'un ensemble de données composé de bouteilles en plastique segmentées, de bouteilles en verre et de canettes de soda métalliques avec des étiquettes, créé à partir d'une combinaison de photos prises par l'auteur lui-même en plus de photos trouvées sur Internet. Ce sont des images avec des propriétés visuelles intéressantes. Ces propriétés comprenaient des blocages (d'autres objets qui obscurcissent partiellement l'objet que nous essayons de détecter), des images floues, des images de bouteilles / canettes, pour n'en nommer que quelques-uns.

Cette dataset contient les données diviser en données de traitement, de validation et de test utilisées dans notre projet

L'ensemble de données comprend 3 catégories : **verre, plastique, métal**. L'ensemble de données comprend actuellement 740 photos :

| Catégorie    | Nb d'image |
|--------------|------------|
| Verres       | 157        |
| Plastique    | 151        |
| Minéraux     | 215        |
| <b>Total</b> | <b>523</b> |

TABLEAU 4.2. Les différentes catégories du deuxième dataset

Un échantillon de mélange de différentes catégories de cette dataset est présenté dans la figure suivante :



FIGURE 4.2. Echantillon du deuxième data set [37]

### 4.3.3. Etiqueter les image

Cette étape consiste à annoter l'ensemble de données afin de définir l'emplacement (cadre de sélection) des objets dans chaque image. Nous les avons étiquetés manuellement avec LabelImg et générés des fichiers XML pour chaque image, c'est vraiment une tâche fastidieuse.

Image suivant montre la première interface de labelImg ,c'est a partir de cette image que on choisi l'ensembles de donnes .

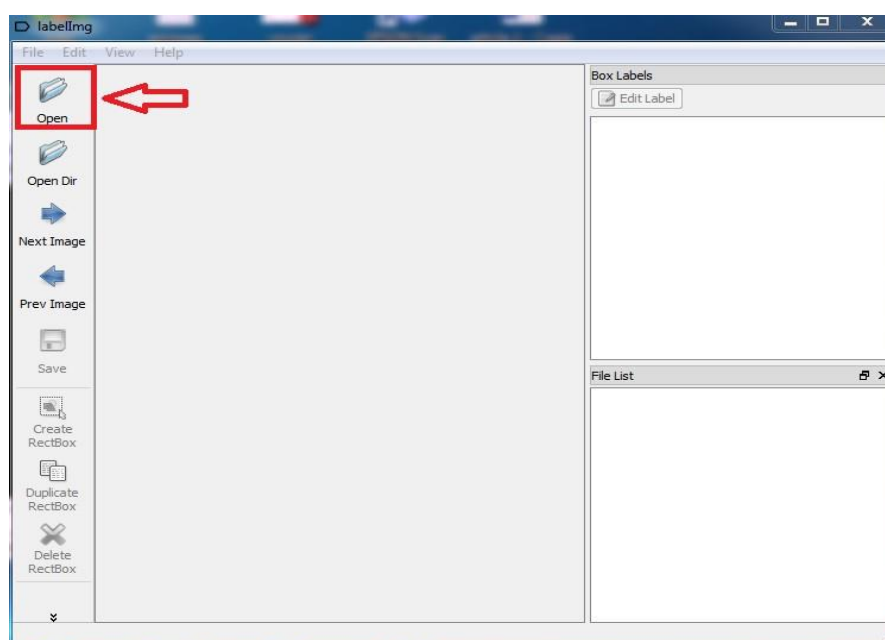


FIGURE 4.3. Choix de fichier dans labelImg

Après le choix des images en clique sur la Button « Create box » est en désigne le box sur l'objet à classer, ensuite on donne un nom au ce box est en clique sur « ok ».

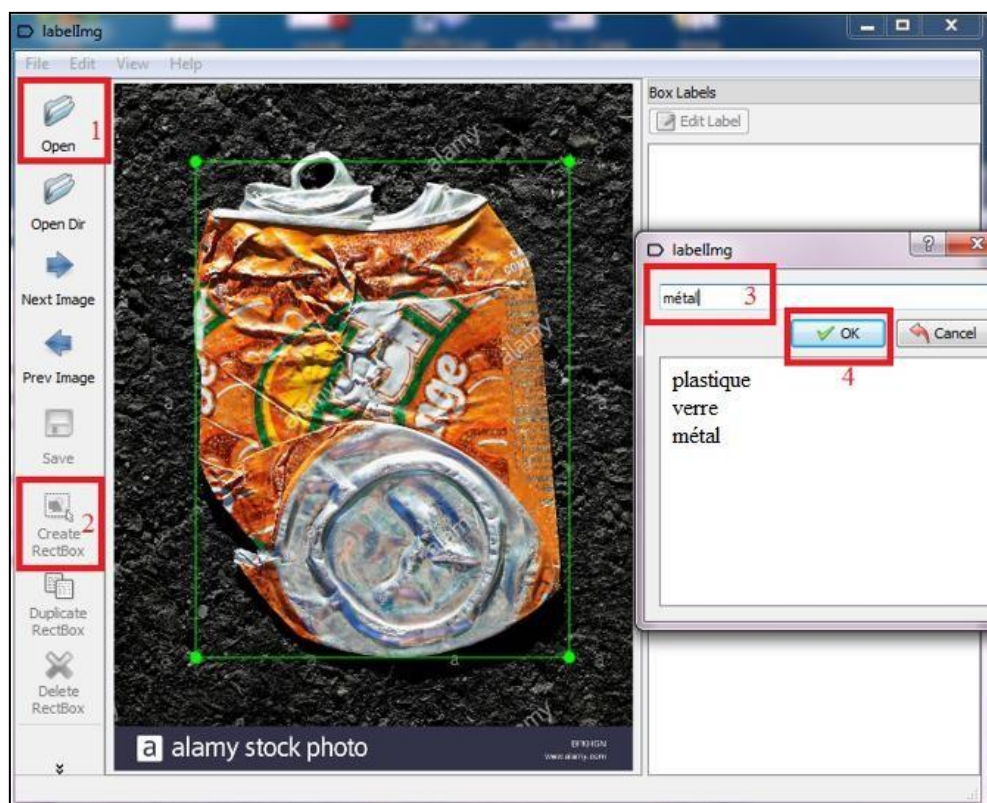


FIGURE 4.4. création de box dans labelImg

## 4.4. Processus des modèles

Un réseau de neurones à convolution (ConvNet ou CNN) est l'un des algorithmes les plus populaires pour l'apprentissage en profondeur, un type d'apprentissage automatique où un modèle apprend à effectuer la tâche de classification directement à partir de texte, d'images, de vidéos ou de sons. Les CNN sont particulièrement utiles pour trouver les motifs dans les images pour détecter et reconnaître des objets, des visages et des poses.

### 4.4.1. Étapes impliquées

- **Étape 1** : démarrer

- **Étape 2** : le système est sous tension. L'OpenCV apparaît à l'écran. L'image des déchets est affichée devant.
- **Étape 3** : Les données de training sont utilisées pour analyser les images. Seuls les points marqués de l'image de test sont analysés avec les données d'entraînement.
- **Étape 4** : Détection et localisation d'objets par un algorithme de délimitation qui découvre les caractéristiques des images qui pourraient facilement identifier l'image.
- **Étape 5** : L'identification de l'objet a lieu. L'image est identifiée à l'aide de l'algorithme CNN.
- **Étape 6** : Prédiction : La prédiction du type de déchets est basée sur l'identification et un indice de probabilité étudié.
- **Étape 7** : Arrivée à la sortie : La prédiction avec la plus forte probabilité est considérée comme la sortie.
- **Étape 8** : Classification de l'objet : L'objet est ensuite classé selon le type de déchets auquel il appartient.
- **Étape 9** : Arrêter.

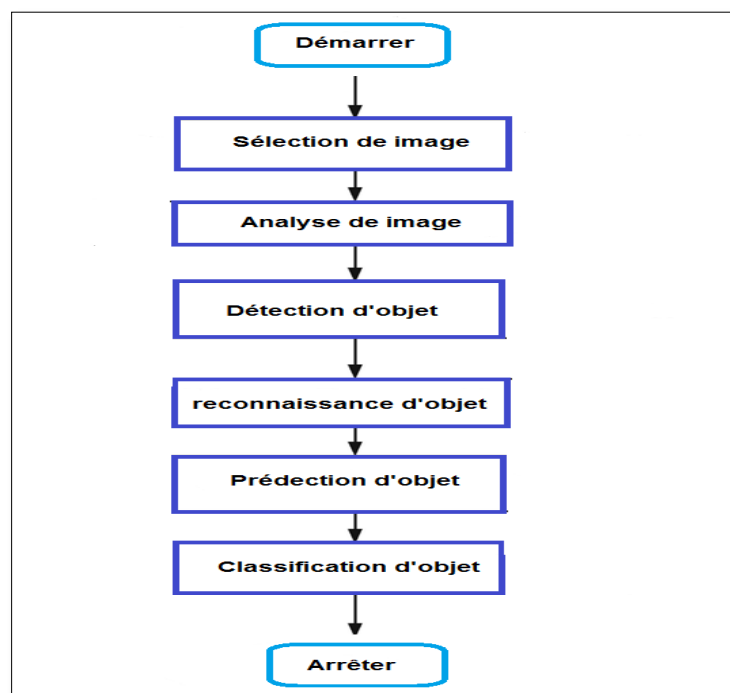


FIGURE 4.5. Le processus proposé

## 4.5. Expérimentation des résultats

### 4.5.1. Model 1

#### ❖ **Trainement**

Le trainement est la première étape à pratiquer après la collecte de l'ensemble de données. Un grand nombre d'entrées doivent être fournies pour former le réseau. Ici, dans le but de reconnaître les images, formez une couche d'entités qui reçoivent une entrée des pixels de l'image. Après 200 époques de trainement, La précision de l'apprentissage augmente avec le nombre d'époque, de même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époque avec un overfitting à partir de l'époque numéro 125 .

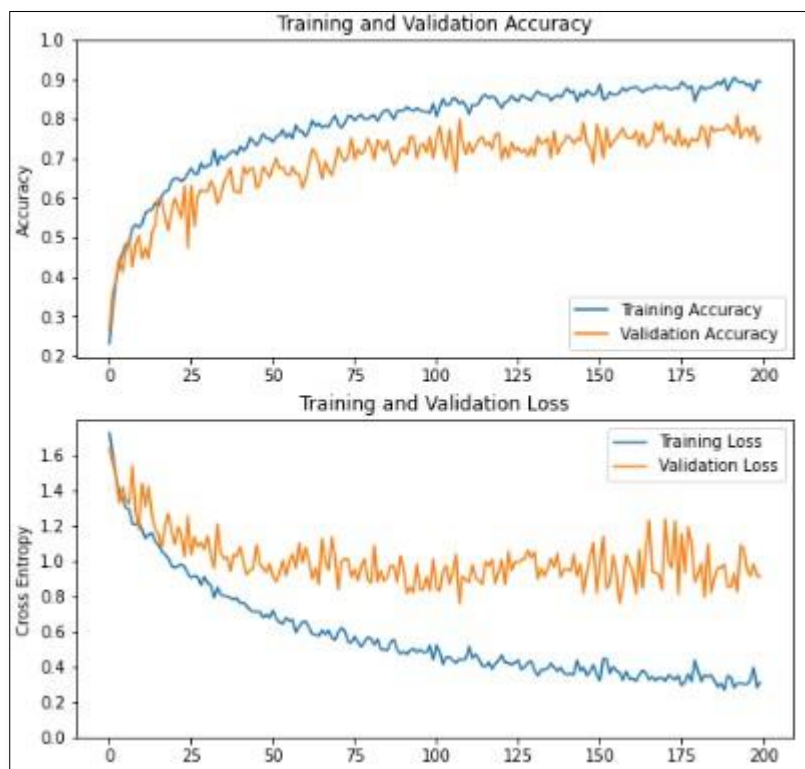


FIGURE 4.6. Précision et Erreur pour le modèle 01

## ❖ Résultat

La matrice de confusion de test suivante, montre que notre premier modèle a mal classé toutes les classes sauf le papier.

|           | Carton | Verre | Métal | Papier | Plastique | Poubelle |
|-----------|--------|-------|-------|--------|-----------|----------|
| Carton    | 2      | 9     | 7     | 17     | 2         | 3        |
| Verre     | 4      | 10    | 15    | 15     | 2         | 4        |
| Métal     | 2      | 3     | 12    | 17     | 4         | 3        |
| Papier    | 6      | 5     | 19    | 24     | 3         | 2        |
| Plastique | 5      | 4     | 9     | 22     | 4         | 4        |
| Poubelle  | 2      | 3     | 5     | 3      | 0         | 0        |

TABLEAU 4.3. Matrice de confusion du modèle 01



### ❖ Exemple

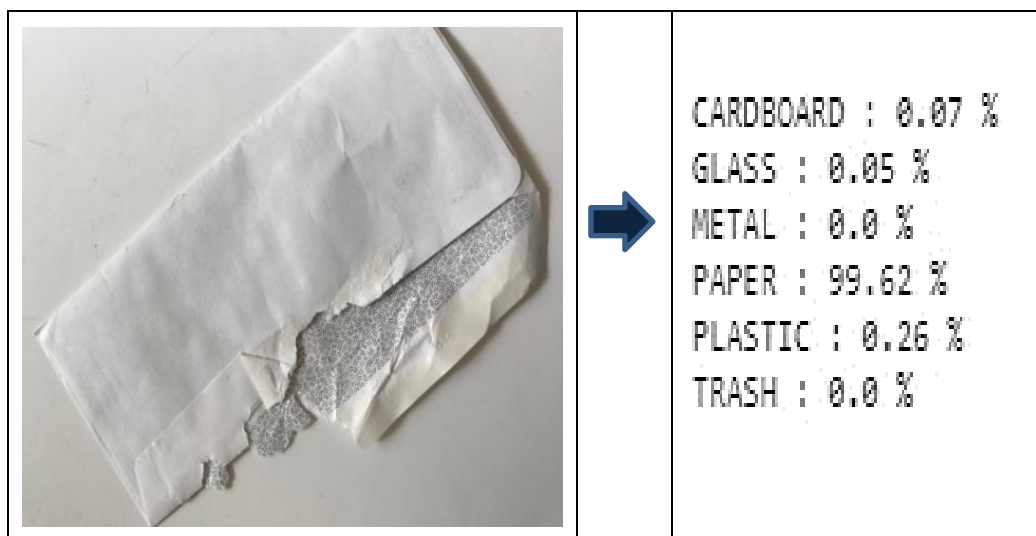


FIGURE 4.7. Exemple de test pour le modèle 01

## 4.5.2. Model 2

### ❖ Training

Après 20 époques de training, le taux d'apprentissage diminue à chaque époque, ce qui nous permet de nous rapprocher de plus en plus de l'optimum.

| Epoch | Train_loss | Valid_loss | Error_rate |
|-------|------------|------------|------------|
| 1     | 1.42       | 0.76       | 0.24       |
| 2     | 0.88       | 0.51       | 0.17       |
| 3     | 0.70       | 0.39       | 0.13       |
| 4     | 0.63       | 0.61       | 0.20       |
| 5     | 0.65       | 0.58       | 0.20       |
| 6     | 0.65       | 0.52       | 0.16       |

|           |      |      |      |
|-----------|------|------|------|
| <b>7</b>  | 0.60 | 0.49 | 0.19 |
| <b>8</b>  | 0.56 | 0.47 | 0.16 |
| <b>9</b>  | 0.56 | 0.36 | 0.12 |
| <b>10</b> | 0.46 | 0.40 | 0.13 |
| <b>11</b> | 0.38 | 0.36 | 0.11 |
| <b>12</b> | 0.37 | 0.37 | 0.12 |
| <b>13</b> | 0.34 | 0.32 | 0.10 |
| <b>14</b> | 0.33 | 0.25 | 0.08 |
| <b>15</b> | 0.28 | 0.24 | 0.09 |
| <b>16</b> | 0.20 | 0.23 | 0.08 |
| <b>17</b> | 0.18 | 0.22 | 0.08 |
| <b>18</b> | 0.20 | 0.22 | 0.08 |
| <b>19</b> | 0.19 | 0.20 | 0.07 |
| <b>20</b> | 0.19 | 0.22 | 0.08 |

TABLEAU 4.4. Entraînement du modèle 02

## ❖ Prédiction de test\_dataset

Pour prédire des nouvelles données on applique la fonction suivante :

```

▶ preds = learn.get_preds(ds_type=DatasetType.Test)
    
```

La matrice de confusion suivante présente la performance de modèle sur les données de test :

|                  | <b>Carton</b> | <b>Verre</b> | <b>Métal</b> | <b>Papier</b> | <b>Plastique</b> | <b>Poubelle</b> |
|------------------|---------------|--------------|--------------|---------------|------------------|-----------------|
| <b>Carton</b>    | 91            | 1            | 1            | 7             | 0                | 1               |
| <b>Verre</b>     | 0             | 118          | 5            | 0             | 3                | 0               |
| <b>Métal</b>     | 1             | 6            | 94           | 0             | 0                | 2               |
| <b>Papier</b>    | 1             | 0            | 1            | 145           | 0                | 2               |
| <b>Plastique</b> | 0             | 7            | 1            | 1             | 109              | 3               |

|                 |   |   |   |   |   |           |
|-----------------|---|---|---|---|---|-----------|
| <b>Poubelle</b> | 0 | 2 | 1 | 2 | 2 | <b>28</b> |
|-----------------|---|---|---|---|---|-----------|

TABLEAU 4.5. Matrice de confusion du modèle 02

On remarque que le modèle a bien classé toutes les classes.

### 4.5.3. Modèle 3

Nous avons formé un modèle Mask RCNN avec une classe principale de modèle : qui est été formés par le biais de fine tuning. Comme expliqué dans le chapitre précédemment, cela signifie que la moitié des modèles ont formé chaque couche dans chaque partie du modèle (réseau de proposition de région, réseau de pyramides d'entités, ResNet). De plus, nous avons formé ce model et un grand ensemble de formation égale à 100 images. En tant que contrôle, nous avons également inclus le modèle pré-entraîné sur le jeu de coco sans aucun apprentissage par transfert. Bien que le modèle ne classe pas les canettes métalliques, les bouteilles font partie des classes sur lesquelles il est formé

Malheureusement, en raison de la complexité de l'entraînement d'un modèle Mask RCNN, il n'est pas très utile de le faire fonctionner localement. Pour vraiment tester notre modèle, nous avons implémenter le modèle dans Google Colab, un interpréteur python en ligne (comme Jupyter) qui vous permet d'exécuter du code sur un GPU à 12 cœurs.

### ❖ **Trainement**

Pour former le modèle, nous avons utilisé 30 époques de 100 étapes chacune, où un lot d'images a été évalué et la perte calculée pour le lot à chaque étape. Nous avons choisi 30 époques car empiriquement, nous n'avons pas vu beaucoup d'amélioration dans les époques successives après 30. À chaque étape, la perte calculée est un composite de nombreuses catégories différentes de perte, y compris la perte de classe rpn, la perte de box rpn, la perte de classe mrcnn, mrcnn perte de bbox et perte de masque mrcnn. La perte de classe RPN et la perte

de boîte englobante est la perte associée aux boîtes englobantes prédites et à la prédiction de classe dans la boîte englobante par le réseau de proposition de région (RPN). La perte de la boîte englobante MRCNN, la perte de masque et la perte de classe sont les pertes associées aux boîtes englobantes, aux masques et aux prédictions de classe finaux associés à chaque boîte englobante.

## ❖ Résultat

Les résultats du calcul de la précision moyenne sur nos ensembles d'images de validation et de test sont contenus dans le tableau suivant. Le meilleur résultat de test à 0,92

Dans l'ensemble, cependant, les résultats de train est de test ont été très bons dans tous les classes. Le modèle a très bien fonctionné lors de la détection, en plaçant des cadres de délimitation serrés, en classifiant l'objet dans le cadre de délimitation et en masquant l'objet. Nous avons inclus de nombreux exemples des résultats du modèle dans la section suivante. Nous pensions initialement que le modèle aurait du mal à distinguer les bouteilles en plastique des bouteilles en verre en raison de la similitude de leur forme.

Par rapport au modèle pré-entraîné sur le coco uniquement, notre modèle est bien mieux performé. Ceci est principalement dû au fait que le modèle pré-entraîné ne classe que les bouteilles parmi les 80 classes sur lesquelles il est formé, il n'a donc pas pu identifier les canettes. Bien que nos ensembles de validation et de test soient uniformément répartis entre les trois classes, tout déséquilibre affecterait davantage les scores du modèle pré-entraîné que nos modèles.

| <b>Le model mask RCNN</b>       | <b>Train</b> | <b>Test</b> |
|---------------------------------|--------------|-------------|
| <b>Fine Tuning</b>              | 0.80         | 0.92        |
| <b>Le modèle Coco prétraité</b> | 0.21         | 0.22        |

TABLEAU 4.6 Résultat du modèle 03

### ❖ Exemples

L'image suivante englobe un exemple de test des différentes classes du modèle implémenter :

#### Image originale

#### Image avec la prediction

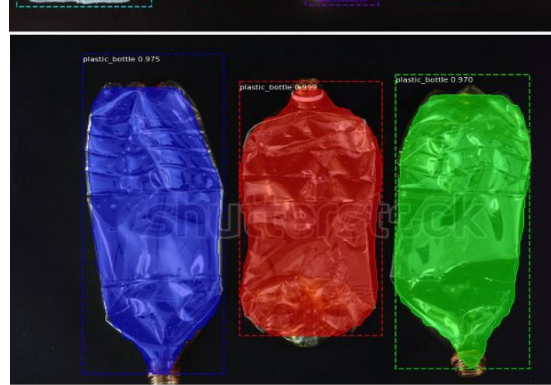




FIGURE 4.8. Exemple de test pour le modèle 03

## 4.6. Synthèse

### ❖ Contribution 1

Comme en déjà dite en a prend notre dataset à partir de TrashNet [28] de la classification des déchets en 6 catégories. Les résultats des travaux appliqués à cette dataset est de :



| N | Systèmes  | Taux de test | Références |
|---|-----------|--------------|------------|
| 1 | CNN 2D    | 22%          | [33]       |
| 2 | SVM       | 63%          | [33]       |
| 3 | Desnet    | 95%          | [34]       |
| 4 | CNN 2D    | 82%          | [35]       |
| 5 | Resnet    | 88,66%       | [36]       |
| 6 | CNN 2D    | 21%          | /          |
| 7 | Resnet 43 | 92 %         | /          |

TABLEAU 4.7 Tableau de Synthèse

Dans ce tableau en a les résultats de l'accuracy des différent travaux, le meilleur résultat est pour le système le Desnet [34] , ensuite notre système où en a obtenons 92% à partir de model de Resnet avec le nombre d'époques 20.

## ❖ Contribution 2

Notre précision de test est de 92 %, tandis que les créateurs originaux de l'ensemble de données ont atteint une précision de test de 63% avec SVM sur une division de 70/30.

### 4.7. Implémentation d'interface graphique

Dans ce stade d'interface graphique nous avons travaillé avec l'outil Qt version 5 sur la plateforme Pycharm.

#### ✓ Interface principale de l'application.

Cette interface englobe 5 fenêtres :

- **La fenêtre 1** : affiche l'image a classé
- **La fenêtre 2** : affiche la probabilité
- **La fenêtre 3** : affiche le label de l'image
- **La fenêtre 4** : affiche le plot de précision et de l'erreur
- **La fenêtre 5** : affiche le processus de l'enrênement

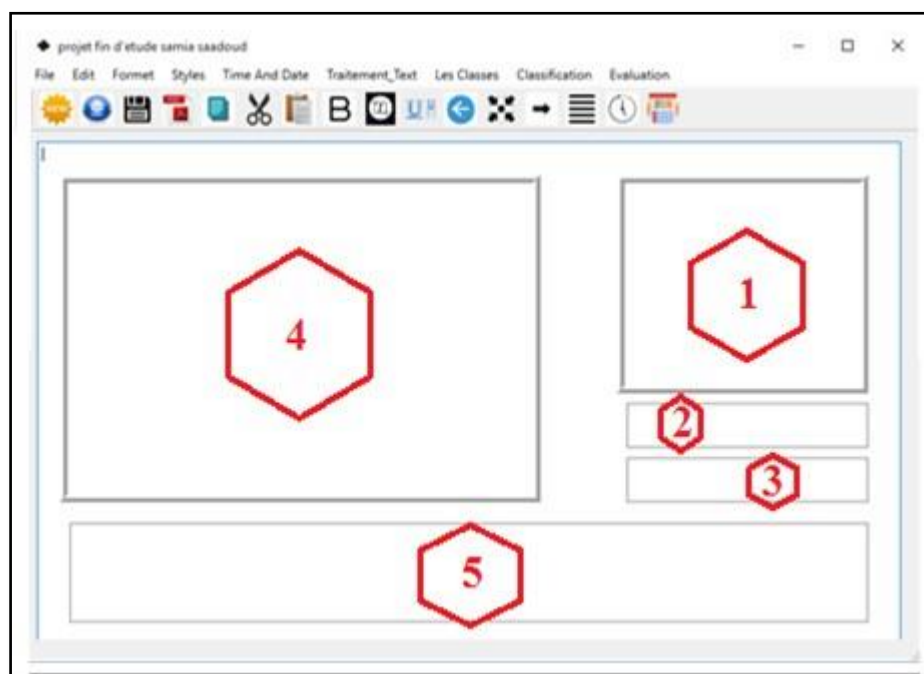


FIGURE 4.9. Interface principale de l'application



### ✓ Fonctionnement de l'application

- Cliquer sur le bouton "select image " à partir de « classification » pour ouvrir une boîte de dialogue pour choisir l'image à partir d'un emplacement local

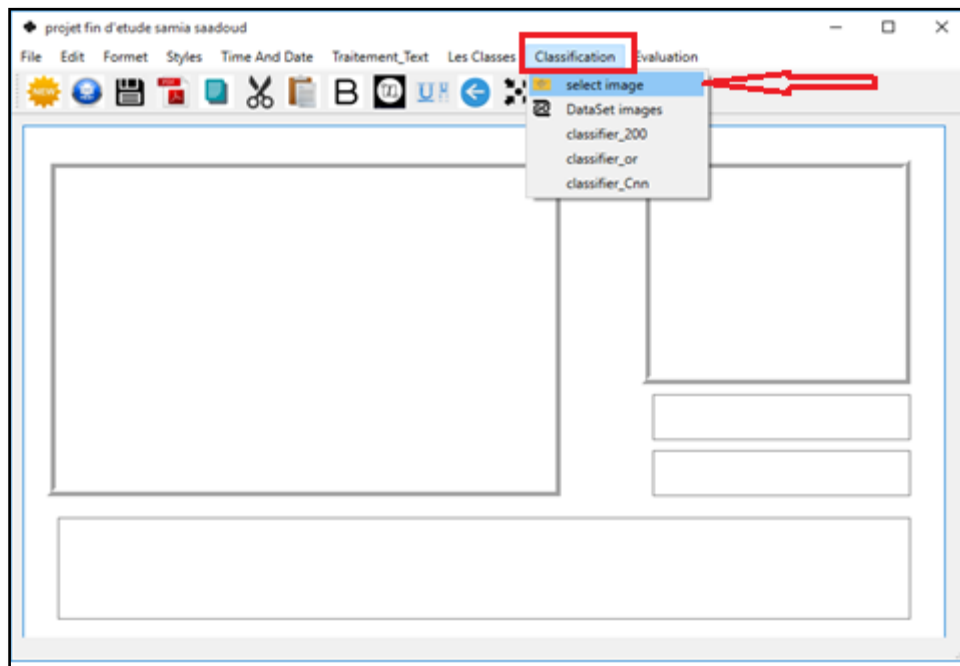


FIGURE 4.10. Sélection d'image

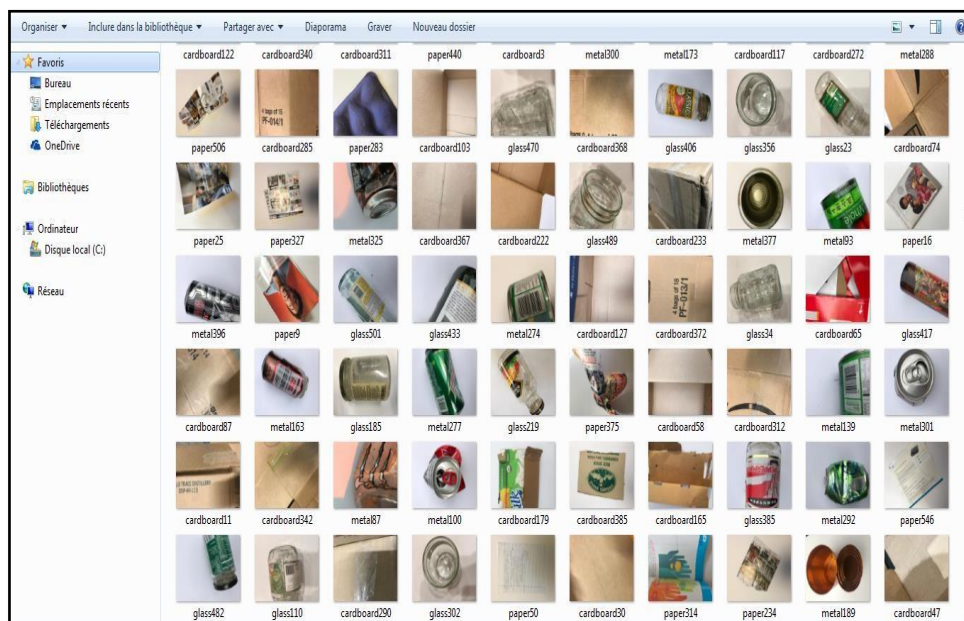


FIGURE 4.11. Le fichier des images

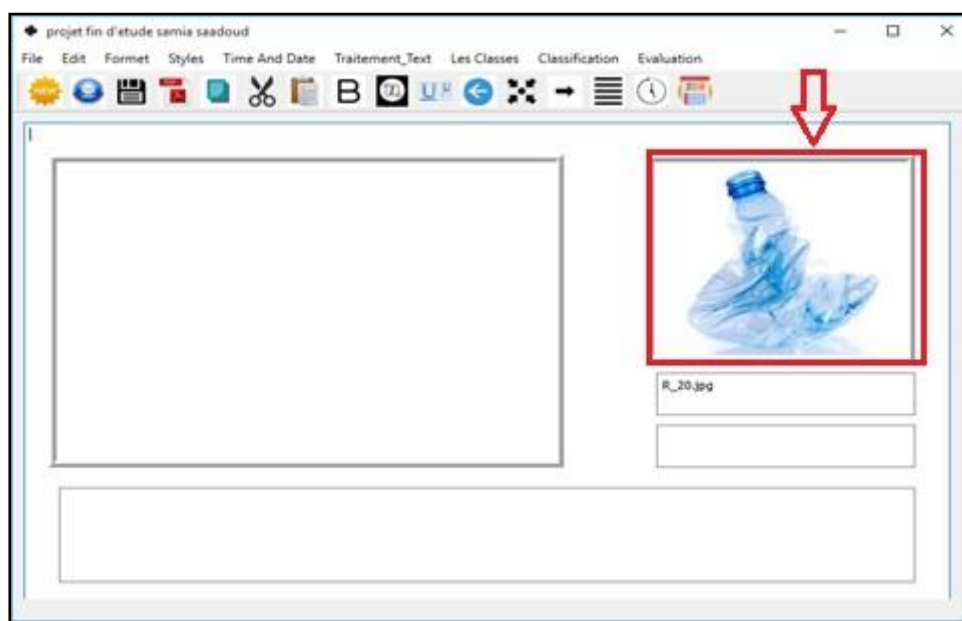


FIGURE 4.12. Affichage de choix

Une fois l'image sélectionner, ont choisi le type de classifier a partie des modèles déjà implémenté, le résultat sera comme suit :

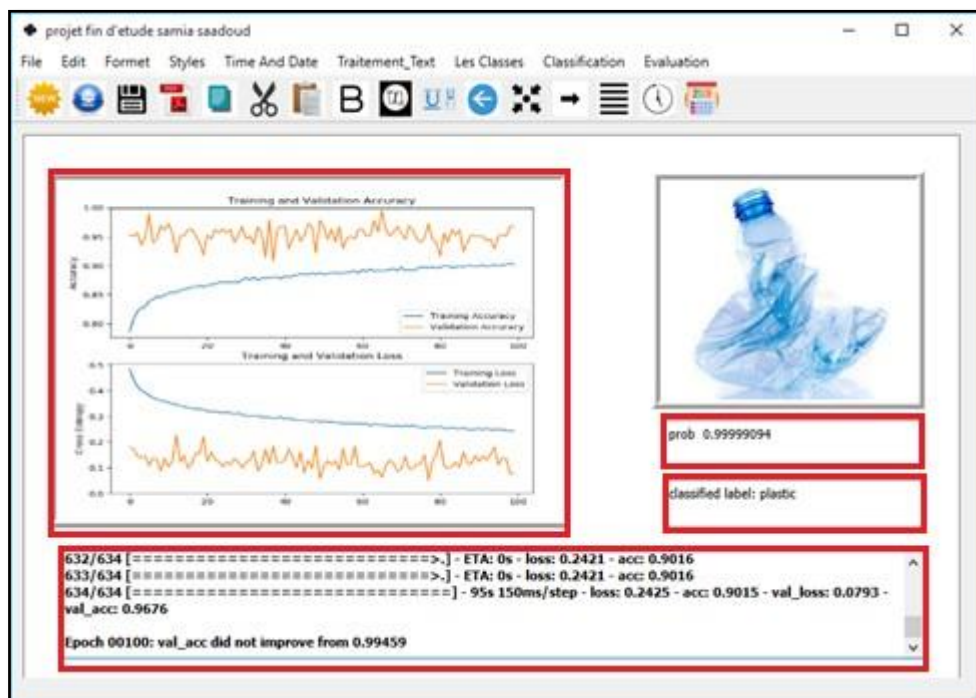


FIGURE 4.13. Résultat final

## 4.8. Conclusion

Nous avons présenté dans ce chapitre une approche de classification basée sur les réseaux de neurones, pour cela on a utilisé trois modèles avec différents datasets et on a montré les différents résultats. La comparaison des résultats trouvés a montré que le deuxième modèle avec le premier dataset est donné de meilleurs résultats, aussi avec le deuxième dataset nous avons atteint le but de détection de multiples objets dans une seule image avec un pourcentage satisfaisable.

Conclusion

Générale

# 1. Conclusion

Ce travail de recherche s'inscrit dans le cadre d'une contribution basée sur les algorithmes de l'intelligence artificielle pour la planification urbaine. La gestion des déchets est l'un des axes de recherche de planification urbaine, elles constituent un grand défi auquel fait face toutes les sociétés actuelles où leur traitement reste une problématique et se traduit par des multiplications négatives

Pour faire face à cette problématique, j'ai adopté une méthodologie de recherche qui consiste à concevoir et implémenter un système de classification automatique d'images dont une perspective de gestion des déchets de manière automatique.

A ce stade nous avons choisi d'appliquer les algorithmes d'apprentissage profond pour classifier les déchets en 6 catégories, aussi une autre contribution a lieu dans cette mémoire, c'est la classification de plusieurs objets dans une image en utilisant un deuxième dataset pour classifier les déchets en 3 types. Pour cela nous avons choisi de réaliser une application intelligente sous le langage de programmation Python qui utilise une source de données contenant des images annotées par des valeurs (Papier, Verre, Plastique, Métal, Carton, Poubelle) et les valeurs (Verre, Métal et Plastique) pour le deuxième dataset.

Nous avons commencé par l'introduction de contexte de planification urbaine puis la gestion des déchets et la détection de notre problématique. Ensuite, nous avons réalisé un état de l'art où nous avons mis l'accent sur les algorithmes d'apprentissage automatique et leur application aux gestion des déchets (approches connexes), puis nous avons proposé notre contribution où nous avons étudié deux datasets et fait une comparaison. Les classificateurs que nous avons appliqués sont le CNN-2D, le Resnet-34 et le Mask-RCNN, ce choix est justifié par la simplicité et l'efficacité de ces méthodes. Les résultats obtenus sont très encourageants, où nous avons atteint un meilleur Taux celui de 92% dans la première contribution avec le Resnet-34, et 92% dans la deuxième contribution avec le Mask-RCNN.

# 2. Perspectives

Notre travail n'est que dans sa version initiale, il reste ouvert pour des améliorations et des travaux de comparaison avec d'autres méthodes de classification.

Nos futures améliorations sont :

- ✓ Nous voulons continuer à étendre notre ensemble de données en ajoutant plus d'images et en ajoutant d'autres catégories et des sous catégories de types de déchets
- ✓ Nous voulons travailler sur des autres algorithmes de IA pour augmenter et comparer les résultats et choisir le meilleur modèle.
- ✓ Nous aimerions également étendre ce projet pour identifier et classer les objets de la vidéo.

Cela peut aider à recycler les installations en traitant le flux de recyclage au lieu des éléments individuels.

# Références

- [1] MERLIN, P. & CHOAY, F. 1988. De l'article/du chapitre Dictionnaire de l'urbanisme et de l'aménagement, distributeur Presses Universitaires de France.
- [2] LABORDE, P. 1994. Les espaces urbains dans le monde. Professeur à l'université Michel-de-Montaigne de Bordeaux-III, éditions NATHAN, 83.
- [3] INEICHEN, J. 2007. Copropolis. Recherche-Action au sein de la communauté Chico Mendes, Recife, Nord-Est du Brésil.
- [4] LAURINI, R. 2014. Information systems for urban planning: a hypermedia cooperative approach, CRC Press.
- [5] GUDER, U. 2003. L'aménagement du territoire et la politique régionale en Allemagne. Notre Europe, Octobre.
- [6] MIAT , 2005 Ministère de l'Intérieur et de l'Aménagement du Territoire (MIAT), un projet d'aménagement durable pour le Luxembourg : Les principales idées du programme directeur, Luxembourg, 2005.
- [7] TOMAS, F. 1998. Vers une nouvelle culture de l'aménagement des villes. Projet urbain. Ménager les gens, aménager la ville, Wavre: Mardaga, 15-34.
- [8] MASBOUNGI, A. & DE GRAVELAINE, F. 2002. Projets urbains en France/French urban strategies. Editions du Moniteur, Paris.
- [9] LAOUAR, M. R. 2005. Contribution pour l'aide à l'évaluation de projets de déplacements urbains. Valenciennes.
- [10] BENSMAIL Salem and AINOUCHE Mohand Cherif (2010). La problématique de la gestion des déchets solides à travers les modes de traitement des déchets ménagers et hospitaliers : Cas de la commune de Bejaia. PhD thesis, ABDERRAHMANE MIRA DE BEJAIA.
- [11] Echopédia (2014). Collecte sélective des déchets. [https://www.ekopedia.fr/wiki/Tri\\_sélectif](https://www.ekopedia.fr/wiki/Tri_sélectif).
- [12] ALAM, P. & AHMADE, K. 2013. Impact of Solid Waste on Health and The Environment.
- [13] NAVARRO, A. 2003. Approche systémique des déchets, Ed. Techniques Ingénieur.
- [14] : A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," in *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229, July 1959,



- doi: 10.1147/rd.33.0210.
- [15] : <https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a> , Will Koehrsen , Jul 3, 2018, /21/02/19
- [16] : [https://R%C3%A9seau\\_neuronal\\_convolutif/16/02/19](https://R%C3%A9seau_neuronal_convolutif/16/02/19)
- [17] Girshick, R. B., Donahue, J., Darrell, T. & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524. arXiv : 1311.2524. Récupérée à partir de <http://arxiv.org/abs/1311.2524>
- [18] Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T. & Smeulders, A. W. M. (2013). Selective Search for Object Recognition. International Journal of Computer Vision, 104 (2), 154–171. doi :10.1007/s11263-013-0620-5
- [19] Ren, S., He, K., Girshick, R. B. & Sun, J. (2015). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. CoRR, abs/1506.01497. arXiv : 1506.01497. Récupérée à partir de <http://arxiv.org/abs/1506.01497>
- [20] Mohanty, S. P. (2018). CrowdAI Mapping Challenge 2018 : Baseline with Mask RCNN. <https://github.com/crowdai/crowdai-mapping-challenge-mask-rcnn>. GitHub
- [21] He, K., Gkioxari, G., Dollar, P. & Girshick, R. B. (2017). Mask R-CNN. CoRR, abs/1703.06870. arXiv : 1703.06870. Récupérée à partir de <http://arxiv.org/abs/1703.06870>
- [22] He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition. CoRR, abs/1512.03385. arXiv : 1512.03385. Récupérée à partir de <http://arxiv.org/abs/1512.03385>
- [23] : REVIEW/ Deep learning/ Yann LeCun 1,2, Yoshua Bengio3 & Geoffrey Hinton4,5 / 28 MAY 2015 | VOL 521 | NATURE /doi:10.1038/nature14539/25/02/19
- [24] : <https://towardsdatascience.com/convolutional-neural-networks-the-biologically-inspired-model-f2d23a301f71> 28/02/19
- [25] : Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps.
- [26] : Glouche, Y. and Couderc, P. (2013). A smart waste management with self-describing objects. In The Second International Conference on Smart Systems, Devices and Technologies (SMART'13).
- [27] : Arebey, M., Hannan, M., Basri, H., Begum, R. A., and Abdullah, H. (2011). Integrated technologies for solid waste bin monitoring system. Environmental

monitoring and assessment, 177(1-4):399–408

- [28] : ABDOLI, S. (2009). Rfid application in municipal solid waste management system
- [29] : Chowdhury, B. and Chowdhury, M. U. (2007). Rfidbased real-time smart waste management system. In Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian, pages 175–180. IEEE
- [30] Swedberg, C. (2008). Rfid helps reward consumers for recycling. RFID Journal, February
- [31] Parlikad, A. K. and McFarlane, D. (2007). Rfid-based product information in end-of-life decision making. Control engineering practice, 15(11):1348–1363.
- [32] Sinha, Arnab, and Paul Couderc. "Using owl ontologies for selective waste sorting and recycling." 2012.
- [33] Thung, Gary and Mingxiang Yang. "Classification of Trash for Recyclability Status." (2016).
- [34] R. A. Aral, S. R. Keskin, M. Kaya and M. Hacıomeroglu, "Classification of trashnet dataset based on deep learning models", Proc. BigData, pp. 2058-2062, Dec. 2018.
- [35] C. Bircanoglu, M. Atay, F. Beser, O. Genc and M. A. Kizrak, "RecycleNet: Intelligent waste sorting using deep neural networks", Proc. INISTA, pp. 1-7, 2018.
- [36] V. Ruiz, Á. Sánchez, J. F. Vélez and B. Raducanu, "Automatic image-based waste classification", Proc. IWINAC, vol. 2, pp. 422-431, 2019.
- [37] [https://github.com/bwproud/mask\\_rcnn\\_recyclables/blob/master/data.zip](https://github.com/bwproud/mask_rcnn_recyclables/blob/master/data.zip)



