

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche  
scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



**Mémoire de Master**

Filière Électronique  
Spécialité électronique des systemes embarquee

présenté par

Ourchane Rahma Intissar

&

Amrouche Chaima

---

## **L'apprentissage profond pour la détection et la reconnaissance des panneaux de signalisations**

---

Proposé par : Boughrira Hamida

Année Universitaire 2018-2019

NOUS REMERCIE ALLAH QUI NOUS A DONNÉ LA FORCE ET LA SAGESSE POUR  
ACHEVER CE MODESTE TRAVAILLE

NOUS REMERCIONS EN PREMIER LIEU LE TOUT PUISSANT DE NOUS AVOIR  
DONNÉ LA CHANCE, LE COURAGE ET LES MOYENS POUR L'ACCOMPLISSEMENT  
DE CE PROJET.

ON REMERCIE CORDIALEMENT, NOTRE ENCADREUR MME BOUGHRIRA  
HAMIDA, DE NOUS AVOIR SUIVI DURANT L'ACCOMPLISSEMENT DE CE  
TRAVAIL.

FINALEMENT, ON REMERCIE TOUTE PERSONNE AYANT CONTRIBUÉ À  
L'ACCOMPLISSEMENT DE CE TRAVAIL DE PRÊT OU DE LOIN ET QUI NOUS ONT  
ÉTÉ BÉNÉFIQUE DURANT PARCOURS POUR LA RÉALISATION DE CE  
TRAVAIL.

MERCI À TOUS

---

## ملخص:

الغرض من هذا العمل هو التصميم وتنفيذ واختبار طريقة الكشف والتعرف على علامات الطريق على أساس الرؤية من قبل الكمبيوتر يتكون النهج المعتمد في هذا العمل من وحدتين رئيسيتين : وحدة الكشف التي تستند على الاهتمام البصري لتحديد مناطق المشهد التي قد تحتوي على علامات الطرق و وحدة التعرف الذي يتمثل دورها في تعيين المعلومات المرئية المكتشفة مع نماذج علامات الطريق المتعلمة مسبقا على مجموعة من الصور الحقيقية لحركة المرور تظهر أداء نظام مطور حاليا.

**كلمات المفاتيح:** الرؤية الاصطناعية ,كشف علامات الطريق, التعلم العميق.

---

## Résumé :

Le but de ce travail est de concevoir, implémenter et tester une méthode de détection et de reconnaissance de panneaux routiers qui se base sur la vision par ordinateur. L'approche adoptée dans ce travail est constituée de deux modules principaux : un module de détection qui se base sur l'attention visuelle pour repérer des zones de la scène susceptibles de contenir des panneaux routiers et un module de reconnaissance dont le rôle est de mettre en correspondance l'information visuelle détectée avec des modèles de panneaux routiers appris à priori. Des tests effectués sur un ensemble d'images réelles du trafic routier montrent les performances du système actuellement développé.

**Mots clés :** vision artificielle, détection des panneaux routiers, apprentissage profond.

---

## Abstract :

The purpose of this work is to design, implement and test a method of detection and recognition of road signs which is based on computer vision. The approach adopted in this work consists of two Main modules: a detection module that is based on visual attention to identify areas of the scene likely to contain road signs and a module of recognition whose role is to map visual information detected with road signs models learned a priori. Tests carried out on a set of real images of road traffic show the performances of the currently developed system.

**Keywords :** artificial vision, detection of road signs, detection of road, deep learning.

---

## **Listes des acronymes et abréviations**

IA: intelligence artificiel

ML: machine learning

DL: deep learning

CNN: convolutional neural network

SVM: support vector machine

MLP: multi layer perceptron

RNN: recurrent neural network

FC : full connected

RPN: Région Proposition Réseaux

YOLO: You Look Only Once

SSD: Single Shot Détecteur

IDLE: Integrated DeveLopment Environment

API : interface de programmation d'application

SSV : seavus secure project file

## Table de matière

Introduction générale ..... page 1

### Chapitre 1 Etat de l'art sur la détection de panneau

I.1.introduction .....	Page3
I .2 Problématique .....	page3
I .3 Solution .....	page3
I.4. L'utilité de la signalisation routière.....	page3
I.5 Les étapes du système .....	Page6
I .6 Les approches suggérées pour la reconnaissance .....	page8
I.7 Définition de la classification .....	Page9
I.8 L'objectif de la classification .....	page9
I.9 Les différentes méthodes de la classification et l'apprentissage.....	Page9
I.10 Conclusion.....	Page12

### Chapitre 2 Deep learning (L'apprentissage profond)

II .1- Introduction .....	page13
II .2 Définition de l'intelligence artificielle (IA) .....	Page14
II. 3 Définition de la machine learning.....	Page14
II. 4 Définition de l'apprentissage profond (deep learning) .....	page14
II. 5 La différence entre la machine learning et l'apprentissage profond.....	Page14
II. 6 Domaines d'application de l'apprentissage profond .....	Page15
II. 7 Les Réseaux de Neurones .....	Page15
II. 8 Présentation de quelques Types de réseaux neuronaux.....	Page16
II.9- Quelques explications sur les CNNs.....	page18
II.10 Choix des paramètres .....	page20
II.11 Définition de Région Proposition Réseaux .....	page23
II.12 Définition de Faster R CNN .....	page23
II.13 L'architecture de Faster R CNN.....	page26
II.14 Conclusion.....	Page26

### Chapitre 3 Conception et simulation

III.1 Introduction.....	page27
III.2 Présentation des outils de développement .....	Page27
III.3-Pipeline de simulation .....	page31
III.4 évaluation.....	page45
III.5.-Conclusion .....	page45

Conclusion générale .....	page46
Bibliographie .....	page47

**Liste des figures**  
**Chapitre I**

<b>Figure I.1:</b> Exemples de panneaux.....	page 4
<b>Figure I.2:</b> Exemples des dangers pouvant être signalés grâce aux signaux routiers.....	page 4
<b>Figure I.3:</b> Exemples des indications fournies par les panneaux sur les directions à suivre.....	page 5
<b>Figure I.4:</b> Exemples des informations liées à l'usage de la route que les panneaux peuvent indiquer.....	page 5
<b>Figure I.5:</b> stratégie générale.....	page 6
<b>Figure I.6:</b> Exemples des trois catégories des formes de panneaux de signalisation les plus rencontrés.....	page 7
<b>Figure I.7:</b> Rectification des panneaux dans le cas des fortes déformations perspectives.....	page 8
<b>Figure I.8:</b> Résultat d'identification du panneau de l'idéogramme «30» détecté dans l'image de la figure I.7.....	page 8
<b>Figure I.9:</b> L'apprentissage supervisé.....	page 10
<b>Figure I.10:</b> Extrait de la classification taxinomique de Linné.....	page 10
<b>Chapitre II</b>	
<b>Figure II.1 :</b> la relation entre l'IA, le ML et le DL.....	page 13
<b>Figure II.2 :</b> la différence entre ML et DL.....	page 15
<b>Figure II.3:</b> un schéma explicatif d'un réseau de neurone artificiel.....	page 16
<b>Figure II.4:</b> Les réseaux de neurones convolutifs.....	page 19
<b>Figure II.5 :</b> exemple de max pooling 2*2.....	page 22
<b>Figure II.6:</b> Les étapes de la liere partie.....	page 23
<b>Figure II.7:</b> la convolution.....	page 24
<b>Figure II.8:</b> L'élimination des pixels avec des valeurs faibles.....	page 24
<b>Figure II.9:</b> le RPN.....	page 25
<b>Figure II.10 :</b> Prédiction des classes et des cadres de sélection.....	page 25
<b>Figure II.11 :</b> architecture générale d'un Faster R CNN.....	page 26

### Chapitre III

<b>Figure III.1</b> : python logo.....	page 28
<b>Figure III.2</b> : schémas de l'environnement de travail.....	page 30
<b>Figure III.3</b> : dataset collectée .....	page 33
<b>Figure III.4</b> : Dataset augmentation .....	page 34
<b>Figure III.5</b> : Dataset converti de RVB en niveaux de gris .....	page 35
<b>Figure III.6</b> labélisation de Dataset .....	page35
<b>Figure III.7</b> : fichier xml obtenu .....	page36
<b>Figure III.8</b> : Le scripte utiliser pour la conversion XML en CSV.....	page 36
<b>Figure III.9</b> : dataframe obtenu par la.....	page 37
<b>Figure III.10</b> : jupyter interface .....	page 37
<b>Figure III.11</b> : Architecture de CNN.....	page39
<b>Figure III.12</b> : organigramme de module de détection.....	page40
Figure III.13: : schémas simplifier du système .....	page 41
Figure III.14 l'entraînement de module .....	page 41
Figure III.15: reconnaissance panneau de signalisation en utilisant le web cam .....	page42
Figure III.16: reconnaissance panneau de signalisation en utilisant le web cam 'critere de forme' .....	page 42
Figure III.17: reconnaissance panneau de signalisation en utilisant une imag.....	page 43
Figure III.18: non détection de panneau de signalisation .....	page 43
Figure III.19: problème détection de panneau de signalisation .....	page 44
Figure III.20: problème reconnaissance de panneau de signalisation.....	page 44

## Liste des tableaux



Tableau III. 1: L'évaluation du module

# Introduction générale

---

## A. Contexte et objectif :

De nos jours, le transport idéal et le plus utilisé est sans conteste possible l'automobile. Ce moyen de transport qui a subi tant d'améliorations techniques aussi bien au niveau de la motorisation que de son aspect esthétique qui mis entre les mains d'individus s'est vite transformé en machine qui tue. La rubrique des faits divers des journaux ne manque pas de nous rappeler le décompte macabre de la liste des décès. Chaque année, on compte plus de 1,2 million de morts et 50 millions de blessés victimes d'accidents de la route occasionnant ainsi à l'humanité un coût impressionnant pour ce qui est de la prise en charge médicale et sociale de ces blessés. Par ailleurs la hausse de la production des véhicules nous pousse davantage à l'inquiétude : en effet le parc automobile qui était de l'ordre de 300 millions d'unités en 1975 va atteindre allégrement le milliard en 2011. [1] Notre pays dans ce domaine occupe une place peu enviable dans le tableau des nations détentrices de records du nombre d'accidents de la circulation. Ainsi la route aux vertus tant louées s'illustre par le vocable de « tueuse d'hommes »

Les experts et les chercheurs dans le domaine de l'automobile voient que la conduite assistée par ordinateur est une solution qui a pour objectif d'aider le conducteur à prendre les bonnes décisions dans des circonstances difficiles (mauvaise vision, fatigue) pour mieux pouvoir contrôler son véhicule et augmenter sa sécurité et la sécurité des autres conducteurs, piétons etc. prenant part à la circulation sur les routes.

La vision artificielle constitue une approche prometteuse pour aborder le problème susmentionné . Différentes installations d'assistance à la conduite pourraient être développées en se basant sur cette approche (système de freinage automatique pour ne pas dépasser la limitation de vitesse autorisée sur la route, système d'alarme qui émet un signal acoustique dès que le conducteur franchi une ligne continue sans le signaler etc.)

Le but de ce travail est de présenter un système de reconnaissance automatique des signaux routiers Algériens , connaissant les caractéristiques et les normes, telles que les informations sur les couleurs utilisées, les formes des signaux ou le lieu de stationnement des signaux routiers.

## **B. Organisation du mémoire :**

Notre travail est divisé en deux parties, dans **la première partie** on fait un état de l'art qui donne une vision générale sur les méthodes de détection et la reconnaissance des plaques , ainsi dans le **premier chapitre** on présente la détection et la reconnaissance d'images et la classification. Le **deuxième chapitre** est consacré, au deep learning, une description plus détaillée sur les réseaux de neurones convolutifs (CNNs et Faster R CNN) qui est la méthode choisie dans notre projet.

Dans la **deuxième partie**, on représente la conception de notre application, ainsi que la méthode d'implémentation de notre travail, on expliquant l'ensemble des choix techniques, (langage de programmation Python) utilisés pour la réalisation de cette application.

# Chapitre 1 Etat de l'art sur la détection de panneaux routiers

---

## **I.1.introduction :**

Depuis une vingtaine d'années, l'industrie automobile et les organismes de recherche ont pour objectif de faire rouler sur les routes des voitures bien automatisées . Ces voitures qui peuvent circuler sans intervention d'un conducteur, utilisent simultanément plusieurs technologies de capteurs et d'intelligence artificielle qui leur permettent de détecter l'environnement, de fusionner les informations obtenues pour l'analyser, comme les panneaux de signalisation routières et sa sera notre thème à traiter dans ce chapitre.

## **I .2 Problématique :**

Malgré l'évolution des systèmes automatisés dans le domaine de l'automobile, les véhicules restent toujours incapables de prendre le rôle du conducteur concernant l'identification des plaques routières par conséquent plusieurs problèmes se posent :

- La sécurité routière, le bilan des accidents augmente (Chaque année, on compte plus de 1,2 million de morts et 50 millions de blessés victimes d'accidents de la route)
- La fluidité du trafic, le temps passe dans les embouteillages (lorsque qu'il s'agit d'un accident dans notre pays toutes les routes sont bloqué pendant des heures)

## **I .3 Solution :**

Il existe plusieurs solutions parmi eux une caméra utilisée pour scanner les panneaux de signalisation. La fonction a pour nom **TSR (Traffic Sign Recognition)**.

## **I.4. L'utilité de la signalisation routière :**

Les signaux routiers indiquent les règles de la circulation établies pour permettre aux véhicules et aux piétons de se déplacer en toute sécurité sur les routes. Les

usagers de la route sont sensés être bien imprégnés de la signalisation routière qui demeure aujourd'hui un des éléments phares de la prévention routière.



**Figure(1)** Exemples de panneaux

Les panneaux de signalisation sont généralement implantés de part et d'autres sur nos routes. Leur contribution à la préservation de la sécurité routière se manifeste par un nombre considérable de fonctions, parmi lesquelles on peut citer :

⇒ Rendre plus sûre la circulation routière en informant les usagers d'éventuels dangers qu'ils peuvent rencontrer.



**Attention rond point**



**Attention travaux**



**attention collèg**

**Figure(2)** : Exemples des dangers pouvant être signalés grâce au x signaux routiers .

⇒ Faciliter cette circulation, en indiquant par exemple les directions à suivre.



**Passer à droite**



**Rambouillet**

**Figure I.3 :** Exemples des indications fournies par les panneaux sur les directions à suivre .

⇒ Donner des informations relatives à l'usage de la route.



**Interdit de faire demi-tour**



**interdit de dépasser**



**interdit de stationner**

**figure I.4** Exemples des informations liées à l'usage de la route que les panneaux peuvent indiquer.

En vue de discerner les panneaux routiers dans des images fixes ou bien dans des séquences vidéo acquises en conditions réelles, la quasi-totalité des algorithmes de traitement d'images se penchent vers deux étapes :

- **La détection des panneaux dans la scène routière.**
- **La reconnaissance des types des panneaux qui viennent d'être détectés dans l'étape de la détection.**

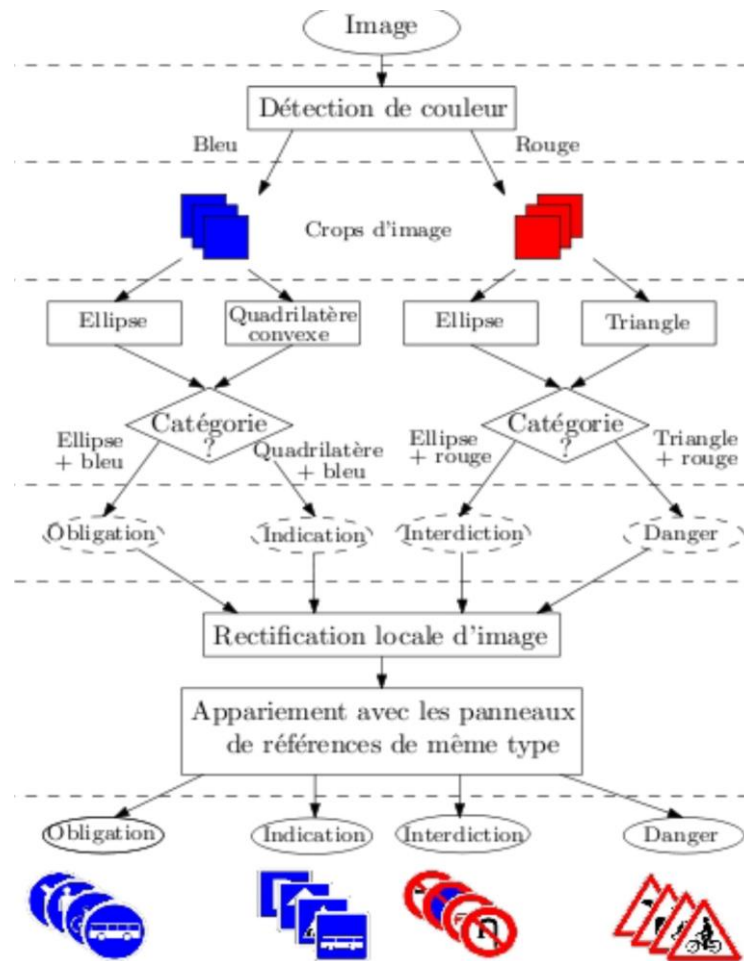


Figure I.5

stratégie générale.

## I.5 Les étapes du système :

### I.5.1 L'étape de détection

Cette étape consiste à déterminer au sein de l'image les régions supposées inclure des panneaux de signalisation (ROI : Regions Of Interests). Les signaux routiers se distinguent par des couleurs et des formes spécifiques qui les démarquent par rapport au reste de la scène. Ces informations (les formes et les couleurs) sont exploitées lors de la phase de détection des panneaux de signalisation.

#### a Utilisation du critère de la couleur

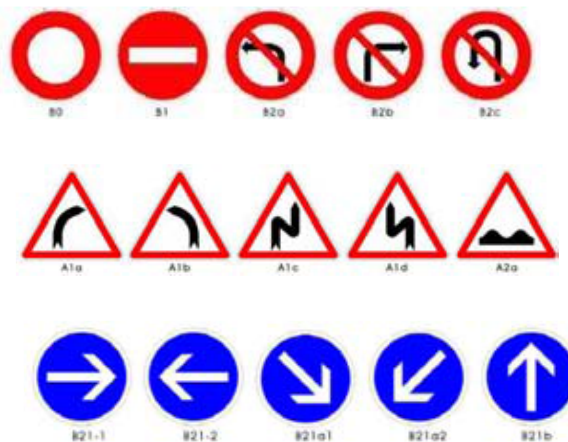
La couleur prédominante de la majeure partie des signaux est soit le rouge, soit le bleu et les idéogrammes sont souvent peints en noir sur un fond blanc. Différents espaces de colorimétrie sont utilisés notamment.

- **L'espace RGB** (Red, Green, Blue) standard où les ratios de R, G et B sont utilisés pour classifier les pixels rouges.

- **L'espace HSV** (Hue, Saturation, Value) est sans doute le plus utilisé vu sa faible sensibilité aux changements de luminosité.

### b Utilisation du critère de la forme

Les signaux routiers les plus communément rencontrés sur nos routes sont de formes circulaires, triangulaires ou bien rectangulaires. La détection de ces formes peut être considérée comme un très bon indicateur de l'existence des panneaux au sein de l'image



**Figure I.6** Exemples des trois catégories des formes de panneaux de signalisation les plus rencontrés.

### I.5.2. L'étape de reconnaissance

L'étape de reconnaissance des panneaux de signalisation résume tous les procédés déployés pour la reconnaissance du type de chaque panneau détecté. Et cela en comparant leurs illustrations (leurs idéogrammes) avec celles d'un ensemble de panneaux de référence stockés dans une base de données. La phase de reconnaissance fait appel à plusieurs descripteurs visuels (SIFT, SURF par exemple) afin de faire l'appariement entre les idéogrammes d'un panneau candidat et ceux des modèles contenus dans la base de référence.

- ❖ Il est impératif de souligner que la qualité des résultats fournis par la détection influe considérablement sur ceux de la reconnaissance. En misant sur cela et dans le but d'obtenir un système performant, on a tenu dans chapitre à relever les différentes approches proposées pour l'étape de reconnaissance.



## I.6 Les approches suggérées pour la reconnaissance

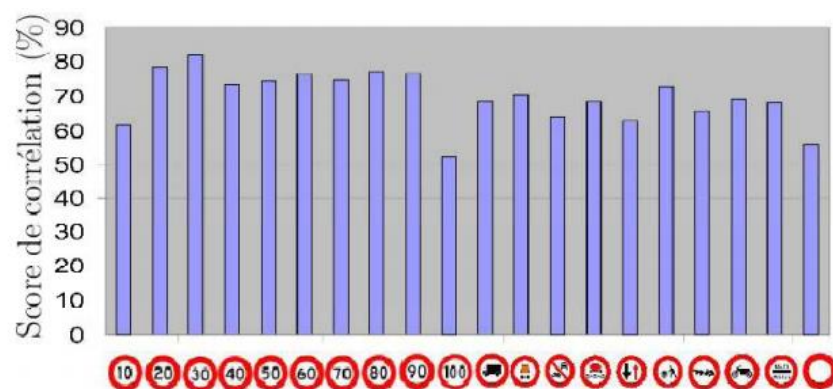
La plupart de ces approches s'appuient sur des techniques d'apprentissage qui requièrent un classifieur (cascade, SVM, réseau de neurones) qui doit être entraîné sur une base d'images et c'est ce qu'on va examiner par la suite, mais il existe d'autres qui sont axées sur une simple corrélation (appariement) entre le panneau détecté et une base de modèles de signaux. On peut donc subdiviser les approches se rapportant à la reconnaissance en deux parties

### I.6.1 Les approches à base de corrélation.

utilise une méthode de corrélation de texture. Cette méthode consiste à faire l'appariement en évaluant un score de corrélation entre les pixels de l'image du candidat et les modèles de référence.



**Figure 1.7** Rectification des panneaux dans le cas des fortes déformations perspectives



**Figure I.8** Résultat d'identification du panneau de l'idéogramme «30» détecté dans l'image de la figure I.7.

## **I.6.2 Les approches a base d'apprentissage**

Les approches avec apprentissage occupent une position prépondérante au sein des approches ayant été développées sous l'angle de la reconnaissance des panneaux de signalisation. Le principe de ces méthodes repose sur le fait de passer en entrée les panneaux repérés lors de la phase de détection en vue d'obtenir en sortie les catégories reliées à chacun d'entre eux. Nous allons présenter l'une des méthodes d'apprentissage « **la classification** » dans ce qui va suivre :

### **I.7 Définition de la classification :**

Classifier une image est une tâche ou une série de méthodes qu'une théorie unifiée pour pouvoir utiliser les images pour les analyses complémentaires ou pour la cartographie, il est souvent important de traduire l'information de fréquence contenue dans les images en information thématique portant sur l'occupation du sol ou la couverture végétale. On a généralement le choix entre deux approches : la classification supervisée et non supervisée.

### **I.8 L'objectif de la classification :**

L'objectif de la classification d'images est d'élaborer un système capable d'affecter un classement automatique d'images. Ainsi, ce système permet d'effectuer une tâche d'expertise qui peut s'avérer coûteuse à acquérir pour un être humain en raison notamment de contraintes physiques comme la concentration, la fatigue et le temps nécessaire pour un volume important de données images.

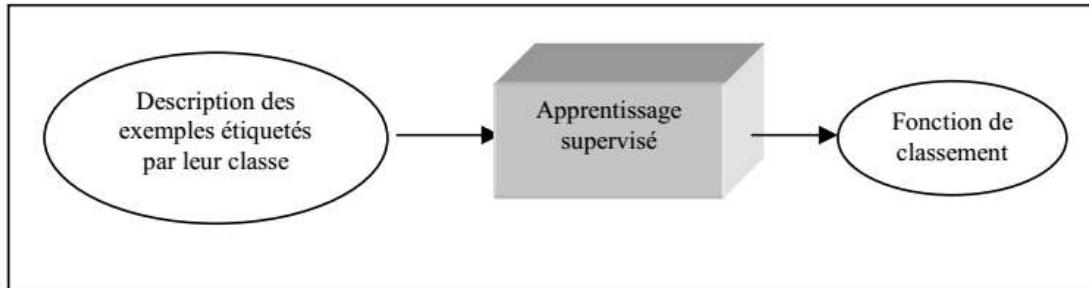
### **I.9 Les différentes méthodes de la classification et l'apprentissage :**

#### **I.9.1 Méthodes supervisées :**

Dans le cas de l'apprentissage supervisé, on dispose d'un ensemble de données étiquetées, ou d'exemples qui se sont vus associés une classe par un professeur ou un expert. Cet ensemble d'exemples constitue la base d'apprentissage.

Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de

construire à partir de la base d'apprentissage, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier

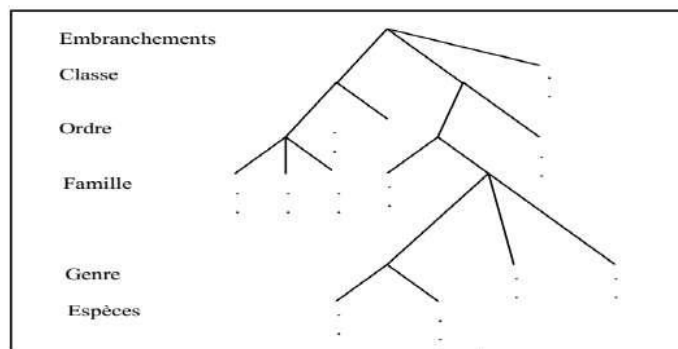


**figure I.9** L'apprentissage supervisé.

### **I.9.2 Méthodes non-supervisées :**

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

On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes. Ce type d'apprentissage, encore appelé Cluster ING ou Cluster Analysais, se trouve en classification automatique et en taxinomie numérique.



**Figure I.10** Extrait de la classification taxinomique de Linné.

### **I.9-3 Méthodes hiérarchiques :**

En classification hiérarchique ascendante. Le procédé consiste à grouper les observations individuelles en classes par a partie de la même classe.

Les méthodes se distinguent par le choix de la distance entre les observations et la définition de la stratégie d'agrégation.

Dans l'algorithme de base, le calcul de la distance (il s'agit plus exactement d'une quantité critère que l'on appelle distance par abus de langage) fait par récurrence à partir de la matrice des distances entre observations.

### **I.9-4 Méthodes non hiérarchiques :**

La classification non hiérarchique ou partitionnement, aboutissant à la décomposition de l'ensemble de tous les individus en m ensemble disjoints ou classes d'équivalence, le nombre m de classes est fixé. Le résultat obtenu est alors une partition de l'ensemble des individus, un ensemble de parties, ou classes de l'ensemble I des individus telles que :

- Toute classe soit non vide.
- Deux classes distinctes sont disjointes.
- Tout individu appartient à une classe.

### **I.9-5 Méthodes non paramétriques :**

Un classifié est dit non paramétrique si aucune distribution statistique paramétrique n'est exploitée, seule la distance spectrale sera alors prise en compte. Cette catégorie comprend notamment les méthodes fondées sur la minimisation de distance, de nouvelles méthodes apparues récemment s'ajoutent à cette catégorie comme les réseaux neuronaux et les Machines à Support Vecteurs (SVM).

### **I.9-6 Méthodes paramétriques :**

Un classifieur est dit paramétrique s'il associe à la signature spectrale (ou profil) une distribution statistique connue, le plus fréquemment pour le traitement d'images, la loi normale ou multi normale. Cette association offre la possibilité d'affecter à chaque pixel une probabilité d'appartenance à une classe donnée

### **I.9-7 Méthodes structurelles :**

Ce type de méthodes exploite des informations structurelles et contextuelle d'un objet, elles analysent l'objet en termes de ses composantes (primitives) et de leurs propriétés, on trouve par exemple l'analyse syntaxique d'une forme ou un objet à partir d'une

grammaire, la distance d'arbres, la distance de graphes (isomorphismes de graphes, de sous-graphes, avec correction d'erreurs, etc.). Dans la méthode structurelle la classe se présente principalement sous la forme de petites régions rondes

### **I.10 Conclusion :**

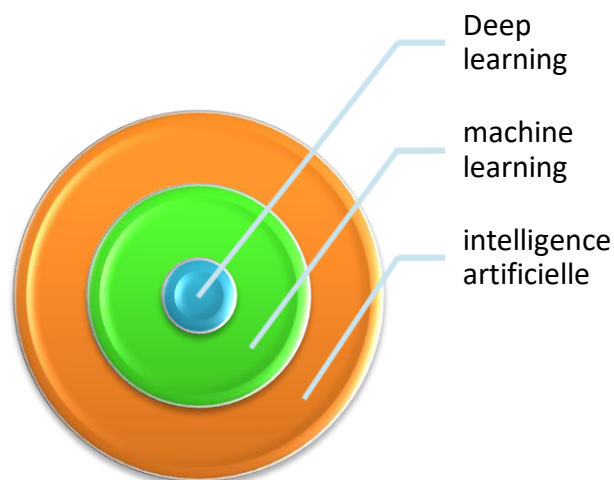
Dans ce chapitre, on a introduit l'utilité de la signalisation routière et on a donné une vision générale sur la détection et la reconnaissance des plaques routières. Dans ce chapitre on a présenté aussi quelques approches de la reconnaissance et les méthodes de classification. Dans le prochain chapitre, nous allons voir le principe de l'apprentissage profond ( Deep learning) et les réseaux de neurones convolutifs (CNNs).

## Chapitre 2 Deep learning (L'apprentissage profond).

---

### II .1- Introduction :

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage automatique champ d'étude de l'intelligence artificielle. Par conséquent, L'apprentissage profond (en anglais deep learning, deep structured learning, hierarchical learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond.



**FigureII.1** : la relation entre l'IA, le ML et le deep learning

## **II .2 Définition de l'intelligence artificielle (IA) :**

Ensemble de méthodes et techniques mises en œuvre pour réaliser des machines autonomes et capables d'apprendre et simuler l'intelligence. Il s'agit d'une science cognitive, qui fait appel à la logique mathématique, la neuroscience et l'informatique.

## **II. 3 Définition de la machine learning :**

Le machine learning est la technologie qui permet aux machines d'apprendre seules à partir des données fournies. Ces algorithmes permettent aujourd'hui de résoudre des équations ou des cas qui semblaient être insolubles.

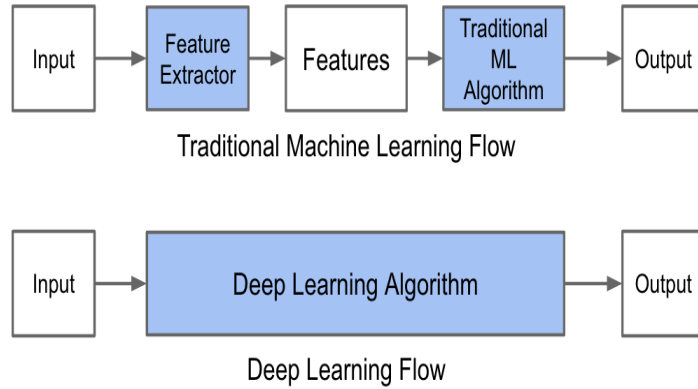
## **II. 4 Définition de l'apprentissage profond (deep learning) :**

L'apprentissage profond (« deep learning ») est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années. Il est basé sur des « réseaux de neurones artificiels », composés de milliers d'unités les « neurones » qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

## **II. 5 La différence entre la machine learning et l'apprentissage profond :**

**Machine learning** est un domaine qui utilise des techniques statistiques (ou mathématiques) pour construire un modèle (ou un système) à partir de données observées.

**Deep Learning** est une famille plus large de Machine Learning qui tente d'apprendre des fonctionnalités de haut niveau à partir des données fournies. Ainsi, le problème qu'il résout est la réduction de tâches consistant à créer de nouvelles fonctions d'extraction pour chaque type de données (speech, image, etc.).



**Figure II.2 :** la différence entre ML et DL

## II. 6 Domaines d'application de l'apprentissage profond :

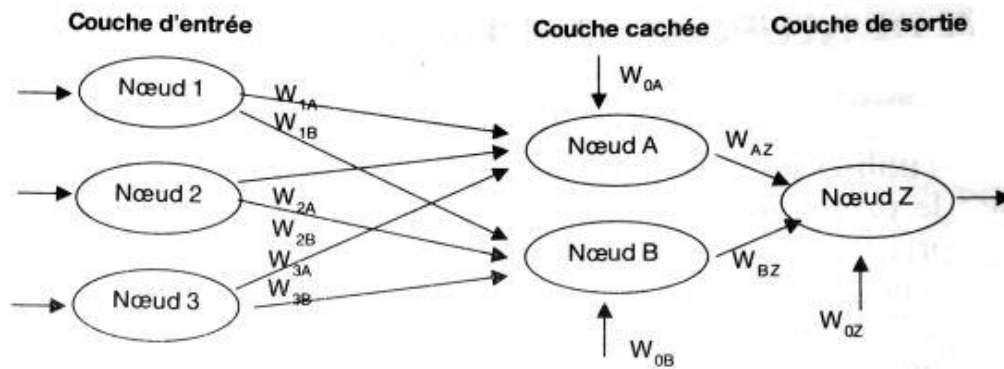
Ces techniques se développent dans le domaine de l'informatique appliquée aux NTIC (reconnaissance visuelle — par exemple d'un panneau de signalisation par un robot ou une voiture autonome — et vocale notamment) à la robotique, à la bio-informatique, la reconnaissance ou comparaison de formes, la sécurité, la santé, etc..., la pédagogie assistée par l'informatique, et plus généralement à l'intelligence artificielle. L'apprentissage profond peut par exemple permettre à un ordinateur de mieux reconnaître des objets hautement déformables et/ou analyser par exemple les émotions révélées par un visage photographié ou filmé, ou analyser les mouvements et position des doigts d'une main, ce qui peut être utile pour traduire le langage des signes, améliorer le positionnement automatique d'une caméra, etc... Elles sont utilisées pour certaines formes d'aide au diagnostic médical (ex. : reconnaissance automatique d'un cancer en imagerie médicale), ou de prospective ou de prédiction (ex. : prédiction des propriétés d'un sol filmé par un robot).

## II. 7 Les Réseaux de Neurones :

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate : le neurone formel. Les réseaux neuronaux sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone. Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage



marketing, reconnaissance de formes, traitement de signal,...) d'estimation (modélisation de phénomènes complexes,...) et prévision (bourse, ventes,...). Il existe un compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à comprendre, mais moins il sera capable de prendre en compte des dépendances trop variées.



**Figure II.3 :** un schéma explicatif d'un réseau de neurone artificiel.

## II. 8 Présentation de quelques Types de réseaux neuronaux :

Il existe beaucoup de types de réseaux neurones, chaque type étant développé pour un objectif particulier.

### II .8-1 Neurone Formel :

Un neurone formel est une représentation mathématique et informatique d'un neurone biologique. Le neurone formel possède généralement plusieurs entrées et une sortie. Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées. Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage. Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone.

Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle.

## **II .8-2 Neurones multicouche :**

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

## **II .8-3 Neurones récurrents :**

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

## **II .8-4 Réseaux de Hopfield :**

Le réseau de neurones d'Hopfield est un modèle de réseau de neurones récurrents à temps discret dont la matrice des connexions est symétrique et nulle sur la diagonale et où la dynamique est asynchrone (un seul neurone est mis à jour à chaque unité de temps).

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

## **II .8-5-Réseaux Neurones Convolutionnels :**

En apprentissage automatique, un réseau de neurone convolutifs (ou réseau de neurones à convolution, ou CNN ou ConvNet) est un type de réseau de neurones artificiels acycliques dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités

d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Dans notre travail on a travaillé avec le reseaux de neurons convolutionels .

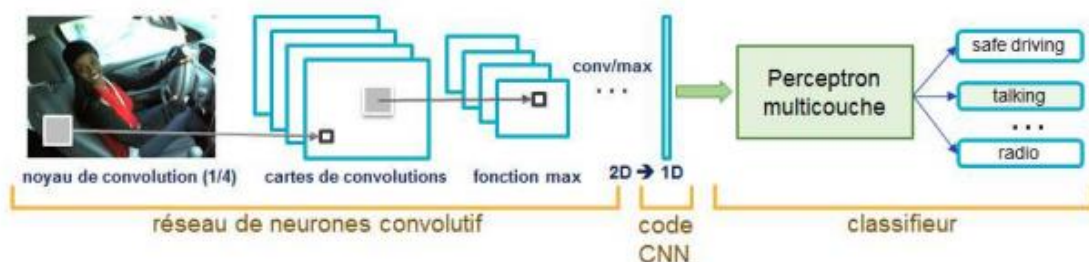
## II.9- Quelques explications sur les CNNs :

### II.9.1- Le CNN :

CNN se traduit par Convolutional Neural Networks, un algorithme très populaire pour la classification des images. Il comprend généralement des couches de convolution, des couches de fonction d'activation, des couches de mise en commun (principalement max\_pool) afin de réduire la dimensionnalité sans perdre de nombreuses caractéristiques.

### II.9.2- Principe d'architecture d'un CNN :

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Figure 2.5).certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.



FigureII.4 : Les réseaux de neurones convolutifs.

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

Une architecture CNN est formée par un empilement de couches de traitement indépendantes :

- **La couche de convolution (CONV)**
- **La couche de pooling (POOL),**
- **La couche de correction (ReLU)**
- **La couche "entièrement connectée" (FC)**
- **La couche de perte (LOSS)**

### **II.9.3-Les blocs de construction :**

#### **II.9.3.1 La couche de convolution :**

Est le bloc de construction de base d'un CNN. Trois paramètres permettent de dimensionner le volume de la couche de convolution la profondeur, le pas et la marge.

**1. Profondeur de la couche :** nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).

**2. 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 zero padding :** parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

### **II.9.3.2 La couche de Pooling :**

Un autre concept important des CNNs est le pooling, ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de  $n$  pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

### **II.9.3.3 La couche de correction (RELU) :**

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. La fonction ReLU (abréviation de Unités Rectifié linéaires) :  $F(x)=\max(0,x)$  Cette fonction force les neurones à retourner des valeurs positives.

### **II.9.3.4 Couche entièrement connectée (FC) :**

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.

### **II.9.3.5 Couche de perte (LOSS) :**

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « Softmax » permet de calculer la distribution de probabilités sur les classes de sortie.

## **II.10 Choix des paramètres :**

Les CNNs utilisent plus de paramètres qu'un MLP standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent

toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling.

### **II.10.1 Nombre de filtres :**

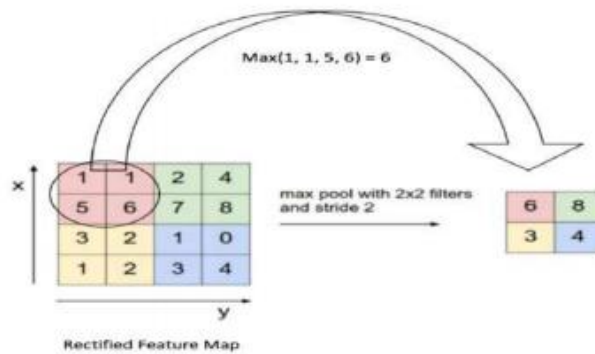
Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre. le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

### **II.10.2 Forme du filtre :**

Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST (28x28) sont habituellement dans la gamme de 5x5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12x12, voire 15x15. Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas.

### **II.10.3 Forme du Max Pooling :**

Les valeurs typiques sont 2x2. De très grands volumes d'entrée peuvent justifier un pooling 4x4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.



**FigureII.5** : exemple de max pooling 22

Par exemple, si vous alimentez une image d'une plaque de « stop » et une autre « danger piéton » l'algorithme peut vous indiquer s'il s'agit d'un stop ou d'un danger piéton.

Mais cela ne s'arrête pas là. De grandes possibilités de calcul apportent de grands progrès.

De nombreux modèles pré-formés sont développés pour les utiliser directement sans passer par la douleur des modèles de formation en raison de la limitation informatique. De nombreux modèles sont également très populaires, tels que VGG-16, ResNet 50, DeepNet, AlexNet by ImageNet.

Pour ce mémoire en particulier, on veut spécifiquement parler d'un algorithme ou d'une idée qu'on a trouvé très intelligente à partir des recherches . De nombreuses personnes implémentent **Faster R-CNN** pour identifier les objets, mais cet algorithme s'intéresse plus particulièrement à la logique et aux maths expliquant comment cet algorithme récupère les objets identifiés.

Les développeurs de l'algorithme l'appellent **Région Proposition Réseaux** abrégé en **RPN** .

Pour générer ces "propositions" pour la région où se trouve l'objet, vous **devez faire** glisser un petit **réseau** sur une carte de caractéristiques convolutionnelle qui est la sortie de la dernière couche de convolution.

## II.11 Définition de Région Proposition Réseaux :

Le réseau de propositions de région (RPN) du réseau de neurones à convolution basé sur une région plus rapide (Faster R-CNN) est utilisé pour décider «où» regarder afin de réduire les exigences de calcul du processus d'inférence global. Le RPN analyse rapidement et efficacement chaque emplacement afin de déterminer si un traitement supplémentaire doit être effectué dans une région donnée.

## II.12 Définition de Faster R CNN:

Faster RCNN est une architecture de détection d'objets présentée par Ross Girshick , Shaoqing Ren , Kaiming He et Jian Sun en 2015. Il s'agit de l'une des célèbres architectures de détection d'objets qui utilise des réseaux de neurones de convolution tels que YOLO (You Look Only Once) et SSD (Single Shot Détecteur).

Elle est composée de 3 parties

### II.12.1 Partie 1: Couches de convolution .

Dans cette couche, nous entraînons des filtres pour extraire les caractéristiques appropriées de l'image.

Les réseaux de convolution sont généralement composés de couches de **convolution**, de couches de **regroupement** et d'un dernier composant qui est la chose entièrement connectée ou étendue qui sera utilisée pour une tâche appropriée telle que la classification ou la détection.

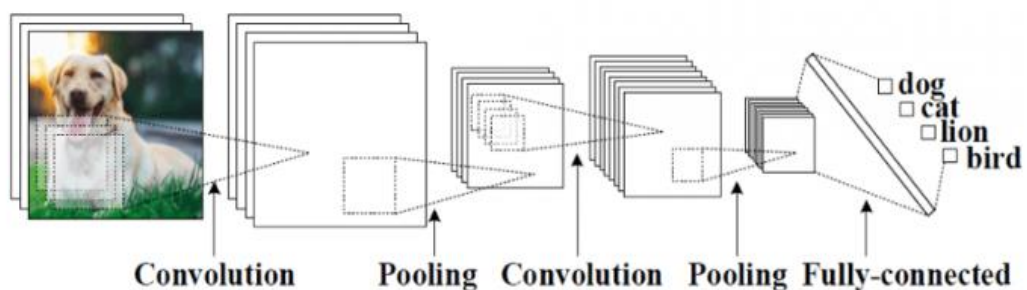
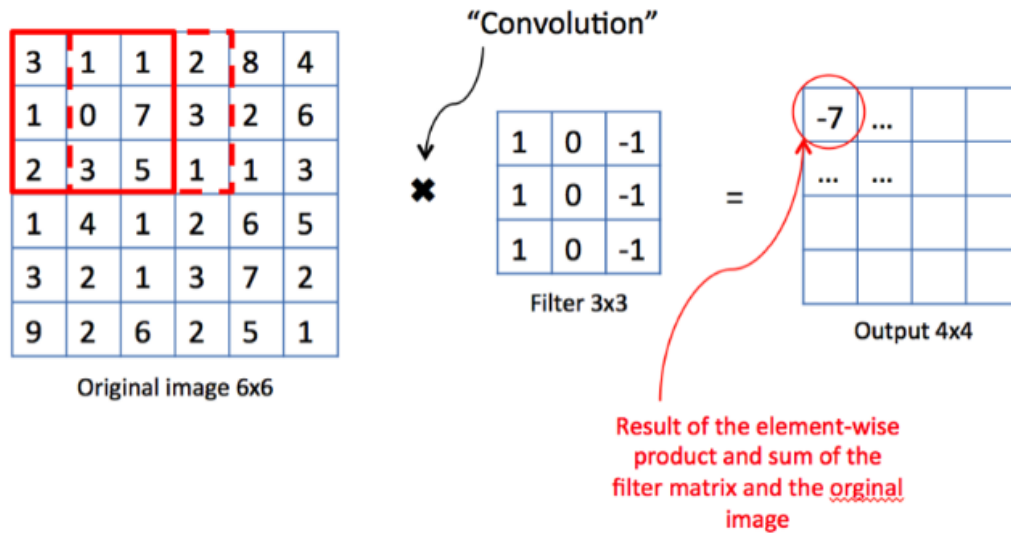


Figure II.6 Les etapes de la 1ere partie .

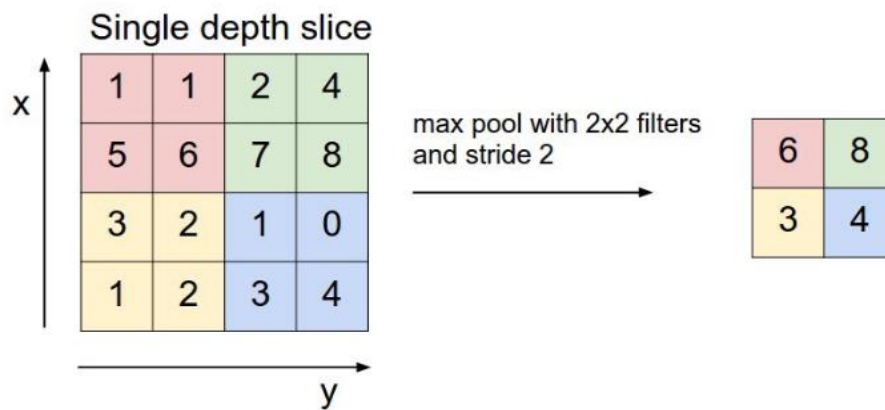


Nous calculons la convolution en faisant glisser le filtre tout au long de notre image d'entrée. Le résultat est une matrice à deux dimensions appelée carte de caractéristiques.



**Figure II.7** la convolution

Le regroupement consiste à réduire le nombre d'entités dans la carte des entités en éliminant les pixels avec des valeurs faibles.



**Figure II.8** L'élimination des pixels avec des valeurs faibles

Et la dernière chose est d'utiliser la couche entièrement connectée pour classer ces caractéristiques, ce qui n'est pas notre cas dans le RCNN plus rapide

### II.12.2 Partie 2 : Réseau de propositions régionales (RPN).

RPN est un petit réseau neuronal glissant sur la dernière carte de caractéristiques des couches de convolution et prédisant s'il y a un objet ou non et prédisant également le cadre de sélection de ces objets.

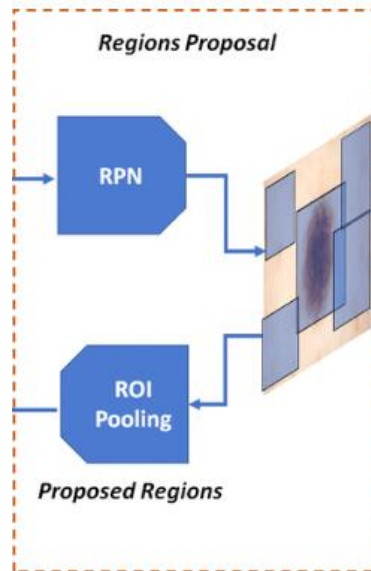


Figure II.9 le RPN

### II.12.3 Partie 3: Prédiction des classes et des cadres de sélection.

Nous utilisons maintenant un autre réseau de neurones entièrement connectés qui prend comme un inpt les régions proposées par le RPN et prédit la classe d'objets (classification) et les boîtes de sélection (Régression).

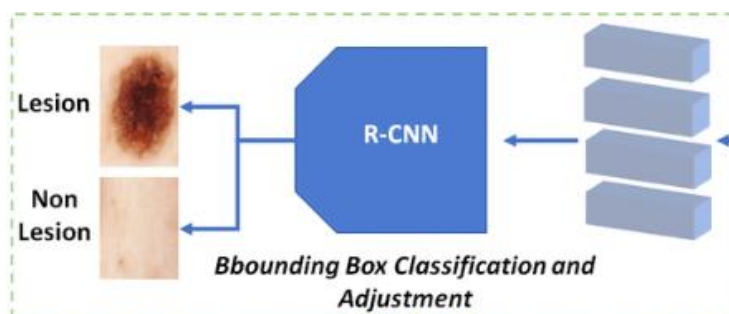
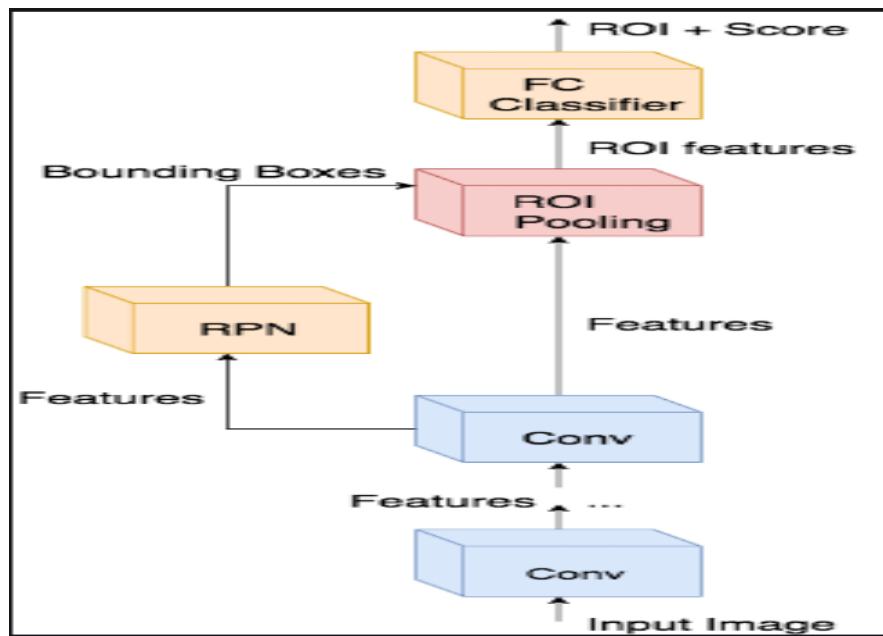


Figure II. Prédiction des classes et des cadres de sélection

### II.13 L'architecture de Faster R CNN :



**Figure II.** architecture generale d'un Faster R CNN.

### II.14 Conclusion :

Dans ce chapitre on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, Architectures....etc). Aussi qu'une vision générale sur l'apprentissage profond, toute on donnant en détail la méthode choisie dans notre travail de recherche qui est le CNNs en général et le Faster R CNN en particulier. Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application.

# Chapitre 3 Conception et simulation

---

## III.1 Introduction

Nous avons évoqué dans le chapitre précédent quels étaient les descripteurs et les classificateurs couramment mis en œuvre pour faire de la détection des panneaux de signalisation dans des images. Dans ce chapitre, nous présentons les différentes étapes réalisées durant le développement de notre application. La reconnaissance des panneaux de signalisation a pour objectif de créer un **réseau de neurones profonds** (DNN) permettant de classer les panneaux de signalisation. Nous devons former le modèle afin qu'il puisse décoder les panneaux de signalisation à partir d'images qu'on a téléchargées nous-même. Ces données doivent être préalablement prétraitées afin de maximiser les performances du modèle. Après avoir choisi l'architecture du modèle, mis au point et formé, le modèle sera testé sur de nouvelles images de panneaux de signalisation trouvés sur le Web. Parce que nous traitons avec la classification des images, un réseau de neurones convolutifs est choisi comme type de DNN, ce qui est un choix courant pour ce type de problèmes. Le code est écrit en **Python** avec l'utilisation de **TensorFlow** bibliothèque. C'est idéal pour apporter des modifications rapides et de haut niveau à notre architecture de modèle. En outre, TensorFlow prend en charge les calculs sur GPU, ce qui peut réellement accélérer les calculs requis.

## III.2 Présentation des outils de développement :

### III.2.1-Matériels utilisés

Le développement de notre application a été réalisé sous une machine qui possède la configuration suivante

- Pc portable Msi
- Disque Dur 1000 GO.
- Système d'exploitation Windows 10 64bit.
- Carte Graphique : Nvidia GeForce 840m.

- Ram: 8,00 GO.
- Processeur : Intel(R) Core(TM) i7 CPU.

### III.2.2- Langage de programmation

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

La simplicité de Python a amené de nombreux développeurs à créer de nouvelles bibliothèques d'apprentissage automatique. En raison de l'énorme collection de bibliothèques, Python est devenu très populaire parmi les experts en apprentissage automatique.



Figure III.1: python logo

## bibliothèques Python

### a. TensorFlow

TensorFlow est une plate-forme open source end to end pour l'apprentissage automatique. Il offre un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires qui permet aux chercheurs de se familiariser avec les technologies de pointe et aux développeurs de créer et de déployer facilement des

applications utilisant ML. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix.

#### **b. Keras**

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

#### **c. Numpy**

Numpy est le paquet fondamental du calcul scientifique en Python. Les tableaux NumPy facilitent les opérations mathématiques et autres types d'opérations avancés sur un grand nombre de données. En règle générale, ces opérations sont exécutées plus efficacement et avec moins de code qu'il n'est possible en utilisant les séquences intégrées de Python.

#### **d. Pandas**

Pandas est une bibliothèque d'apprentissage automatique en Python qui fournit des structures de données de haut niveau et une grande variété d'outils d'analyse. L'une des grandes caractéristiques de cette bibliothèque est la capacité de traduire des opérations complexes avec des données à l'aide d'une ou deux commandes. Les pandas ont tellement de méthodes intégrées pour grouper, combiner des données, filtrer ainsi que des fonctionnalités chronologiques.

#### **e. Pillow**

Python Imaging Library (abrégé en PIL) (dans les nouvelles versions dénommées Pillow) est une bibliothèque qui prend en charge l'ouverture, la manipulation et l'enregistrement de nombreux formats de fichier image.

## f. Matplotlib

Matplotlib est une bibliothèque de traçage Python 2D qui produit des images de qualité publication dans une variété de formats papier et d'environnements interactifs sur toutes les plateformes.

## g. OpenCV

OpenCV est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.

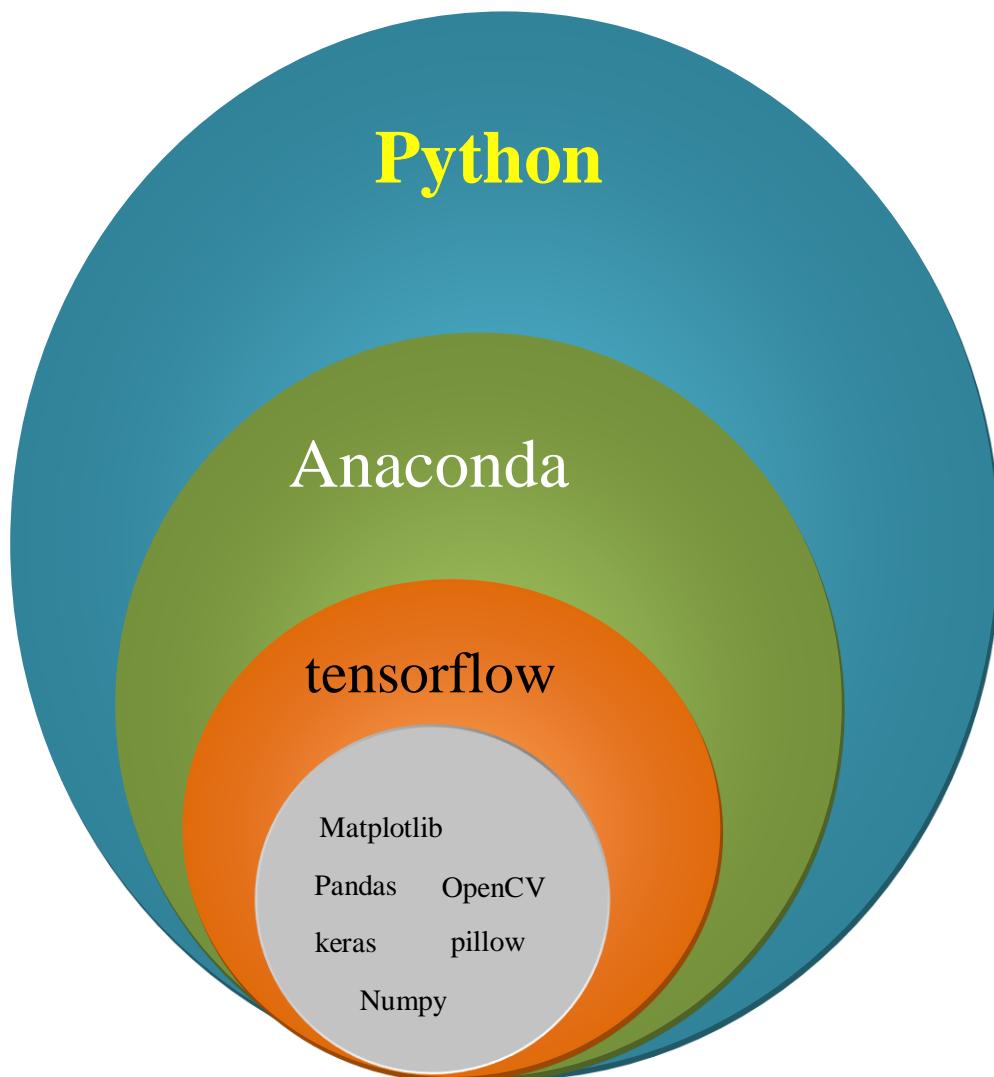


Figure III.2:schemas de l'environnement de travail

### III.3-Pipeline de simulation

Mon pipeline se compose de 6 étapes, qui sont assez courantes dans les problèmes de classification:

1. Chargement de données
2. Exploration et visualisation de jeux de données
3. Prétraitement des données
4. Préparation de l'environnement de travail
5. Conception, entraînement et le test d'un modèle CNN

#### 1. Chargement des données

Selon les pays, l'aspect visuel des panneaux de signalisation présente une grande variabilité, ce qui rend plus difficile la réussite des systèmes de classification. Soit le classificateur doit être réglé avec précision, soit une grande collection d'images doit être utilisée. il existe plusieurs de données prête à utiliser comme :

🚦 The European traffic sign dataset (ETSD)

Qui rassemble des dataset accessibles au public de 6 pays:

- la Belgique avec le jeu de données KULBelgium Traffic Signs , L'ensemble de données d'origine est divisé en 62 classes avec 4 561 images pour l'entraînement et 2 528 images pour les tests. Pour la ETSD, sa classe35 était divisée en 3 classes (3 signes obligatoires), la classe 32 en 6 classes (6 signes de prohibition) aboutissant à 69 classes européennes.
- la Croatie avec MASTIFdatasets , Il y a 3 jeux de données collectés pour différentes années (2009-2011). Le jeu de données 2011 fournit de 1 013 images avec la même résolution que dans le jeu de données 2010 720×576 pixels. 1 429 panneaux de signalisation ont été cultivés et répartis dans 53 classes selon la réalité du terrain. Il ont divisé la classe "b31" en 2 classes d'interdiction, "c35" en 2 classes de règles spéciales, "e19" en 2 classes d'informations



complémentaires et avons fusionné 12 classes en 2. Un total de 41 classes a été créé pour leur jeu de données européen.

- la France avec le jeu de données Stereopolis :
  - a. Le dataset de Stereopolis , comprend des images acquises à Paris, en France, saisissant une photo tous les 5 mètres. Il est composé de 847 images de résolution  $960 \times 1080$  pixels. Ils ont recadré les panneaux de signalisation avec la vérité au sol fournie et obtenu 271 panneaux de signalisation répartis en 10 classes. Ils ont divisé la classe "a13a" en 2 classes de danger et "b6a1" en 2 classes d'interdiction, générant 12 classes pour notre jeu de données.
  - b. Le dataset de UTBM est composé de 86 classes avec 2 631 signes. La résolution de l'image varie comme décrit dans la Section III-B. Un total de 1 863 images ont été étiquetées, comprenant 4 séquences capturées avec la caméra 3D Bumblebee et 6 séquences avec Canon Eos M.
- la Allemagne avec le référentiel TSR allemand le bien connu (GTSRB) :
  - b) Le jeu de données GTSRB contient un total de 51 839 images comprenant 43 classes. Les ensembles de l'entraînement et de test ont été utilisés pour leurs ensembles de données européens comme dans leurs définitions d'origine.
  - c) Le jeu de données GTSDb comprend 900 images complètes avec une résolution de  $1360 \times 1024$  pixels. Initialement, 43 classes sont étiquetées conformément à GTSRB, mais ils ont étendu manuellement la labellisation du jeu de données avec des valeurs de retour non prises en compte dans l'image. Ils ont obtenu 1 187 ROL composés de 46 classes supplémentaires. Au total, nous avons obtenu 89 classes pour notre ensemble de données (46 classes étiquetées + 43 classes originales). Ils ont nommé ces images avec le préfixe "GTSDb\_" suivi d'un numéro d'index unique

- les Pays-Bas avec la base de données RUG Traffic Sign Image RUG Dataset contient 48 images de taille  $360 \times 270$  pixels au format PNG. À l'origine, l'ensemble de données était utilisé pour classifier 3 classes (piéton, obligatoire pour les vélos et les intersections).ils ont étiqueté tous les panneaux de signalisation vus, y compris les 3 classes car aucune vérité au sol n'est fournie. 75 signes appartenant à 12 classes ont été recadrés et enregistrés avec le préfixe "neth\_" suivi d'un numéro unique.
- la Suède avec la base de données suédoise Traf Sign Sign (ensemble de données STS) : STS Dataset fournit 2 ensembles (Set1 et Set2) avec un total de 3 777 images étiquetées manuellement avec RoIS. 6 363 signes ont été extraits des images complètes de la vérité divisées en 19 classes. Ils ont séparé la classe "AUTRE" en 34 classes. Nous avons nommé les panneaux avec les préfixes "set1\_" et "set2\_" suivis d'un index unique pour chaque ensemble correspondant.

Il y'a aussi de Dataset malaisienne, chinoise...etc

## 1. Exploration et visualisation de jeux de données

On a décidé de créer notre propre dataset qui correspond au panneau de signalisation retrouver dans les routes algériennes. Nous avons téléchargé plus que 1500 images, En règle générale, on a basé sur les différentes **conditions d'éclairage** et l'image peut être **floue**, **pivotée** ou **redimensionnée**. En effet, il s'agit d'échantillons extraits d'images du monde réel



Figure III.3: Dataset collectée

## 2. Prétraitement des données

Pour générer **des données** dites **augmentées**, on a choisi au hasard des images à copier. Pour fournir des informations supplémentaires au modèle, nous avons pivoté cette copie de manière aléatoire et modifié sa luminosité. La bibliothèque **OpenCV** a été utilisée pour toutes ces opérations

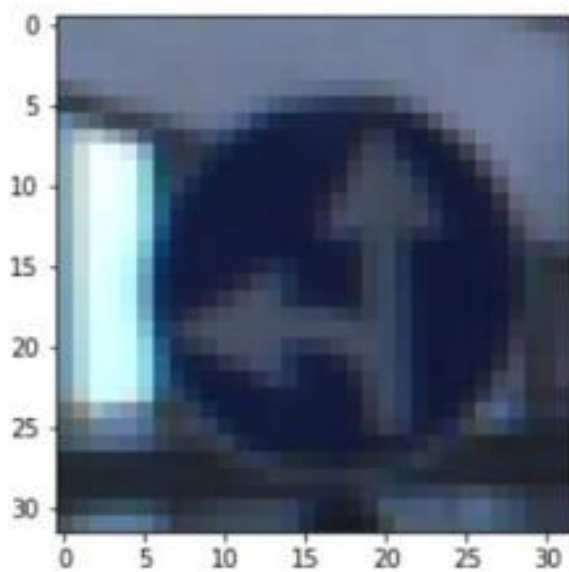


Figure III.4: Dataset augmentation

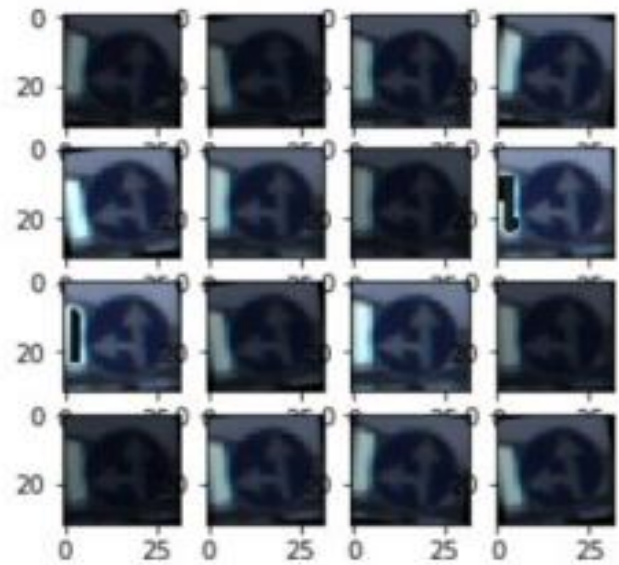


Figure III.3: Dataset collectée

Ensuite, nous avons converti les images de RVB en niveaux de gris. En conséquence, nous avons 3 fois moins de données à traiter, ce qui influe fortement sur le temps de l'entraînement. En outre, dans le document [0] traitant de la reconnaissance des panneaux de signalisation, les auteurs découvrent que le fait de rejeter les informations de couleur peut même améliorer le résultat final. Enfin, on a également normalisé et redimensionné. Voici un exemple d'image de panneau de signalisation avant et après niveaux de gris et normalisation



Figure III.5: Dataset converti de RVB en niveaux de gris

### a. Labélisation et étiquetage

L'étiquetage est la définition de  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$  et  $Y_{max}$ . Pour cela on a utilisé le logiciel labelimag. un cadre autour de chaque plaque de signalisation dans chaque image et sauvegarder sous fichier xml. Une fois que nous avons étiqueté et enregistré chaque image, il y aura un fichier xml pour chaque image.



Figure III.6: labélisation de Dataset

```

<?xml version="1.0"?>
- <annotation>
  <folder>PLAQUE ROUTIERE</folder>
  <filename>ATT travaux 27.jpg</filename>
  <path>C:\Users\salim\Documents\PLAQUE ROUTIERE\ATT travaux 27.jpg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>262</width>
    <height>192</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>attention travaux</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>53</xmin>
      <ymin>18</ymin>
      <xmax>170</xmax>
      <ymax>140</ymax>
    </bndbox>
  </object>

```

Figure III.7: fichier xml obtenu

## b. Conversion de xml en csv

La conversion des fichiers xml doit être à l'aide de la bibliothèque panda et de la bibliothèque lxml



```

File Edit Format Run Options Window Help
import os
import glob
import pandas as pd
import xml.etree.ElementTree as ET

def xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                    int(root.find('size')[0].text),
                    int(root.find('size')[1].text),
                    member[0].text,
                    int(member[4][0].text),
                    int(member[4][1].text),
                    int(member[4][2].text),
                    int(member[4][3].text)
                    )
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_df

def main():
    for folder in ['train', 'test']:
        image_path = os.path.join(os.getcwd(), ('images/' + folder))
        xml_df = xml_to_csv(image_path)
        xml_df.to_csv(('images/' + folder + '_labels.csv'), index=None)
        print('Successfully converted xml to csv.')

main()

```

Figure III.8: Le scripte utilisé pour la conversion XML en CSV

Après la conversion on obtient le dataframe présenté sous un fichier de microsoft excel comme ce ci :

	A	B	C	D	E
1	filename,width,height,class,xmin,ymin,xmax,ymax				
2	danger inconu 10.jpg,242,208,danger inconnu,27,38,184,182				
3	danger inconu 11.jpg,220,165,danger inconnu,62,24,151,98				
4	danger inconu 12.jpg,220,165,danger inconnu,166,30,202,62				
5	danger inconu 13.jpg,194,260,danger inconnu,47,137,143,221				
6	danger inconu 13.jpg,194,260,vitesse limite 30,50,17,146,110				
7	danger inconu 14.jpg,193,260,vitesse limite 30,48,15,147,112				
8	danger inconu 14.jpg,193,260,danger inconnu,47,138,143,221				
9	danger inconu 15.jpg,267,189,danger inconnu,95,6,240,140				
10	danger inconu 17.jpg,330,152,danger inconnu,180,20,314,140				
11	danger inconu 17.jpg,330,152,None,15,21,154,143				
12	danger inconu 18.jpg,300,168,danger inconnu,37,24,179,136				
13	danger inconu 19.jpg,262,192,danger inconnu,52,14,211,143				
14	danger inconu 20.jpg,179,281,danger inconnu,17,26,160,150				
15	danger inconu 21.jpg,301,167,danger inconnu,150,31,280,145				
16	danger inconu 22.jpg,301,167,danger inconnu,139,5,212,90				
17	danger inconu 22.jpg,301,167,sens interdit,30,92,48,112				
18	danger inconu 22.jpg,301,167,obliger de tourner a gauche,29,2,48,25				

Figure III.9: dataframe obtenu par la conversion

### 3. Préparation de l'environnement de travail

Plusieurs options se présentent pour utiliser python, parmi eux en distingue l'environnement qui possède de nombreuses fonctionnalités dont l'analyse des data. Anaconda est une distribution gratuite et à code source ouvert des langages de programmation Python et R pour l'informatique scientifique (informatique des données, applications d'apprentissage automatique, traitement de données à grande échelle, analyse prédictive, etc.).

Jupyter est notre environnement de développement choisi, une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala. Jupyter est une évolution du projet IPython.

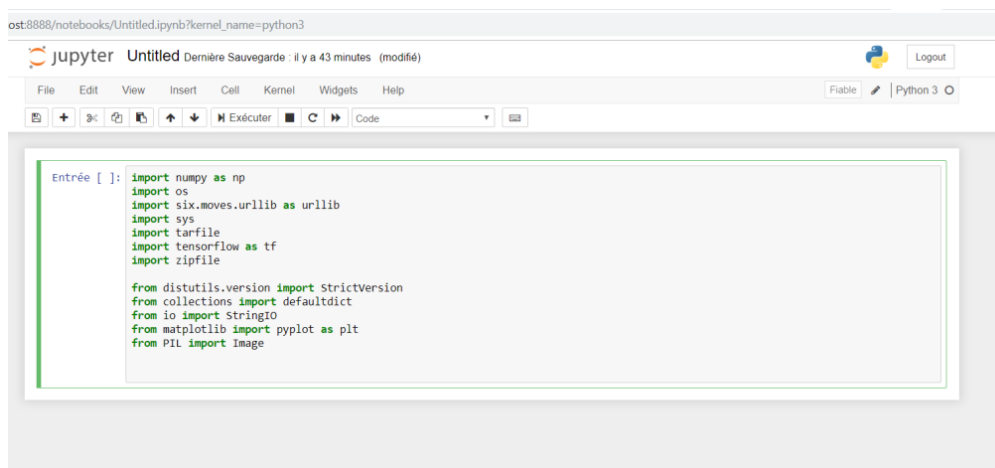


Figure III.10: jupyter interface sur windows

➤ Etape 1

La première étape consiste à créer un environnement de travail virtuel dans anaconda

```
create -n tensorflow1 pip python=3.5
```

Ensuite, l'activation l'environnement et la mise à jour pip en émettant:

```
activate tf1.12
```

```
(tf1.12) C:\Users\sali>python -m pip install --upgrade pip
```

Installation tensorflow dans cet environnement en émettant:

```
pip install --ignore-installed --upgrade tensorflow
```

➤ Etape 2

Installation des dépendances

```
conda install -c anaconda protobuf
```

```
>pip install pillow
```

```
>pip install lxml
```

```
>pip install Cython
```

```
>pip install contextlib2
```

```
>pip install matplotlib
```

```
>pip install pandas
```

```
>pip install opencv-python
```

## 4. Conception, Entraînement et test d'un modèle CNN

Il est enfin temps d'alimenter les données sur le réseau de neurones. Le choix de l'architecture du réseau, le réglage encore et encore de différents paramètres, est probablement la tâche la plus ardue. Il n'y a pas de règles claires pour l'optimisation du modèle. Lorsqu'il s'agit de réseaux de neurones profonds, nous devons attendre les résultats de chaque modèle testé pendant un temps relativement long. Bien sûr, cela dépend de la puissance de traitement disponible et le matériel utilisé.

Pour donner un bref aperçu de la complexité du modèle, on va nommer les paramètres de modèle les plus importants. Comme il s'agit d'un aperçu général du projet de reconnaissance des panneaux de signalisation, qui ont été déjà expliqués dans le chapitre précédent. Ces paramètres, parfois appelés «hyperparamètres», sont: **batch size**, le **nombre d'époques**, le **taux d'apprentissage** (Learning rate), la **régularisation des pertes** (Loss), le taux d'**abandon** ou le type de regroupement (pooling type). Ils sont souvent discutés et référencés par les chercheurs. Cependant, le choix du type et de la structure du modèle est tout aussi important. Le réseau de neurones convolutifs, convient très bien à notre tâche. Mais il existe de nombreux sous-types bien établis de CNN comme **LeNet** (années 1990), **AlexNet** (2012), **GoogLeNet** (2014) ou **VGGNet** (2014). Ils diffèrent par le nombre de couches de neurones (profondeur du modèle), les connexions entre elles, le nombre d'opérations ou les paramètres mis à jour à chaque itération.

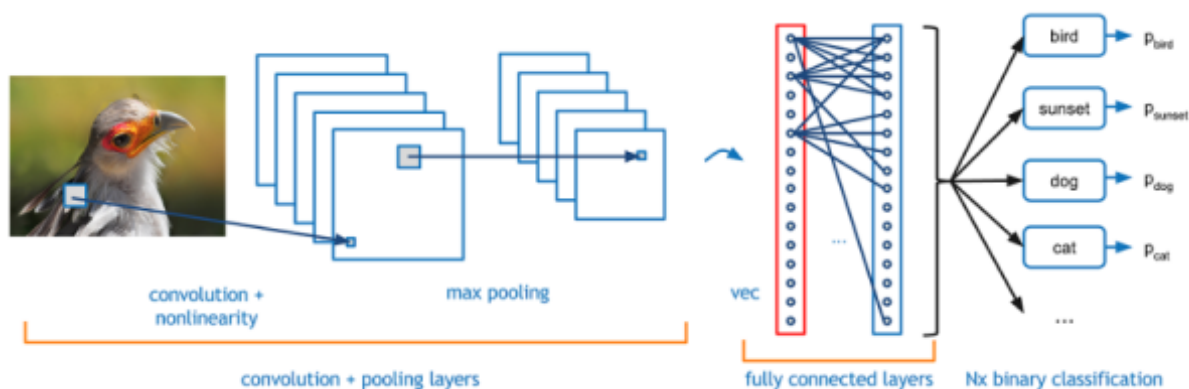


Figure III.11: Architecture de CNN

Le concept de réseaux de neurones convolutifs qu'on a déjà discuté dans le chapitre précédent. Ils ont beaucoup de succès dans la reconnaissance d'image. La partie clé à comprendre, qui distingue CNN des réseaux de neurones traditionnels, est l'opération de **convolution**. Ayant une image à l'entrée, CNN la scanne plusieurs fois pour rechercher certaines **fonctionnalités**. Ce balayage (convolution) peut être défini avec 2 paramètres principaux: foulée et type de remplissage. Comme nous le voyons sur l'image ci-dessous, le

processus de la première convolution nous donne un ensemble de nouvelles images, présentées ici dans la deuxième colonne (couche). Chaque cadre contient des informations sur une fonction et sa présence dans une image numérisée. Le cadre résultant aura des valeurs plus grandes aux endroits où une entité est fortement visible et des valeurs plus basses lorsqu'il n'y en a pas ou peu. Ensuite, le processus est répété



pour chacune des images obtenues un nombre de fois choisi. Dans ce projet, on a choisi un modèle classique faster R-cnn inception V2.

## 5.1 Conception de model

### A. Module de détection

Le module de détection se base sur l'attention visuelle [2]. En utilisant des caractéristiques visuelles intrinsèques aux panneaux routiers considérés, ce module a pour but de détecter les zones de l'image susceptibles de contenir de panneaux routiers. La figure 13 illustre les étapes de l'algorithme de détection

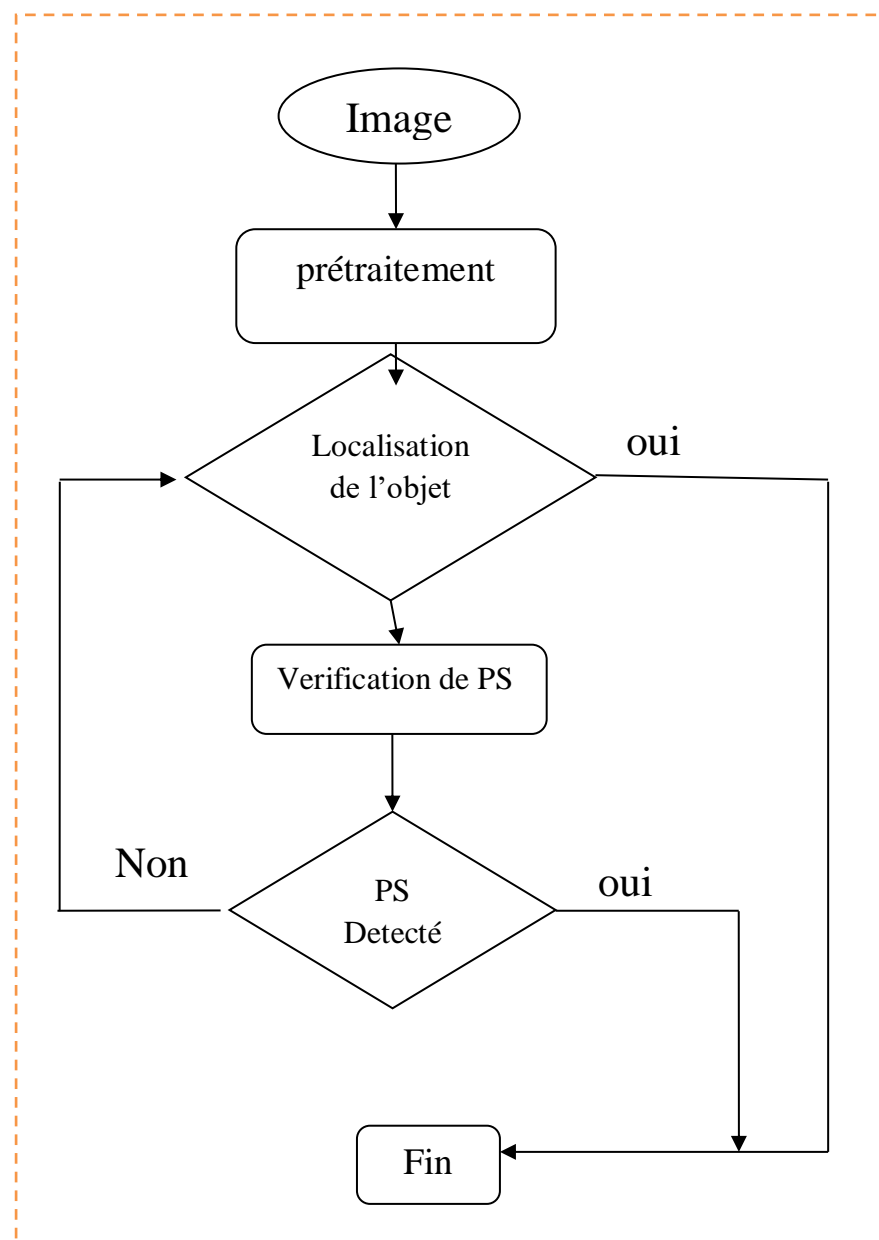


Figure III.12: organigramme de module de détection

## B. module de reconnaissance

Les zones supposées contenir des panneaux sont analysées, Une extraction des caractéristiques géométriques de l'image sera faite. Ces objets sont comparés à des modèles de panneaux appris. Le score de cette comparaison permet de vérifier et de reconnaître le contenu des zones d'intérêt.

- L'architecture simplifiée de modèle utilisé

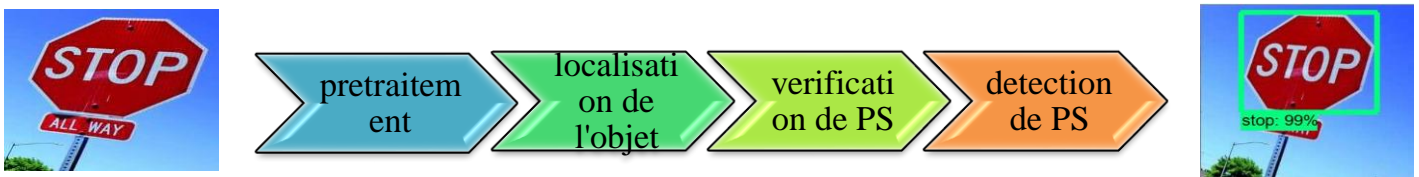


Figure III.13: schémas simplifier du système

## 5.2 L'entraînement de module

```
INFO:tensorflow:global step 1: loss = 2.7215 (24.911 sec/step)
INFO:tensorflow:global step 2: loss = 2.5805 (6.990 sec/step)
INFO:tensorflow:global step 2: loss = 2.5805 (6.990 sec/step)
INFO:tensorflow:global step 3: loss = 2.4232 (7.612 sec/step)
INFO:tensorflow:global step 3: loss = 2.4232 (7.612 sec/step)
INFO:tensorflow:global step 4: loss = 2.2665 (7.653 sec/step)
INFO:tensorflow:global step 4: loss = 2.2665 (7.653 sec/step)
INFO:tensorflow:global step 5: loss = 2.0440 (7.709 sec/step)
INFO:tensorflow:global step 5: loss = 2.0440 (7.709 sec/step)
INFO:tensorflow:global step 6: loss = 1.8033 (7.552 sec/step)
INFO:tensorflow:global step 6: loss = 1.8033 (7.552 sec/step)
INFO:tensorflow:global step 7: loss = 1.5728 (7.614 sec/step)
INFO:tensorflow:global step 7: loss = 1.5728 (7.614 sec/step)
INFO:tensorflow:global step 8: loss = 1.5073 (7.574 sec/step)
INFO:tensorflow:global step 8: loss = 1.5073 (7.574 sec/step)
INFO:tensorflow:global step 9: loss = 1.2728 (7.702 sec/step)
INFO:tensorflow:global step 9: loss = 1.2728 (7.702 sec/step)
INFO:tensorflow:global step 10: loss = 1.1442 (7.614 sec/step)
INFO:tensorflow:global step 10: loss = 1.1442 (7.614 sec/step)
INFO:tensorflow:global step 11: loss = 1.2138 (7.679 sec/step)
INFO:tensorflow:global step 11: loss = 1.2138 (7.679 sec/step)
INFO:tensorflow:global step 12: loss = 1.0293 (7.605 sec/step)
INFO:tensorflow:global step 12: loss = 1.0293 (7.605 sec/step)
INFO:tensorflow:global step 13: loss = 0.9164 (7.718 sec/step)
INFO:tensorflow:global step 13: loss = 0.9164 (7.718 sec/step)
INFO:tensorflow:global_step/sec: 0.109502
INFO:tensorflow:global_step/sec: 0.109502
INFO:tensorflow:Recording summary at step 13.
INFO:tensorflow:Recording summary at step 13.
INFO:tensorflow:global step 14: loss = 0.7916 (9.130 sec/step)
```

Figure III.14: l'entraînement de module

Chaque étape d'entraînement rapporte la perte. Il commencera haut et diminuera au fur et à mesure que l'entraînement avance.

Lorsque les pertes atteignent une valeur inférieure de 0.5 on met un fin à l'entraînement

### 5.3 tester le model

Pour tester le model on a utilisé

- ✓ Le web cam de notre laptop

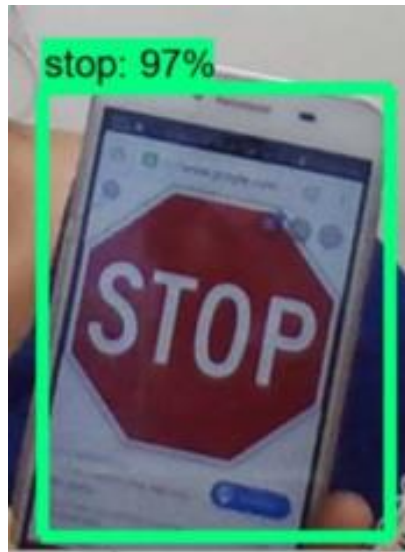


Figure III.15:reconnaissance panneau de signalisation en utilisant le web cam

Ps : le critère de notre modèle est critère de la forme et pas de la couleur c'est pour sa il n'a pas reconnu le feu rouge dans le test illustré dans la figure suivante

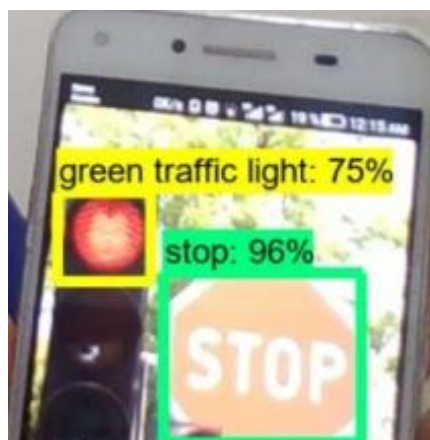


Figure III.16:reconnaissance panneau de signalisation en utilisant le web cam 'critere de forme'

✓ Une image

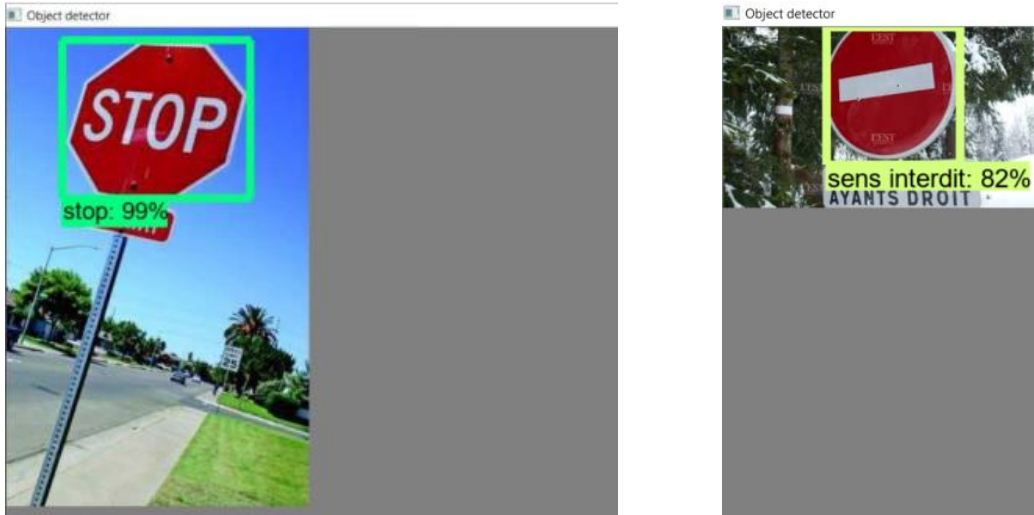


Figure III.17:reconnaissance panneau de signalisation en utilisant une image

❖ Les problèmes rencontrés durant les tests

Pendant les tests on a rencontré quelques problèmes dans la détection

- Les panneaux n'étaient ni détectés ni classifiés comme illustré dans la figure suivante :



Figure III.18: non détection de panneau de signalisation

- La reconnaissance de panneau mais un problème dans la détection

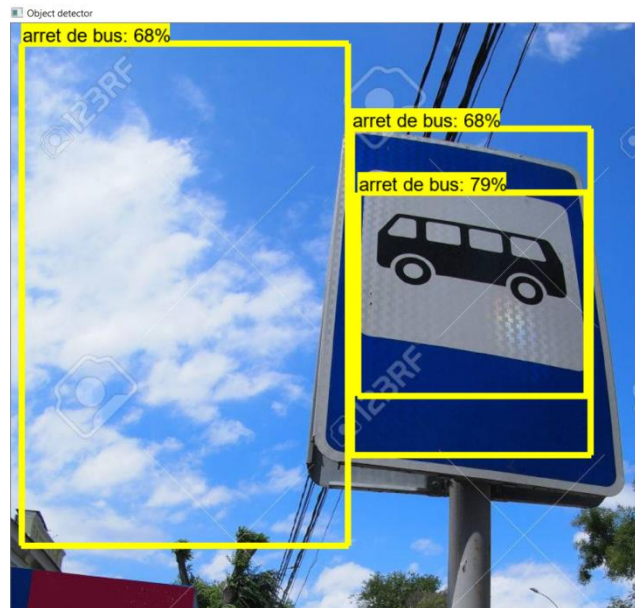


Figure III.19: probleme détection de panneau de signalisation

Pour rectifier cette erreur on a ré-labilisé et ré-étiqueté notre dataset

- Détection et reconnaissance de tous les panneaux comme sens interdit dans le 5eme essai et comme arrêt de bus dans le dernier



Figure III.20: probleme reconnaissance de panneau de signalisation

### III.4 évaluation

L'évaluation du code est effectuée sur un ensemble de 900 images de scènes routières contenant au total 1500 signaux routiers. Les résultats de cette évaluation sont résumés dans le tableau qui suit.

Types de signaux	Signaux détectés	Nombre de signaux	totaux Pourcentage de détection
PS stop	<b>97</b>	<b>98</b>	<b>99%</b>
PS sens interdit	<b>45</b>	<b>70</b>	<b>64%</b>
PS vitesse limite 80	<b>60</b>	<b>80</b>	<b>75%</b>
PS rond-point	<b>49</b>	<b>75</b>	<b>65%</b>
Totalités de PS utilisé	<b>251</b>	<b>323</b>	<b>77.7%</b>

Tableau III. 1: L'évaluation du module de la détection

Les tests montrent que le module actuellement implémenté est assez performant (taux de détection de 77.7%).

### III.5.-Conclusion

Après avoir achevé notre conception nous avons donné les outils nécessaires pour la réalisation de notre travail. Nous avons présenté aussi l'environnement de développement. A la fin on a présenté notre application en donnant quelques captures d'écran qui explique le déroulement de travail, ainsi que les résultats obtenus, les erreurs et les problèmes reconcentré.

La détection et la reconnaissance des objets est une tâche importante dans le domaine de la vision par ordinateur, la classification d'image. Bien que les capacités des activités réalisées dans le domaine de classification des images soient nombreuses, aucune méthode n'est jugée faible à 100%, mais au fur et à mesure les nouveaux travaux essayent d'améliorer les scores pour des meilleurs résultats. C'est dans ce cadre que s'inscrit notre travail, on a donné les grandes lignes pour l'élaboration d'un outil de détection et de reconnaissance automatique des signaux routiers

Pour réaliser notre travail de classification on a utilisé le deep learning, la méthode d'apprentissage qui a montré ses performances ces dernières années et nous avons choisi la méthode (Faster RCNNs) comme méthode de classification, ce choix est justifié par la simplicité et l'efficacité de la méthode. Le résultat obtenu lors de la phase de test confirme l'efficacité de notre approche. Notre travail n'est que dans sa version initiale, on peut dire que ce travail reste ouvert pour des travaux de comparaison et/ou d'hybridation avec d'autres méthodes de classification.

1. Z. Saidane and C. Garcia., "Automatic scene text recognition using a convolutional neural network.," In Workshop on Camera-Based Document Analysis and Recognition, 2007.
2. D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High performance neural networks for visual object classification.," Technical Report IDSIA-01-11, 2011.
3. <http://www-lisic.univ-littoral.fr/~verel/TEACHING/08-09/sac-M1/cRdNV9.pdf>
4. [http://conf.laas.fr/ignotus/archives/Doncescu\\_reseaux\\_neurones.pdf](http://conf.laas.fr/ignotus/archives/Doncescu_reseaux_neurones.pdf)
5. <https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss?hl=fr>
6. [https://fr.wikipedia.org/wiki/R%C3%A9tropropagation\\_du\\_gradient](https://fr.wikipedia.org/wiki/R%C3%A9tropropagation_du_gradient)
7. yacine ouassar : réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus, université pierre et marie curie paris , le 23 avril 2004
8. <https://openclassrooms.com>
9. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, Nov. 1998.
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.
11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Computer Vision and Pattern Recognition (CVPR), 2015.
12. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.
13. C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," CoRR, vol. abs/1602.07261, 2016.
14. K.He,X.Zhang,S.Ren,andJ.Sun,"Deepresiduallearningforimagerecognition," CoRR, vol. abs/1512.03385, 2015.
15. A. Veit, M. J. Wilber, and S. J. Belongie, "Residual networks are exponential ensembles of relatively shallow networks," CoRR, vol. abs/1605.06431, 2016.
16. Q. Liao and T. A. Poggio, "Bridging the gaps between residual learning, recurrent neural networks and visual cortex," CoRR, vol. abs/1604.03640, 2016.



17. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.
18. T. B. Dalal, N., "Histograms of oriented gradients for human detection," pp. 886–893, 2005.
19. P.F.Felzenszwalb, R.B.Girshick, D.McAllester, andD.Ramanan, "Objectdetection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 1627–1645, Sept. 2010.
20. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.
21. J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selectivesearchforobjectrecognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
22. I. Endres and D. Hoiem, "Category independent object proposals," in *Proceedings of the 11th European Conference on Computer Vision: Part V, ECCV'10*, (Berlin, Heidelberg), pp. 575–588, Springer-Verlag, 2010.
23. D. Hoiem, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from an image," *Int. J. Comput. Vision*, vol. 91, pp. 328–346, Feb. 2011.
24. K.He,X.Zhang,S.Ren,andJ.Sun,"Deepresiduallearningforimagerecognition," *CoRR*, vol. abs/1512.03385, 2015.
25. J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selectivesearchforobjectrecognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
26. R. Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*, 2015