

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 Systèmes Embarqués

Présenté par

DJEZZAR Khalida

&

ESSID Ikram

---

**Détection automatique des panneaux de signalisation  
routière basée sur YOLOv11**

---

Proposé par : Pr. NAMANE ABDERRAHMANE & Mr. AOUINANE BILAL

Année Universitaire 2024-2025

# Remerciement

Nous tenons dans un premier temps, à remercier grandement Pr NAMANE Abderrahmane, enseignant chercheur au Département d'Electronique, pour nous avoir accompagné durant tout notre cursus dans cette spécialité, et qu'il a su croire en ce projet et nous a fait confiance pour le mener à bien.

Ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mr AOUINANE Bilal, qui nous a fait l'honneur de nous encadrer dans ce mémoire.

Nous remercions vivement aussi tous les membres de ce jury, dont Mr KABIR Yacine, Mme BOUGHRIRA Hamida.

Qui nous ont fait l'honneur de participer au jury pour l'intérêt qu'ils ont bien voulu porter à ce travail, en apportant sa valeureuse contribution en tant qu'experts et profond connaisseur du domaine (Électronique des systèmes embarqués).

Nos remerciements par une immense reconnaissance envers tous nos enseignants qui nous ont accompagnés durant notre cursus dans cette spécialité.

Merci à nos familles. Il nous est impossible d'exprimer en quelques mots tout ce que nous devons, pour leurs encouragements et leur appui moral qui nous ont permis de mener à bon terme ce travail.

Enfin, nos remerciements vont à tous ceux qui nous ont soutenus ou qui, d'une manière ou d'une autre, ont contribué à l'élaboration de ce travail.

# Dédicace

C'est avec une profonde gratitude et sincères mots que je dédie ce modeste travail de fin d'étude :

A mes parents, que dieu me les gardes :

Ma chère mère pour sa gentillesse, son affectation, sa douceur, sa tendresse, ses encouragements éternels et sans elle rien n'aurait pu être possible.

Mon cher père pour son encouragement, sa patience, son aide continuel sur le long chemin de mes études et son soutien financier.

A mes sœurs, à mon binôme et à tous mes amis qui je les souhaite beaucoup de bonheur et de réussite dans la vie.

Et enfin, à tous ceux qui ont cru en moi et qui m'ont aidé à réaliser ce travail.

*ESSID Ikram*

# Dédicace

Je dédie ce travail à toutes les personnes qui m'ont accompagné de près ou de loin dans cette belle aventure, avec leur amour, leur soutien et leur confiance.

## **À mon père,**

Parti trop tôt, mais vivant en moi à chaque instant.

Ton absence est une douleur silencieuse, mais ton amour, ton courage, et tes sacrifices restent gravés dans mon cœur.

Chaque réussite, chaque pas en avant, est un hommage à ta mémoire. Ce travail, je te le dédie, avec tout mon amour, tout mon respect et toute ma gratitude.

**Repose en paix, Papa. Tu es et tu seras toujours avec moi.**

## **À ma mère,**

Pour son amour inconditionnel, sa force silencieuse, ses prières discrètes et sa présence rassurante à chaque étape de ma vie.

## **À mon frère Rabah,**

Pour son soutien, ses conseils et ses encouragements constants.

## **À mes sœurs Zahra, Amina,**

Pour leur affection, leur patience, leurs mots toujours réconfortants, et leur soutien de tous les instants.

## **À mes nièces Lydia, Cyrine et mon neveu Adem,**

Ces petits trésors qui illuminent ma vie et me rappellent la beauté de l'innocence et de l'espoir.

## **À mes amies Amira, Imen, Chahrazed et mon binôme Ikram,**

Pour leur écoute, leur bonne humeur, leur amitié fidèle et l'amour sincère qu'elles m'ont témoigné, qui m'ont aidé à traverser les défis avec sérénité. Je vous aime.

Enfin, à tous ceux qui m'ont soutenu, conseillé ou simplement cru en moi. Qu'ils trouvent ici l'expression de ma profonde reconnaissance, Merci du fond du cœur.

*Khalida*

## ملخص:

يهدف مشروعنا إلى تصميم نظام ذكي يعتمد على التعلم العميق للكشف والتعرف على إشارات المرور. يتم استخدام نموذج YOLO (You Only Look Once) كطريقة تعلم خاضعة للإشراف لاكتشاف الأجسام في الصور الرقمية. تم اختيار YOLOv11 كنموذج للتعلم الخاضع للإشراف نظرًا لحفّته وقدرته على العمل بكفاءة في الزمن الحقيقي، وقد تم تدريبه باستخدام بيانات مشروحة عبر منصة Roboflow. تُعد Roboflow أداةً لشرح الصور التي تحتوي على مناطق اهتمام (ROI) يجب اكتشافها.

قمنا بتنفيذ تقنيات مختلفة باستخدام إصدارات متعددة من YOLOv11 ضمن واجهة رسومية لاكتشاف إشارات المرور في الصور ومقاطع الفيديو والبت من كاميرا الويب. يتضمن النظام أيضًا وحدات للتعرف الضوئي على الأحرف (OCR)، وإدارة الدقة (DPI) لاختبار مدى قوة النظام المقترح، بالإضافة إلى استخدام صور تحتوي على ضوضاء وصور تم التقاطها في ظروف مرور حقيقية.

الكلمات الرئيسية: معالجة الصور، التعلم العميق، YOLOv11، Roboflow، لوحة التوقيع.

## Résumé :

Notre projet a pour objectif de concevoir un système intelligent basé sur l'apprentissage profond pour la détection et la reconnaissance des panneaux de signalisation routière. Le modèle **YOLO** (You Only Look Once) est utilisé comme apprentissage supervisé pour détecter des objets se trouvant sur une image numérique. Le **YOLOv11**, été choisi comme modèle d'apprentissage supervisé pour sa légèreté et sa capacité à fonctionner efficacement en temps réel, entraîné à partir de données annotées via la plateforme **Roboflow**. Le **Roboflow** est un outil pour l'annotations des images sur lesquelles figurent des régions d'intérêt à détecter (ROI, Region of interest). Nous avons mis en œuvre diverses techniques utilisant différentes versions de Yolo11 au sein de l'interface graphique pour détecter des panneaux de signalisation sur des images, des vidéos et webcam. Il intègre notamment des modules d'OCR (Reconnaissance Optique de Caractères), une gestion de la résolution (DPI) pour tester la robustesse du système proposé, des images bruitées utilisé ainsi que des images prise dans des conditions réelles de circulation.

**Mots clés :** YOLOv11, traitement d'image, apprentissage profond, Roboflow, panneaux de signalisation.

**Abstract:**

Our project aims to design an intelligent system based on deep learning for the detection and recognition of road traffic signs. The YOLO (You Only Look Once) model is used as a supervised learning approach to detect objects in digital images. YOLOv11 was chosen as the supervised learning model for its lightweight nature and its ability to operate efficiently in real-time, trained using annotated data from the Roboflow platform. Roboflow is a tool for annotating images that contain regions of interest (ROI) to be detected.

We implemented various techniques using different versions of YOLOv11 within a graphical interface to detect traffic signs in images, videos, and webcam streams. The system also integrates OCR (Optical Character Recognition) modules, resolution (DPI) management to test the robustness of the proposed system, and includes the use of noisy images as well as images captured under real-world traffic conditions.

**Keywords:** YOLOv11, image processing, deep learning, Roboflow, sign plate.

## Liste des acronymes et abréviations

<b>YOLO</b>	You Only Look Once
<b>ROI</b>	Region of Interest
<b>OCR</b>	Optical Character Recognition
<b>DPI</b>	Dots Per Inch
<b>IA</b>	Intelligence Artificielle
<b>CNN</b>	Convolutional Neural Network
<b>OpenCV</b>	Open Source Computer Vision Library
<b>ADAS</b>	Advanced Driver Assistance Systems
<b>GED</b>	Gestion Électronique des Documents
<b>ILSVRC</b>	ImageNet Large Scale Visual Recognition Challenge
<b>RNA</b>	Réseau de Neurones Artificiels
<b>ANN</b>	Artificial Neural Network
<b>RNN</b>	Recurrent Neural Network
<b>FCL</b>	Fully Connected Layer
<b>ReLU</b>	Rectified Linear Unit
<b>IPS</b>	Instructions Per Second
<b>CSP</b>	Cross Stage Partial
<b>API</b>	Application Programming Interface
<b>C3k2</b>	C3 block with kernel size 2
<b>SPPF</b>	Spatial Pyramid Pooling Fast
<b>C2PSA</b>	Convolutional block with Parallel Spatial Attention
<b>U-Net</b>	U-shaped Network

<b>FPS</b>	Frames Per Second
<b>F1-Score</b>	Harmonic mean of precision and recall
<b>TP</b>	True Positive
<b>FP</b>	False Positive
<b>FN</b>	False Negative
<b>TN</b>	True Negative
<b>GPU</b>	Graphics Processing Unit
<b>TPU</b>	Tensor Processing Unit
<b>VS Code</b>	Visual Studio Code
<b>Git</b>	Git Version Control System
<b>mAP</b>	Mean Average Precision
<b>IoU</b>	Intersection over Union
<b>PR</b>	précision-rappel

## Contents

Liste des figures .....	X
Liste des tableaux .....	XII
<b>Introduction générale</b> .....	1
<b>Chapitre 01 :</b> .....	2
<b>Généralité sur les panneaux de signalisation routière</b> .....	2
1.1 Introduction .....	5
1.2 Définition du panneau de signalisation.....	5
1.2.1 Fonctions des panneaux de signalisation .....	6
1.2.2 Type de panneaux de signalisation routière.....	6
1.2.3 Cinq grandes familles de panneaux de signalisation routière .....	8
1.2.4 Différents types de signalisation routière.....	11
1.2.5 Panneaux temporaires.....	11
1.2.6 Couleurs des panneaux .....	12
1.2.7Panneau "P" .....	12
1.2.8 Panneau bleu.....	12
1.3 Traitement d'image.....	14
1.3.1 Définition d'image .....	14
1.3.2 Bruit sel et poivre.....	15
1.3.3 Bruit gaussien .....	15
1.3.4 Bruit speckle .....	16
1.3.5 Flou .....	17
1.3.6 Points par pouce (DPI).....	17
1.3.7 Reconnaissance Optique de Caractères (OCR) .....	19
1.4 Conclusion.....	21
<b>Chapitre 02 :</b> .....	22
<b>Traitement d'images et Apprentissage profond</b> .....	22
2.1 Introduction .....	23
2.2 Introduction au traitement d'image.....	23
2.3Vision par ordinateur.....	24
2.4 L'Apprentissage profond.....	24
2.4.1 Réseaux neuronaux artificiels .....	25
2.4.2 Architecture de base du perceptron .....	25
2.4.3 Architecture du réseau neuronal .....	26

2.4.4 Structure du réseau neuronal.....	27
2.4.5 Réseaux neuronaux convolutifs (CNN).....	28
2.5.1 Algorithme YOLO.....	35
2.5.2.....	Architecture de YOLO11
.....	37
2.5.3 Présentation des différentes tailles de modèles YOLOv11 .....	38
2.5.4 Les ancres (ancres) .....	40
2.5.5 Evaluation du model YOLOv11.....	41
2.6 Conclusion :.....	43
<b>Chapitre 03 :</b> .....	44
<b>Implémentation, Résultat et Discussion</b> .....	44
3.1 Introduction .....	45
3.2 Plats-forme .....	45
3.2.1 Google Colab.....	45
3.2.2 Roboflow.....	46
3.3 Logiciels utilisés -Visual Studio Code (VS Code) . .....	46
3.4 Dataset .....	47
3.5 Détails del'implémentation .....	47
3.5.1 Annotation du Datasets via Roboflow .....	48
3.5.2 Entraînement du modèle YOLOv11 .....	49
3.6 Interface graphique .....	50
3.6.1 Interprétions Globale avec l'interface graphique.....	64
3.7 Conclusion.....	71
<b>Conclusion générale</b> .....	72

## Liste des figures

<b>Figure 1.1</b> : Panneau de signalisation de type sens-interdit [3].....	6
<b>Figure 1.2</b> : Signification de chacune panneaux de signalisation [5].....	6
<b>Figure 1.3</b> : Panneau triangle. ....	7
<b>Figure 1.4</b> : Panneau rond. ....	7
<b>Figure 1.5</b> : Panneau de forme carrée.....	7
<b>Figure 1.6</b> : Panneau en forme de flèche.....	8
<b>Figure 1.7</b> : Panneau rectangulaire.....	8
<b>Figure 1.8</b> : Familles de panneaux de signalisation [6].....	8
<b>Figure 1.9</b> : Signaux d'interdiction ou de restriction [7].....	9
<b>Figure 1.10</b> : Signaux d'avertissement de danger [8].....	9
<b>Figure 1.11</b> :Panneaux de priorité [9].....	10
<b>Figure 1.12</b> : Signaux de fin d'interdiction ou de restriction [10]. ....	10
<b>Figure 1.13</b> :Signaux d'obligations [11]. ....	11
<b>Figure 1.14</b> :Panneaux temporaires [13]. ....	11
<b>Figure 1.15</b> : Le panneau "P" [15].....	12
<b>Figure 1.16</b> : Exemple de panneau bleu [16]. ....	12
<b>Figure 1.17</b> : Panneau de limitation de vitesse 60 km/h en niveaux de gris – Algérie .....	14
<b>Figure 1.18</b> : Application du Bruit impulsif a une image de panneau de <i>Rond-point</i> en Algérie (P = 0.05).....	15
<b>Figure 1.19</b> : Panneau <i>Arrêt et stationnement interdit</i> en Algérie avec ajout de bruit gaussien (m = 0, $\sigma = 0.15$ ).....	16
<b>Figure 1.20</b> : Panneau de signalisation <i>Attention rond-point</i> en Algérie affecté par un bruit speckle.....	16
<b>Figure 1.21</b> : Image du panneau <i>Cédez le passage à l'intersection</i> en Algérie – flou gaussien appliqué (Taille 5×5, $\sigma = 2$ ).....	17
<b>Figure 1.22</b> : La Résolution des Images (le rôle de PDI) [8].....	18
<b>Figure 1.23</b> : Extraction automatique de texte d'une image avec EasyOCR [25].....	19
<b>Figure 1.24</b> : Reconnaissance de texte automatique avec TesseractOCR [27].....	20
<b>Figure 2.1</b> : Neurones : (a) biologique, (b) artificiel [33].....	25
<b>Figure 2.2</b> : Architecture de base d'un perceptron [36]. ....	26
<b>Figure 2.3</b> : Architecture d'un réseau de neurones artificiels [38].....	27
<b>Figure 2.4</b> : Perceptrons multicouches [40]. ....	28
<b>Figure 2.5</b> : Architecture CNN [42]. ....	29
<b>Figure 2.6</b> : Exemple d'une convolution avec filtre de taille (3x3) [43].....	29
<b>Figure 2.7</b> : Schéma du parcours de la fenêtre de filtre sur l'image [45]. ....	30
<b>Figure 2.8</b> :Zero Padding [46].....	31
<b>Figure 2.9</b> : Max Pooling [48].....	31
<b>Figure 2.10</b> : Average pool [49]. ....	32
<b>Figure 2.11</b> : La couche entièrement connectée (Fully Connected Layer, FCL) [50]. ....	32

<b>Figure 2.12</b> : Fonction d'activation (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LeakyReLU [56].	34
<b>Figure 2.13</b> : YOLOv11 [58].	35
<b>Figure 2.14</b> : Architecture de YOLOv11 [72].	37
<b>Figure 2.15</b> : Matrice de confusion [75].	41
<b>Figure 3.1</b> : Exemples d'images issues du dataset personnalisé.	48
<b>Figure 3.2</b> : Procédure d'annotation des panneaux de signalisations à l'aide de Roboflow.	49
<b>Figure 3.3</b> : l'image contenant l'interface graphique Tkinter.	51
<b>Figure 3.4</b> : Affichage des statistiques de détection dans l'interface.	52
<b>Figure 3.5</b> : Courbes d'Entraînement et de Validation du Modèle YOLOv11m.	53
<b>Figure 3.6</b> : Analyse des prédictions via la matrice de confusion.	54
<b>Figure 3.7</b> : Évaluation des Performances sur Images Nettes en Plein Jour.	56
<b>Figure 3.8</b> : Résultats Expérimentaux en Conditions de Faible Luminosité.	57
<b>Figure 3.9</b> : Tests de Robustesse sur Images Floues : Scènes de Jour et de Nuit.	57
<b>Figure 3.10</b> : Étude de la Robustesse en Condition Dégradée : Bruit Gaussien, Poivre et Sel, Speckle.	58
<b>Figure 3.11</b> : comparatif de l'Accuracy et de la Précision (%) des variantes YOLOv11 en fonction du seuil de confiance.	60
<b>Figure 3.12</b> : Graphe d'évolution de l'Accuracy (%) des modèles YOLOv11 en fonction du seuil de confiance.	61
<b>Figure 3.13</b> : Graphe d'évolution de précision des modèles YOLOv11 en fonction du seuil de confiance.	61
<b>Figure 3.14</b> : Comparaison de l'accuracy et de la précision a seuil de 0.9 des modèles YOLOv11.	63
<b>Figure 3.15</b> : Graphe de Résultats pour le Modèle YOLOv11n.	64
<b>Figure 3.16</b> : Graphiques de Résultats pour le Modèle YOLOv11s.	65
<b>Figure 3.17</b> : Graphiques de Résultats pour le Modèle YOLOv11m.	66
<b>Figure 3.18</b> : Graphiques de Résultats pour le Modèle YOLOv11l.	67
<b>Figure 3.19</b> : Graphiques de Résultats pour le Modèle YOLOv11x.	68

## Liste des tableaux

<b>Tableau 1.1</b> : Panneaux de signalisation routière en Algérie.....	13
<b>Tableau 2.1</b> : comparatif entre les différentes variantes de l'architecture YOLOv11.....	40
<b>Tableau 2.2</b> : Définition des termes TP, TN, FP, FN.....	42
<b>Tableau 2.3</b> : Exemple concret de TP, TN, FP, FN (application sur les panneaux "Stop").....	42
<b>Tableau 3.1</b> : Analyse des performances des modèles YOLOv11 sur 433 images à différents seuils de confiance.....	59
<b>Tableau 3.2</b> : Discussion des résultats pour les différentes version du model YOLO11.....	62
<b>Tableau 3.4</b> : Comparaison des modèles YOLOv11 (n, s, m, l, x).....	69

---

## ***Introduction générale***

---

L'intelligence artificielle (IA) constitue aujourd'hui l'un des domaines de recherche les plus dynamiques et prometteurs, émergeant comme une force transformatrice dans de nombreux domaines. Parmi les domaines de recherche très connus dans IA on a le « deep learning » ou bien l'apprentissage au profond dans le domaine du traitement d'images, grâce à ces algorithmes, elle est possible d'analyser l'image, détecter et reconnaître les objets dans l'image.

La sécurité routière est un enjeu majeur dans notre société moderne. Pour y contribuer, les nouvelles technologies jouent un rôle de plus en plus important, notamment grâce à l'intelligence artificielle. Dans le cadre de ce projet de fin d'études, nous nous sommes intéressés à la détection automatique des panneaux de signalisation à l'aide des techniques de vision par ordinateur à partir d'images, vidéos, webcam.

En vue d'atteindre cet objectif, nous avons utilisé le modèle **YOLOv11**, est un modèle d'apprentissage profond basé sur les réseaux neuronaux convolutifs (CNN), qui est largement reconnu pour sa rapidité exceptionnelle, efficacité, et sa précision. Nous l'avons conçu sur mesure pour détecter les panneaux de signalisation, notamment une base de données personnelle avec 2200 images, grâce à cette base de données, nous permet d'entraîner notre modèle de détection d'objets personnalisé dans les meilleures conditions possibles.

Afin de démontrer l'efficacité et l'applicabilité de notre système de détection de panneaux de signalisation, nous avons développé une interface graphique conviviale à l'aide de la bibliothèque Tkinter. Cette interface constitue une plate-forme interactive permettant aux utilisateurs de charger facilement des images et d'obtenir, en temps réel, les résultats de la détection. Chaque panneau détecté est clairement annoté sur l'image traitée, accompagné de ses informations spécifiques.

Au fil de ce travail de recherche, nous procéderons à une évaluation approfondie des performances de notre modèle de détection de panneaux de signalisation. Nous analyserons la précision des prédictions, l'exactitude de la classification des panneaux, ainsi que la robustesse du système face à des variations d'images (angles, luminosité, occlusions, etc.). Nous discuterons également des défis rencontrés et des pistes d'amélioration possibles, notamment à travers l'intégration de techniques avancées d'apprentissage profond et l'exploitation de bibliothèques telles qu'OpenCV. Cette recherche s'inscrit dans le domaine en plein essor de la détection d'objets et met en évidence la capacité du deep learning à répondre efficacement à des problématiques concrètes, telles que l'identification automatique et fiable des panneaux de signalisation.

Ce mémoire se compose de trois chapitres, suivis d'une conclusion générale :

- ❖ **Chapitre 1** : Ce chapitre explique l'importance des panneaux de signalisation et des modèles de détection pour les systèmes d'aide à la conduite, notamment en Algérie où des particularités comme l'usage de deux langues, la diversité des routes et la dégradation des panneaux compliquent leur reconnaissance. Il présente également les principales techniques de traitement d'image utilisées pour les détecter efficacement.
- ❖ **Chapitre 2** : Le deuxième chapitre expose les bases théoriques du projet, en présentant les principes de l'apprentissage profond, le fonctionnement des réseaux de neurones et les architectures courantes. Il met l'accent sur l'architecture YOLO, utilisée pour la détection et la reconnaissance des panneaux de signalisation.
- ❖ **Chapitre 3** : Ce chapitre décrit la mise en œuvre du système de détection des panneaux de signalisation, en intégrant le traitement d'image et le deep learning, avec une focalisation sur le modèle YOLO. Il analyse les résultats pour évaluer les performances en conditions réelles.

---

***Chapitre 01 :***  
***Généralité sur les panneaux de***  
***signalisation routière***

---

## **1.1 Introduction**

La signalisation routière joue un rôle important en matière de sécurité routière car elle contribue à la lisibilité de la route. Utilisés depuis de nombreuses années pour organiser, sécuriser et fluidifier les déplacements. Les progrès technologiques et les moyens de communication ont certainement changé ces outils, mais ils restent des éléments essentiels pour assurer la voie aux chauffeurs et aux piétons et leur permettre de connaître les différents risques. La signalisation routière est un outil indispensable pour aider les gens à naviguer dans les règles de circulation et à éviter les dangers de la route [1].

Tous les pays du monde possèdent des panneaux de signalisation, dont la plupart obéissent à des normes visuelles précises pour une compréhension rapide et universelle. Il existe plusieurs types de panneau, y compris de danger, d'interdiction, d'obligation et d'indication. Il est essentiel de les reconnaître rapidement et correctement, en particulier dans un environnement routier très fréquenté ou pour les systèmes qui fonctionnent automatiquement.

Avec les récents progrès de l'intelligence artificielle et de la vision par ordinateur, il est devenu impératif de développer des systèmes automatiques de détecter et de reconnaître les panneaux de signalisation. Ces techniques permettent d'identifier de manière précise et en temps réel les panneaux à partir d'images, vidéos, webcam ou d'application mobile. Cela est un sujet pour les voitures autonomes, les systèmes d'aide à la conduite (ADAS), mais aussi pour l'analyse du trafic et la sécurité routière.

## **1.2 Définition du panneau de signalisation**

Les panneaux de signalisation sont des panneaux installés sur le bord de la route pour informer les usagers de la route. Ce sont des symboles visuels qui transmettent des informations sur les règles de circulation et d'autres informations pertinentes. Les illustrations remplacent les mots. En tant que dispositifs de contrôle de la circulation, avertir des dangers, ou indiquer des obligations ou des directions. Ils nécessitent une attention particulière, du respect et une réponse appropriée de la part du conducteur (voir la figure 1.1) [2].



Figure 1.1 : Panneau de signalisation de type sens-interdit [3].

### 1.2.1 Fonctions des panneaux de signalisation

Les panneaux de signalisation ont plusieurs fonctions :

- Informer les usagers des dangers qu'ils pourraient rencontrer.
- Indiquer les directions à suivre.
- Indiquer ou rappeler certaines prescriptions particulières de la police.
- Donner des informations quant à l'état et à l'usage de la route [4]

### 1.2.2 Type de panneaux de signalisation routière

Il existe 5 types de panneaux : triangulaire, rond, carré, en forme de flèche et rectangulaire (voir la figure 1.2) [5].







Forme						
Situé à	Agglomération : 50m Hors Agglomération: 150m	à l'endroit	à l'endroit	à l'endroit ou avant	à l'endroit	à l'endroit où il faut changer de direction
Signifie	Danger	Ordre Interdiction	Obligation	Indication	Localisation	Direction

Figure 1.2 : Signification de chacune panneaux de signalisation [5].

• **Panneau triangulaire :** C'est un *panneau de danger*. Il prévient de la présence d'un danger potentiel : rétrécissement de la chaussée, passage d'enfants, virage, pentes, chute de pierres, etc. Il est placé à 50 mètres avant le danger en agglomération et à 150 mètres hors agglomération. Le conducteur doit y être très attentif et adapter sa conduite (ralentissement, changement de voie...)(voir la figure 1.3).



**Figure 1.3 :** Panneau triangle.

• **Panneau rond :** ce type de panneaux a pour fonction de donner un ordre, d'interdire ou d'obliger limitation de vitesse, direction à prendre, allumage des feux, etc. Il est placé à l'endroit même où le conducteur doit s'exécuter et adapter sa conduite. Les panneaux ronds à fond blanc bordé de rouge sont des panneaux d'interdiction et les panneaux ronds à fond bleu sont des panneaux d'obligation (voir la figure 1.4).



**Figure 1.4 :** Panneau rond.

• **Panneau de forme carrée :** Il a pour fonction d'indiquer et d'informer présence d'un parking, d'une impasse, d'une piste cyclable, etc. Il est placé soit à l'endroit de la prise de décision, soit avant. Il permet au conducteur d'anticiper ses choix et ses déplacements afin de ne pas gêner la circulation (voir la figure 1.5).



**Figure 1.5 :** Panneau de forme carrée.

• **Panneau en forme de flèche** : Il a pour fonction d'indiquer une direction, il est blanc lorsqu'il indique un lieu-dit et vert lorsqu'il indique une ville. Il peut également être complété de chiffres qui annoncent la distance (voir la figure 1.6).



Figure 1.6 : Panneau en forme de flèche.

• **Panneau rectangulaire** : Il indique une localisation il est bleu pour un lieu tel qu'une aire d'autoroute, marron pour un lieu à visiter, rouge et blanc pour une ville ou un village. Il peut aussi indiquer plusieurs directions et distances, notamment sur les autoroutes (voir la figure 1.17).



Figure 1.1 : Panneau rectangulaire.

### 1.2.3 Cinq grandes familles de panneaux de signalisation routière

Il existe cinq types de panneaux de signalisation routiers, les panneaux de danger, les panneaux d'interdiction, les panneaux d'obligation, les panneaux d'indication et les panneaux de direction (voir la figure 1.8) [6].

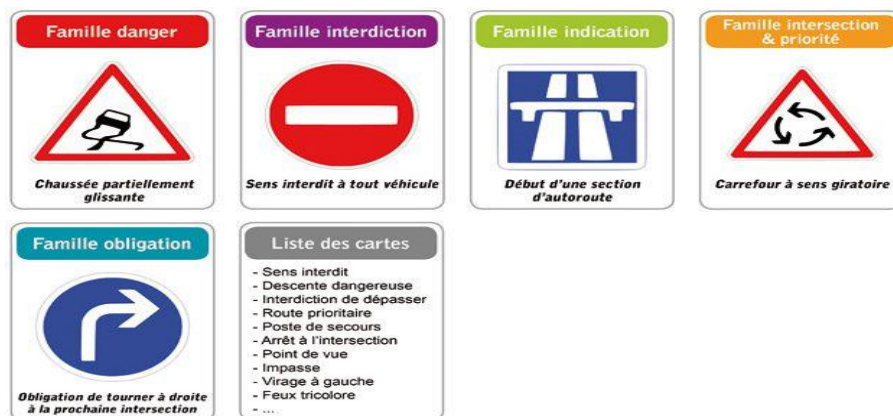


Figure 1.8 : Familles de panneaux de signalisation [6].

- **Panneaux d'interdiction ou de restriction** : Ces types de signaux interdisent ou limitent certaines actions à ceux qui les trouvent devant dans la direction de leur marche et de l'endroit où ils se trouvent. Ces signes sont circulaires et avec un bord rouge. Comme montre la figure suivante(voir la figure 1.9) [7].



Figure 1.2 :Signaux d'interdiction ou de restriction [7].

- **Panneaux d'avertissement de danger** : Les signes de danger ont pour mission d'indiquer la nature d'un danger, leur objectif se conformer aux règles de comportement et d'éviter les chocs éventuels lors de conduite. Sa forme est triangulaire avec un bord rouge, [8] comme indique la figure suivante. (Voir la figure 1.10).



Figure 1.3: Signaux d'avertissement de danger [8].

- **Panneaux de priorité** : Ils sont destinés à informer les usagers de la route aux règles de priorité spéciales aux intersections ou aux passages étroits. À l'intérieur de cette classe, nous pouvons trouver deux des signes les plus importants qui existent *stop* et *Cédez le passage* [9]. Comme vous pouvez le voir (Figure 1.11), ils n'ont pas de formulaire ou une couleur spécifique.

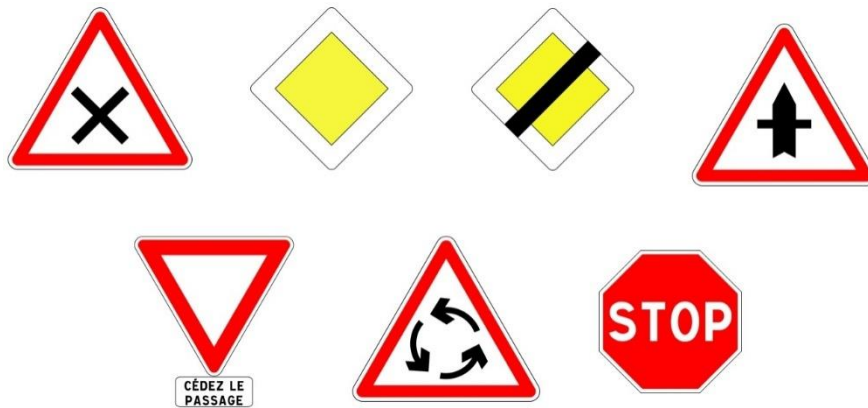


Figure 1.4: Panneaux de priorité [9].

- **Panneaux de fin de l'interdiction ou de la restriction** : Ces panneaux de fin d'interdiction ou de la restriction signalent que l'interdiction ou de la limitation est terminée. Ils sont ronds et blancs avec une ligne diagonale noire (Figure 1.12) [10].

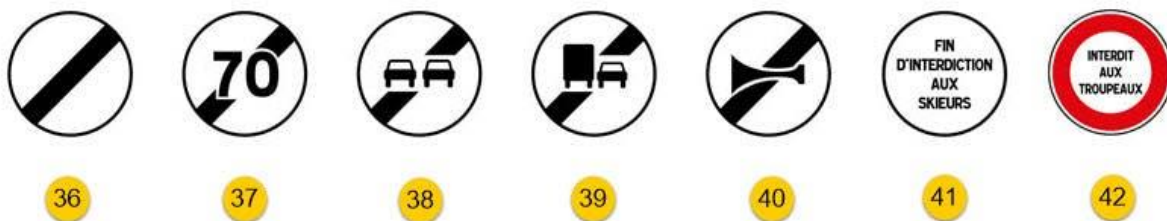


Figure 1.5: Signaux de fin d'interdiction ou de restriction [10].

- **Panneaux des obligations** : Ils sont destinés à obliger les usagers à respecter certaines prescriptions et certaines règles à partir du niveau du panneau ou devant celui-ci. Ces panneaux peuvent donc indiquer non seulement une direction que l'utilisateur devra suivre, mais aussi une vitesse minimum à respecter, ou encore une voirie obligatoire pour certains types d'usagers. Ils sont ronds avec un couleur bleu (voir la figure 1.13) [11].



### 1.2.6 Couleurs des panneaux

- Un fond blanc signifie que l'indication ou la prescription donnée par le panneau est permanente.
- Un fond jaune signifie que l'indication ou la prescription donnée par le panneau est temporaire.
- Un fond bleu peut être synonyme d'obligation ou de simple indication [14].

### 1.2.7 Panneau "P"

Le panneau "P" blanc sur fond bleu est un panneau de signalisation de type C (indication). Il signifie qu'il y a une aire de stationnement autorisé ou un parking à proximité (voir la figure 1.15) [15].

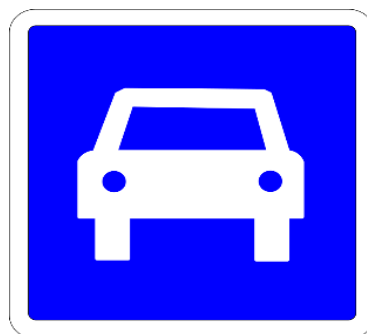


**Figure 1.15:** Le panneau "P" [15].

### 1.2.8 Panneau bleu

Dans le code de la route les panneaux de signalisation carrés bleus sont faits pour les indications. Ces panneaux servent à informer et fournir une indication aux conducteurs.

Par exemple avec ceux que l'on retrouve à l'entrée des parkings (le panneau P) ou des impasses (voir la figure 1.15) [16].






**Figure 1.16 :** Exemple de panneau bleu [16].

## 1.2.9 Panneaux de signalisation routière en Algérie

En Algérie, les panneaux de signalisation routière sont conformes aux normes internationales (notamment la Convention de Vienne sur la signalisation routière), avec quelques particularités locales (langue arabe et parfois française). Ils sont regroupés selon leur fonction :

Tableau 1.1 : Panneaux de signalisation routière en Algérie

Catégorie	Image du panneau	Nom du panneau	Description	Dimension
Panneaux de danger		Passage pour piétons	Signale un lieu fréquenté par les piétons.	<b>Hauteur du triangle</b> : 900 mm (standard) <b>Peut varier</b> : de 600 mm (zones urbaines) à 1000 mm (routes principales)
Panneaux d'interdiction		Accès interdit	Empêche l'accès à tous les véhicules dans cette direction.	<b>Dimensions variables</b> selon le type d'information Exemple : 600 mm × 600 mm ou 900 mm × 600 mm
Panneaux d'obligation		Obligation de tourner à droite	Le conducteur doit impérativement tourner.	<b>Diamètre standard</b> : 600 mm à 900 mm 600 mm : zones urbaines 900 mm : routes nationales et autoroutes

<p><b>Panneaux de direction</b></p>		<p>Direction vers une ville</p>	<p>Indique le nom d'une ville ou d'un axe routier.</p>	<p><b>Largeur :</b> de 1000 mm à 2000 mm</p> <p><b>Hauteur :</b> de 400 mm à 800 mm</p> <p>Dépend du nombre de lignes de texte et du lieu</p>
-------------------------------------	---	---------------------------------	--	---

### 1.3 Traitement d'image

Le traitement d'images est une branche du traitement de signal dédiée aux images et vidéo. Le traitement d'images est l'ensemble des opérations effectuées sur l'image, afin d'en améliorer la lisibilité et d'en faciliter l'interprétation. C'est, par exemple, le cas des opérations de rehaussement de contraste, réduction du bruit et correction d'un flou. C'est aussi l'ensemble d'opérations effectuées pour extraire des "informations" de l'image comme la segmentation et l'extraction de contours. [17]

#### 1.3.1 Définition d'image

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de  $R \times R$  dans  $R$  où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de  $R \times R \times R$  dans  $R$ . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de  $R \times R$  dans  $R \times R \times R$  (voir la figure 1.17) [18].



**Figure 1.17** : Panneau de limitation de vitesse 60 km/h en niveaux de gris – Algérie.

### 1.3.2 Bruit sel et poivre

Le bruit poivre et sel, également appelé bruit impulsif, est une forme de dégradation aléatoire affectant les images numériques. Il se manifeste par l'apparition de pixels noirs et blancs dispersés de manière aléatoire dans l'image, d'où son nom évoquant le poivre (pixels noirs) et le sel (pixels blancs). [19]

Ce type de bruit survient lorsque certains pixels prennent des valeurs extrêmes, soit 0 (noir) soit 255 (blanc) dans une image codée sur 8 bits. Les causes possibles incluent des erreurs de transmission de données, des dysfonctionnements ou la présence de particules fines sur les éléments du capteur de la caméra, ou encore des emplacements mémoire défectueux dans le matériel (voir la figure 1.18).



**Figure 1.18** : Application du Bruit impulsif à une image de panneau de *Rond-point* en Algérie ( $P = 0.05$ ).

### 1.3.3 Bruit gaussien

Ce type de bruit très courant en traitement d'image et en traitement du signal. Il se caractérise par une distribution statistique normale (ou gaussienne), ce qui signifie que les valeurs du bruit suivent une courbe en cloche centrée autour de zéro. Ce bruit apparaît souvent lors de la capture, de la transmission ou de la compression d'images, et se manifeste par des variations aléatoires des intensités des pixels [19].

Le bruit gaussien est généralement dû à des fluctuations électroniques dans les capteurs, à des interférences ou à des perturbations aléatoires dans les systèmes d'acquisition. Sa

présence peut dégrader la qualité visuelle des images et affecter les performances des algorithmes de traitement ou de reconnaissance.

Comprendre et modéliser le bruit gaussien est essentiel pour développer des méthodes efficaces de réduction de bruit, telles que les filtres linéaires ou non linéaires, afin d'améliorer



la qualité des images et la fiabilité des analyses ultérieures (voir la figure 1.19).

**Figure 1.19** : Panneau *Arrêt et stationnement interdit* en Algérie avec ajout de bruit gaussien ( $m = 0, \sigma = 0.15$ ).

### 1.3.4 Bruit speckle

Le bruit speckle est un type de bruit multiplicatif qui apparaît fréquemment dans les images issues de systèmes utilisant des ondes cohérentes, comme les radars, les ultrasons ou certains types d'imagerie laser. Ce bruit se manifeste par une texture granuleuse ou tachetée, due aux interférences constructives et destructives entre les ondes réfléchies par les différentes surfaces de l'objet observé. Le speckle dégrade la qualité des images en introduisant des variations aléatoires de l'intensité, ce qui complique l'analyse et la détection des détails importants. Pour atténuer ce bruit, des techniques spécifiques de filtrage et de traitement d'image sont nécessaires (voir la figure 1.20) [20].



**Figure 1.20** :Panneau de signalisation *Attention rond-point* en Algérie affecté par un bruit speckle.

### 1.3.5 Flou

Le flou est un phénomène fréquent en traitement d'image, qui se manifeste par une perte de netteté et de détails visuels dans une image. Il peut résulter de plusieurs causes, telles qu'un mouvement de la caméra pendant la prise de vue (flou de mouvement), une mauvaise mise au point de l'objectif (flou optique), ou encore des limitations matérielles. Le flou peut affecter la qualité d'interprétation d'une image, notamment dans des applications sensibles comme la surveillance, la médecine ou la vision artificielle. [19]

En traitement d'image, la détection et la suppression du flou (ou "défloutage") sont des étapes importantes dans la restauration d'image, visant à améliorer la lisibilité ou à récupérer les détails perdus. Cela peut être réalisé à l'aide de méthodes mathématiques comme la déconvolution ou à l'aide d'approches modernes basées sur l'intelligence artificielle (voir la figure 1.21).



**Figure 1.21** : Image du panneau *Cédez le passage à l'intersection* en Algérie – flou gaussien appliqué (Taille  $5 \times 5$ ,  $\sigma = 2$ ).

### 1.3.6 Points par pouce (DPI)

Points par pouce (dots per inch), est une unité de mesure utilisée principalement pour quantifier la résolution d'une image ou d'un dispositif d'impression. Il indique combien de points (ou pixels) sont affichés ou imprimés sur une longueur d'un pouce (2,54 cm).

Dans le contexte des écrans, le DPI détermine la densité de pixels, influençant ainsi la netteté et la qualité visuelle des images affichées. Plus le DPI est élevé, plus les images sont détaillées et précises [21].

Dans le domaine de l'impression, le DPI décrit la finesse avec laquelle une imprimante dépose de l'encre sur le papier. Un DPI élevé signifie une meilleure qualité d'impression avec des détails plus fins.

- **La loi du DPI**

La relation fondamentale entre la taille physique, la résolution en pixels et le DPI s'exprime par la formule suivante :

$$\text{Taille en pouces} = \frac{\text{Taille en pixels}}{\text{DPI}}$$

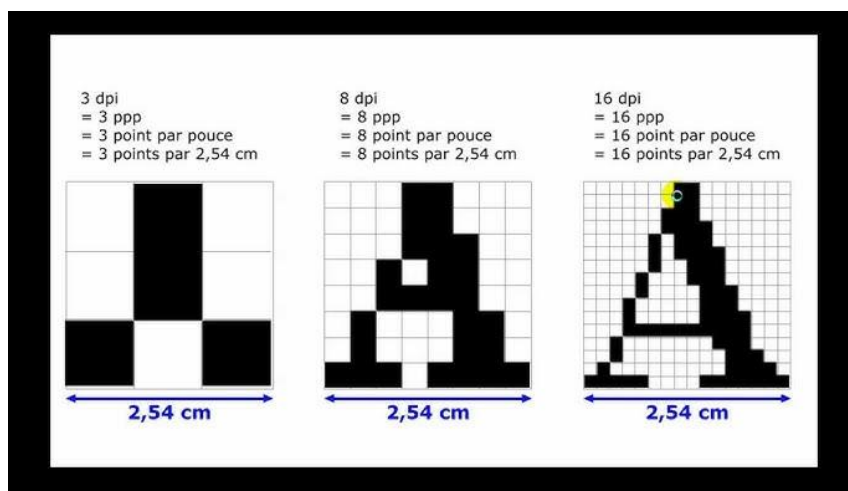
Ou inversement :

$$\text{DPI} = \frac{\text{Taille en pixels}}{\text{Taille en pouces}}$$

Par exemple, une image de 3000 pixels de large affichée ou imprimée avec un DPI de 300 aura une largeur physique de :

$$\frac{3000}{300} = 10 \text{ pouces} \approx 25,4 \text{ cm}$$

En résumé, Le DPI est un indicateur clé de la qualité visuelle, que ce soit à l'écran ou sur support papier. Il permet d'ajuster la densité de pixels par unité de longueur, influençant directement le rendu final d'une image. Une bonne maîtrise du DPI est essentielle en



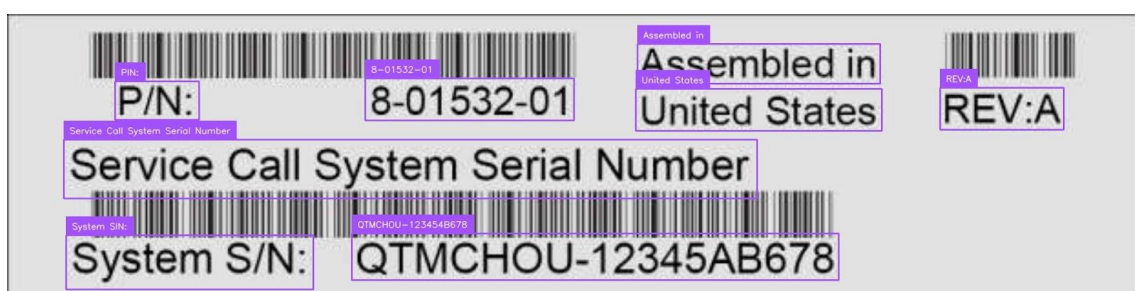
infographie, en impression, et dans tout domaine lié à la production visuelle (voir la figure 1.22).

**Figure 1.22** : La Résolution des Images (le rôle de PDI) [8].

### 1.3.7 Reconnaissance Optique de Caractères (OCR)

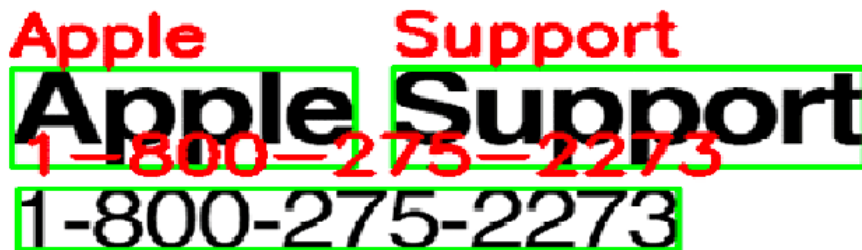
L'OCR, ou Reconnaissance Optique de Caractères, est une technologie avancée qui permet de convertir automatiquement du texte contenu dans une image — comme une photo, un scan de document, une page imprimée ou manuscrite — en texte numérique exploitable par un ordinateur. Le processus commence par l'analyse de l'image pour repérer les zones contenant du texte, puis utilise des algorithmes d'intelligence artificielle et de reconnaissance de formes pour identifier chaque caractère, mot ou ligne. Ce texte ainsi extrait peut ensuite être copié, modifié, indexé, traduit ou stocké dans une base de données. L'OCR joue un rôle clé dans la numérisation et la gestion électronique de documents (GED), permettant de transformer des archives papier en fichiers numériques facilement consultables. Elle est aussi largement utilisée dans la lecture automatique de plaques d'immatriculation pour les systèmes de surveillance ou de péage, dans l'extraction d'informations à partir de formulaires, dans les applications mobiles de traduction en temps réel, ou encore dans les outils d'accessibilité pour les personnes malvoyantes. Grâce à l'OCR, les images deviennent intelligibles pour les machines, ouvrant la voie à une automatisation efficace du traitement de texte dans de nombreux secteurs comme l'administration, la santé, la finance ou encore la logistique [23].

- **EasyOCR** : est une bibliothèque Python open source qui permet de réaliser de la reconnaissance optique de caractères (OCR) facilement et efficacement. Elle utilise des techniques avancées d'apprentissage profond (deep learning) pour détecter et extraire automatiquement le texte présent dans des images, même dans des conditions complexes comme des polices variées, des arrière-plans compliqués ou plusieurs langues. Facile à installer et à utiliser, EasyOCR supporte plus de 80 langues, ce qui en fait un outil polyvalent pour des applications allant de la numérisation de documents à la lecture automatique de panneaux, tickets, ou toute autre image contenant du texte. Grâce à sa simplicité d'usage et ses bonnes performances, EasyOCR est très apprécié dans la communauté des développeurs et chercheurs travaillant sur la vision par ordinateur et le traitement d'images (voir la figure 1.23) [24].



**Figure 1.23** : Extraction automatique de texte d'une image avec EasyOCR [25].

- **Tesseract OCR** : est un moteur de reconnaissance optique de caractères open source très populaire, initialement développé par Hewlett-Packard dans les années 1980, puis repris et amélioré par Google. Cette technologie permet de convertir automatiquement le texte présent dans des images telles que des scans, des photos de documents, des pages imprimées, ou même des captures d'écran en texte numérique éditable. Grâce à sa capacité à reconnaître de nombreuses langues et alphabets, Tesseract s'adapte à une large variété d'applications, que ce soit pour la numérisation de documents administratifs, la lecture automatique de plaques d'immatriculation, l'extraction de texte sur des formulaires, ou la reconnaissance de texte dans des images complexes. Sa précision dépend souvent de la qualité de l'image et d'un prétraitement efficace, comme la conversion en niveaux de gris ou le nettoyage du bruit. En plus d'être puissant et précis, Tesseract est apprécié pour sa flexibilité, puisqu'il peut être intégré facilement dans des projets Python, C++ ou autres, ce qui en fait une solution incontournable pour les développeurs et chercheurs travaillant dans le domaine du traitement d'images et de la vision par ordinateur. Enfin, en étant open source, il bénéficie d'une large communauté qui contribue à son amélioration continue, assurant ainsi sa pérennité et son



évolution constante (voir la figure 1.24) [26].

**Figure 1.24** : Reconnaissance de texte automatique avec TesseractOCR [27].

## 1.4 Conclusion

La signalisation routière constitue un élément fondamental pour garantir à la fois la sécurité et la fluidité du trafic. Son interprétation rapide et fiable est indispensable, particulièrement dans le contexte actuel où les systèmes automatisés et les véhicules autonomes prennent une place croissante. Grâce aux progrès considérables réalisés dans le domaine du traitement d'image et de l'intelligence artificielle, il est désormais possible de concevoir des systèmes capables de détecter, reconnaître et interpréter automatiquement les panneaux de signalisation à partir de données visuelles. Ces technologies ne se limitent plus à de simples aides à la conduite, elles ouvrent la voie à des systèmes de transport intelligents, plus sûrs et plus performants, adaptés aux enjeux modernes de mobilité.

Les avancées en traitement d'image permettent de surmonter de nombreux défis, tels que les variations d'éclairage, les conditions météorologiques, ou encore la diversité des panneaux selon les pays et les régions. En combinant des algorithmes sophistiqués de reconnaissance visuelle avec des modèles d'apprentissage profond, ces systèmes automatisés deviennent de plus en plus précis et fiables, renforçant la sécurité routière et améliorant l'expérience de conduite.

Le chapitre suivant se concentrera sur les bases théoriques du traitement d'image, en détaillant les principales techniques et méthodes utilisées pour la reconnaissance des panneaux de signalisation. Il présentera également les algorithmes et outils qui permettent de mettre en œuvre ces systèmes innovants, offrant ainsi un aperçu complet des fondements nécessaires à leur développement.

---

## ***Chapitre 02 :***

### ***Traitement d'images et Apprentissage profond***

---

## 2.1 Introduction

La détection d'objets est l'une des applications les plus concrètes et utiles de l'intelligence artificielle dans notre quotidien. Elle joue un rôle essentiel dans des domaines variés tels que la surveillance routière, les systèmes de sécurité, ou encore les véhicules dotés d'une assistance à la conduite automatique ou semi-autonome. Ce chapitre propose une exploration approfondie des avancées récentes dans le domaine de la reconnaissance des panneaux de signalisation. Il met en lumière la synergie entre les techniques de traitement d'image et les approches modernes d'apprentissage profond, qui permettent d'améliorer la précision et la robustesse des systèmes de détection. Une attention particulière est portée au modèle YOLOv11, qui illustre l'évolution continue des architectures dédiées à la vision par ordinateur et leur efficacité face aux défis complexes posés par les environnements routiers réels.

## 2.2 Introduction au traitement d'image

Le traitement d'image constitue un domaine fondamental de la vision par ordinateur, visant à analyser, interpréter et transformer des images numériques afin d'en extraire des informations utiles. Grâce aux avancées technologiques récentes, notamment en apprentissage automatique et en intelligence artificielle, ce domaine a connu une évolution remarquable, facilitant l'automatisation de tâches complexes autrefois réservées à l'œil humain.

L'objectif du traitement d'image est de permettre à un système informatique de "comprendre" une image : cela peut aller de simples opérations comme le filtrage et l'amélioration du contraste, jusqu'à des tâches plus complexes comme la détection d'objets, la reconnaissance de formes ou l'identification de caractéristiques spécifiques dans un environnement visuel.

Aujourd'hui, le traitement d'image est au cœur de nombreuses applications concrètes, telles que : la surveillance intelligente, diagnostic médical assisté par ordinateur, systèmes de conduite autonome, ou encore reconnaissance de panneaux de signalisation. Grâce à des algorithmes d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN), et à des architectures performantes comme YOLO, il est désormais possible de traiter des images en temps réel avec un niveau élevé de précision et de fiabilité.

### 2.3 Vision par ordinateur

La vision par ordinateur est un domaine de l'intelligence artificielle qui vise à permettre aux machines de comprendre et d'analyser le contenu visuel d'images ou de vidéos. En s'appuyant sur des algorithmes avancés, comme les réseaux de neurones convolutifs (CNN) ou YOLO, elle rend possible la détection, la classification et la segmentation d'objets en temps réel, ouvrant la voie à de nombreuses applications industrielles et innovantes.

- **Détection** : La détection est également la tâche principale prise en charge par YOLOv11, tout en intégrant des fonctionnalités avancées. En plus de détecter des objets dans des images ou des vidéos avec des cadres de délimitation. Les objets détectés sont automatiquement classés selon leurs catégories respectives, offrant une analyse précise et complète adaptée aux applications en temps réel et aux besoins industriels.
- **Classification** : La classification consiste à attribuer une catégorie à une image en fonction de son contenu. YOLOv11 permet d'effectuer cette tâche de manière rapide et précise, en s'appuyant sur une architecture optimisée inspirée des versions précédentes. Grâce à son efficacité, il est possible de classer de grandes quantités d'images avec de hautes performances, tout en conservant la légèreté et la rapidité qui caractérisent la famille des modèles YOLO.
- **Segmentation** : La segmentation consiste à diviser une image en plusieurs régions distinctes selon les objets qu'elle contient. YOLOv11 prend en charge cette tâche en attribuant à chaque région une étiquette précise, permettant d'isoler chaque objet de manière pixel par pixel. YOLOv11 utilise une architecture optimisée intégrant des éléments avancés inspirés de U-Net pour offrir une segmentation rapide et précise, tout en restant adaptée aux applications en temps réel.

### 2.4 L'Apprentissage profond

L'apprentissage profond a marqué un tournant décisif dans le domaine de la vision par ordinateur à partir de 2012, lorsque l'équipe dirigée par Geoffrey Hinton a remporté le prestigieux concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [28]. En utilisant un réseau de neurones profond, leur solution a nettement surpassé les performances des approches traditionnelles, révélant ainsi le potentiel exceptionnel de cette technologie.

Depuis cet événement fondateur, l'apprentissage profond s'est imposé comme un pilier majeur de la vision artificielle. Il a donné naissance à une nouvelle vague d'innovations, avec l'émergence de nombreuses startups et projets industriels exploitant ses performances pour

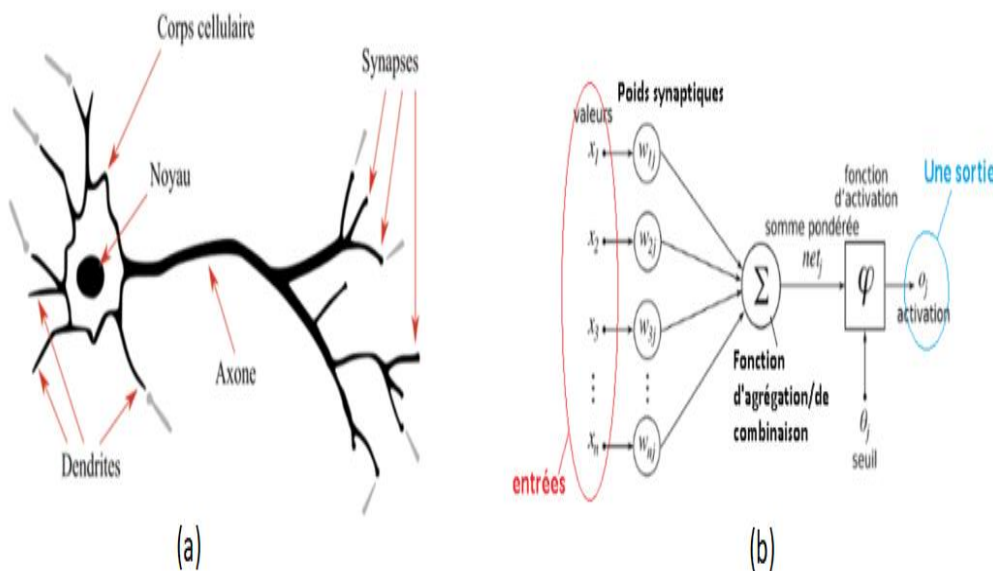
des applications variées : reconnaissance faciale [29], conduite autonome [30], diagnostic médical [31], etc.

L'importance de cette avancée a été largement reconnue : en 2013, le MIT Technology Review a classé l'apprentissage profond parmi les dix technologies les plus innovantes de l'année, soulignant son impact croissant sur l'industrie et la recherche scientifique.

### 2.4.1 Réseaux neuronaux artificiels

Les réseaux neuronaux artificiels (RNA), ou Artificial Neural Networks (ANN), sont des modèles d'apprentissage automatique inspirés du fonctionnement du cerveau humain. Dans le système nerveux biologique, les neurones communiquent entre eux par l'intermédiaire de synapses. De manière analogue, les RNA reproduisent ce principe en utilisant des unités de calcul appelées neurones artificiels, interconnectés par des poids [32].

Ces poids représentent la force ou l'importance des connexions entre les neurones, et jouent un rôle central dans l'apprentissage. En effet, ils sont ajustés automatiquement lors de l'entraînement du réseau, en fonction des erreurs commises, afin d'améliorer progressivement les performances du modèle (Voir figure 2.1).



**Figure 2.1** : Neurones : (a) biologique, (b) artificiel [33].

### 2.4.2 Architecture de base du perceptron

Le perceptron appelé aussi réseau monocouche, est un modèle simple de neurone artificiel introduit par Frank Rosenblatt [34]. Il constitue la base des réseaux de neurones. Ce modèle prend plusieurs entrées numériques, chacune multipliée par un poids, puis additionne

ces valeurs pondérées. Le résultat est ensuite transmis à une fonction d'activation, généralement une fonction seuil (voir la figure) [35].

Si la somme obtenue dépasse un certain seuil, le perceptron produit une sortie de 1 ; sinon, il renvoie 0. Il fonctionne donc comme un classificateur binaire, capable de séparer deux classes de manière linéaire.

Lors de l'apprentissage, les poids sont ajustés automatiquement pour réduire l'écart entre les sorties produites et les sorties attendues. Bien que limité aux problèmes linéaires, le perceptron a jeté les bases des modèles plus complexes, comme les réseaux multicouches.

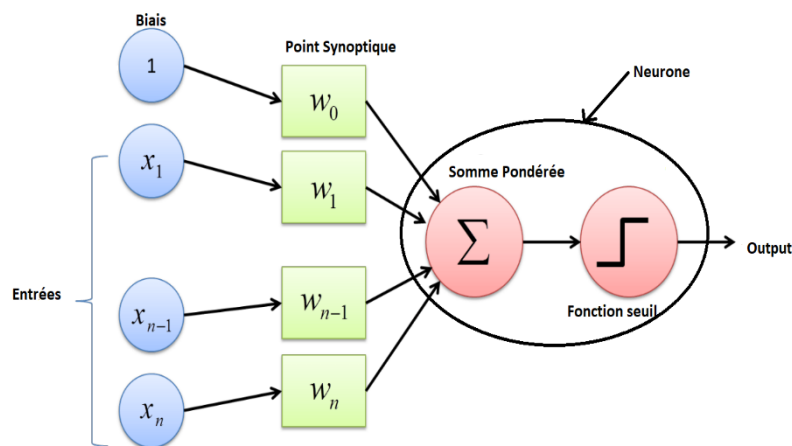


Figure 2.2 : Architecture de base d'un perceptron [36].

### 2.4.3 Architecture du réseau neuronal

L'architecture d'un réseau de neurones désigne la manière dont les neurones sont organisés et connectés entre eux à l'intérieur du réseau. Elle définit la structure globale du modèle, c'est-à-dire le nombre de couches, le type de couches (entrée, cachées, sortie), le nombre de neurones par couche, et la nature des connexions entre les neurones.

La plupart des réseaux de neurones adoptent des structures homogènes, utilisant des neurones identiques avec des fonctions d'activation similaires. Cependant, certaines architectures plus spécialisées peuvent intégrer des types de neurones distincts, chacun dédié à une sous-tâche précise.

Le choix de l'architecture dépend étroitement du problème à traiter : classification, régression, reconnaissance d'images, traitement de texte, etc. Par exemple, les réseaux convolutifs (CNN) sont bien adaptés à l'analyse d'images, tandis que les réseaux récurrents (RNN) sont conçus pour les données séquentielles.

En général, un réseau de neurones est composé de plusieurs couches : une couche d'entrée qui reçoit les données brutes, une ou plusieurs couches cachées qui réalisent des traitements complexes et abstraits, et enfin une couche de sortie qui produit la prédiction finale. La performance du réseau dépend en grande partie de la manière dont cette architecture est conçue et optimisée (voir la figure 2.3) [37].

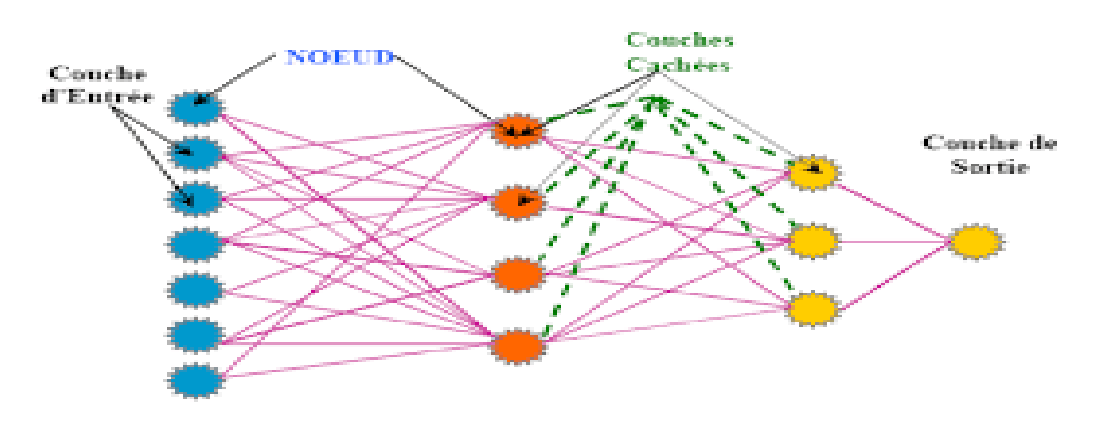


Figure 2.3: Architecture d'un réseau de neurones artificiels [38].

- **Entrées  $x_1$  à  $x_n$**  : Ce sont les valeurs ou caractéristiques extraites de l'ensemble de données. Chaque entrée comprend  $n$  variables dépendantes, qui fournissent des informations cruciales pour le modèle.

- **Poids Synoptiques  $w_1$  à  $w_n$**  : Ce sont les paramètres associés à chaque variable d'entrée. Les poids déterminent l'importance relative de chaque variable dans le calcul du modèle et influencent la contribution de chaque entrée au résultat final.

- **Biais ( $b$ )** : Il s'agit du terme constant ajouté au modèle, souvent appelé biais. Ce terme permet d'introduire un degré de flexibilité supplémentaire dans le modèle, en tenant compte de variations ou ajustements supplémentaires comme le flou ou le lissage gaussien sur les variables d'entrée.

#### 2.4.4 Structure du réseau neuronal

Les réseaux neuronaux artificiels (RNA) sont composés de neurones artificiels. Le nombre de neurones dans un réseau peut varier en fonction des exigences spécifiques de l'application. Ces neurones sont organisés en couches. La structure la plus courante d'un réseau inclut une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Contrairement aux neurones biologiques, qui communiquent par des impulsions électriques, les neurones d'un ANN transmettent l'information à travers les couches et les nœuds du réseau. Au niveau des nœuds, l'importance du signal d'entrée est déterminée par des poids

associés à chaque connexion. La valeur de ces poids peut être positive ou négative. Un neurone devient actif si le poids est positif, tandis qu'il reste inactif si le poids est négatif. Chaque neurone additionne les entrées reçues et les multiplie par les poids correspondants avant de passer à l'étape suivante du calcul (voir la figure 2.4) [39].

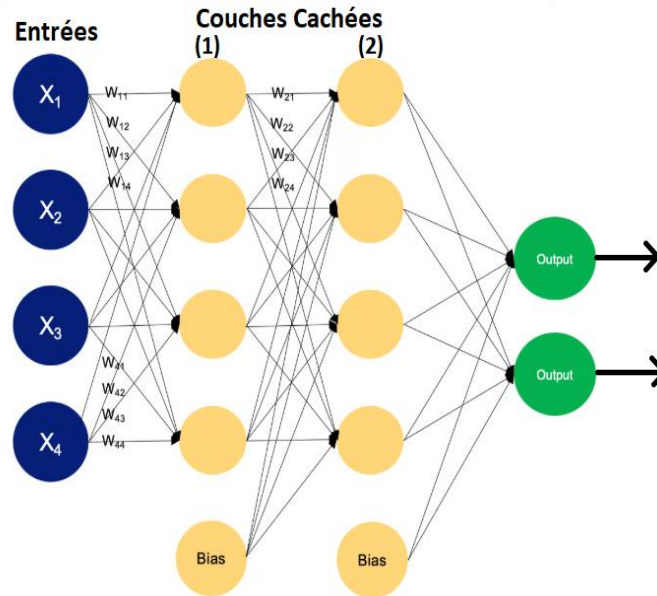


Figure 2.4: Perceptrons multicouches [40].

### 2.4.5 Réseaux neuronaux convolutifs (CNN)

Les réseaux neuronaux convolutifs (CNN) sont inspirés des mécanismes biologiques et sont largement utilisés en vision par ordinateur pour des tâches telles que la classification d'images et la détection d'objets. Les CNN possèdent une architecture tridimensionnelle, où l'étendue spatiale et la profondeur correspondent respectivement aux caractéristiques et aux canaux de couleur des données. Ils se composent de couches de convolution et de couches de sous-échantillonnage. Les couches de convolution appliquent des filtres pour mapper les activations d'une couche à l'autre, permettant de capturer les relations spatiales. Les connexions dans un CNN sont relativement peu nombreuses, les couches de niveau inférieur étant responsables de la détection de formes simples, tandis que les couches de niveau supérieur capturent des structures plus complexes. Les couches de sous-échantillonnage, quant à elles, réduisent la taille des empreintes spatiales. Les CNN ont démontré d'excellentes performances dans la reconnaissance d'images et la détection d'objets, [41] souvent surpassant les capacités humaines dans certaines situations.

- **Architecture des Réseaux Neuronaux Convolutifs** : L'élément central d'un réseau de neurones convolutifs (CNN) est la couche convolutive, comme illustré dans la figure 2.5.

Cette couche applique une série de filtres (ou noyaux) à l'image d'entrée afin d'en extraire des caractéristiques importantes. Un filtre est une petite matrice (souvent de taille 3x3, 5x5, etc.) qui se déplace (ou convolue) sur toute l'image, en multipliant ses valeurs avec celles de la portion correspondante de l'image. Le résultat de cette opération produit une carte de caractéristiques (ou feature map), qui représente la réponse du filtre à différentes parties de l'image (voir la figure 2.6).

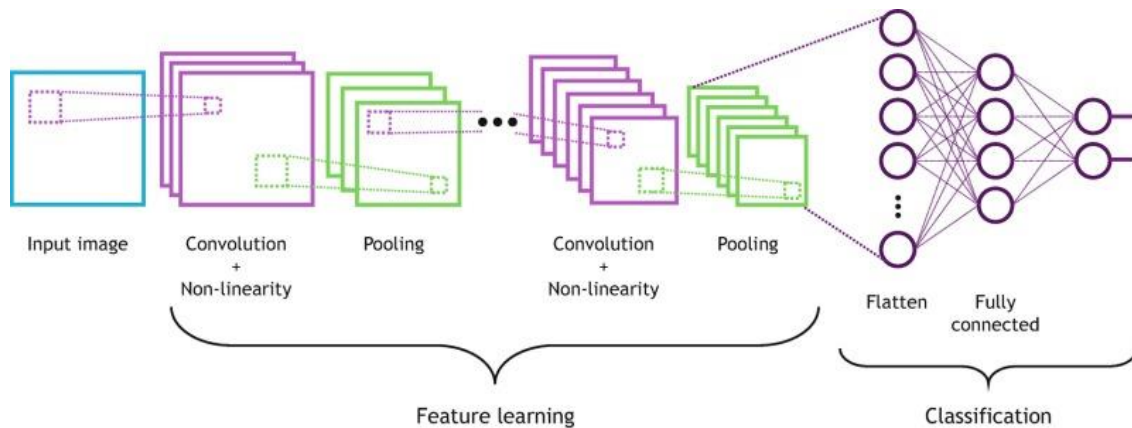


Figure 2.5 : Architecture CNN [42].

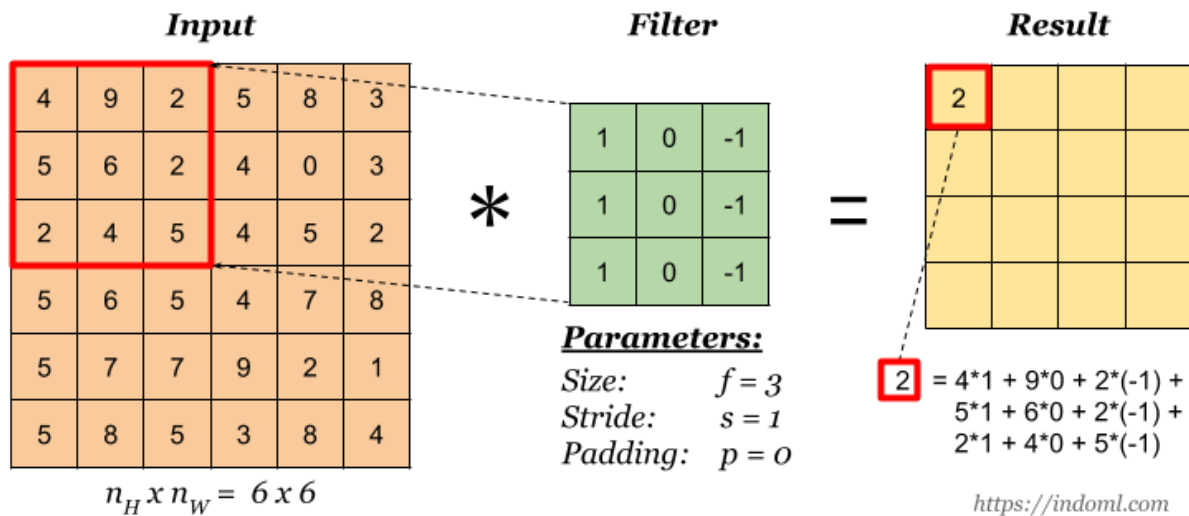


Figure 2.6: Exemple d'une convolution avec filtre de taille (3x3) [43].

Chaque filtre est appris automatiquement pendant l'entraînement et est spécialisé pour détecter certains motifs visuels : par exemple, les premiers filtres peuvent détecter des bords horizontaux ou verticaux, des textures simples, ou des changements d'intensité, tandis que les couches plus profondes peuvent reconnaître des formes complexes ou des structures spécifiques comme des yeux, des roues ou des panneaux.

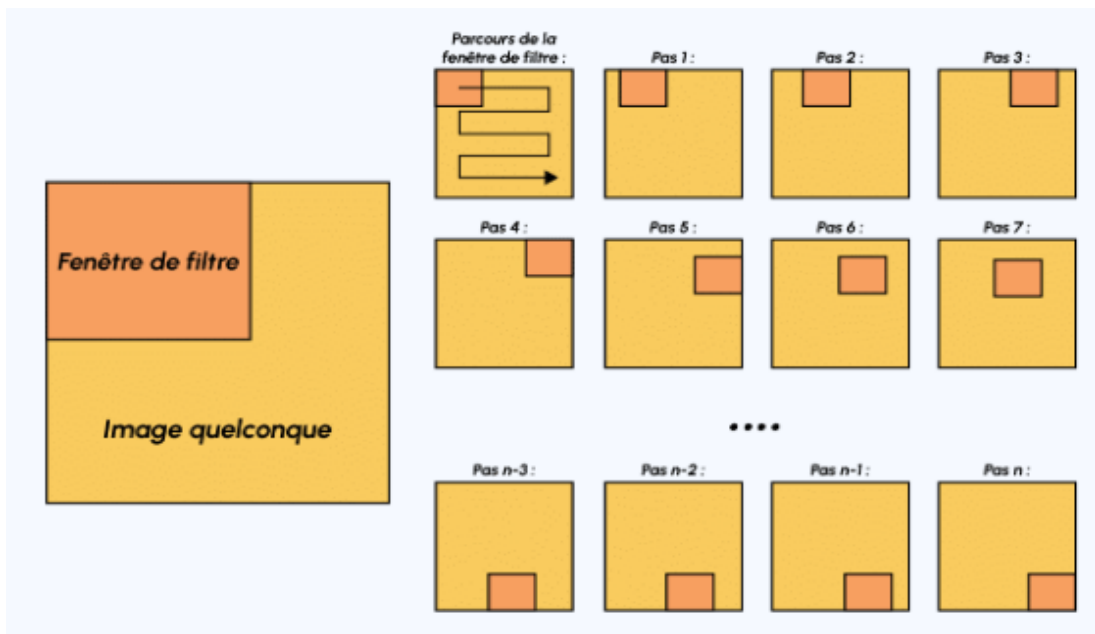
En utilisant plusieurs filtres à chaque couche convolutive, le CNN parvient à extraire une grande variété de caractéristiques visuelles à différents niveaux de complexité, ce qui

rend ces réseaux particulièrement efficaces pour des tâches de reconnaissance d'image, comme la détection de panneaux de signalisation (voir la figure 2.6) [44].

### -Eléments de base d'un CNN :

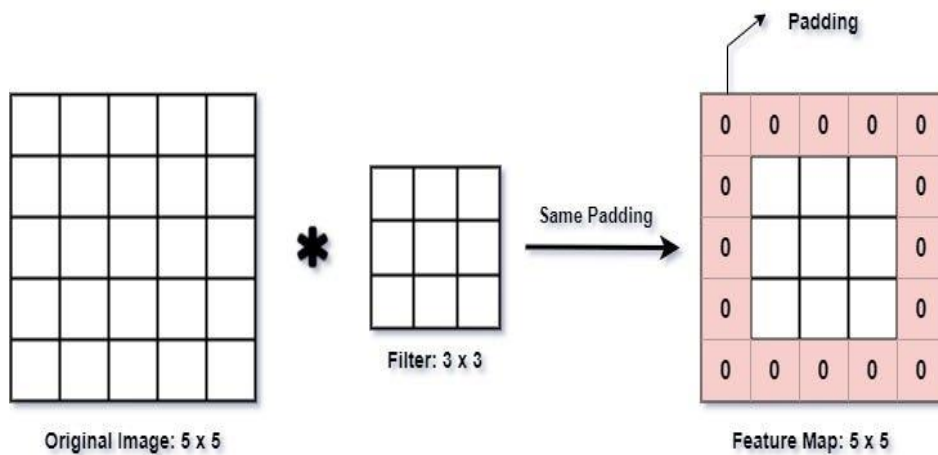
**1- Couche de convolution (CONV) :** Cette couche est utilisée pour extraire des cartes de caractéristiques (*feature maps*) à partir de l'image d'entrée. Son fonctionnement repose sur trois hyperparamètres clés :

- **Profondeur de la couche :** Correspond au nombre de filtres utilisés, chaque filtre détectant des motifs spécifiques (bordures, textures, formes). Par exemple, 64 filtres génèrent 64 cartes de caractéristiques distinctes.
- **Pas (stride) :** ce paramètre régule le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent, ce qui entraîne un volume de sortie plus grand (voir la figure 2.7).



**Figure 2.7 :** Schéma du parcours de la fenêtre de filtre sur l'image [45].

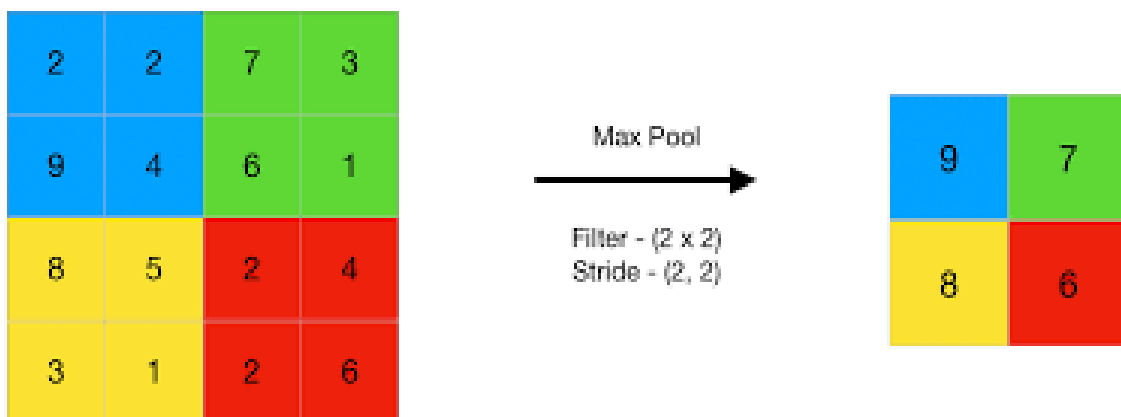
- **Le zero padding ou marge à zéro :** consiste à ajouter des zéros autour des bords du volume d'entrée. La taille de cette marge est le troisième hyperparamètre. Cette opération permet de contrôler la dimension spatiale du volume de sortie, et dans certains cas, il peut être souhaitable de conserver la même taille que celle du volume d'entrée (voir la figure 2.8).



**Figure 2.8:** Zero Padding [46].

2- **Couche de pooling** :réduit la dimensionnalité des cartes de caractéristiques, tout en gardant les informations les plus importantes.

- **Max pooling** : La couche de Max Pooling est utilisée pour réduire la dimension spatiale (hauteur et largeur) des cartes de caractéristiques tout en conservant les informations les plus importantes (voir la figure 2.9). Elle divise l'image en petites régions (par exemple, des blocs de 2×2 pixels) et conserve seulement la valeur maximale de chaque région. Cela permet de réduire le nombre de paramètres, d'accélérer le calcul, de limiter le surapprentissage et d'introduire une forme d'invariance aux translations (déplacements mineurs dans l'image) [47].



**Figure 2.9 :** Max Pooling [48].

- **Average pooling** : La couche Average Pooling fonctionne comme Max Pooling, mais au lieu de prendre la valeur maximale dans une région, elle calcule la moyenne des valeurs. Cela donne une représentation plus lisse et globale des caractéristiques, en conservant une partie de toutes les informations locales. Elle est moins agressive que Max Pooling, mais aussi moins efficace pour capturer les éléments dominants. Elle est

parfois utilisée dans les couches profondes des CNN, ou en combinaison avec d'autres méthodes (voir la figure 2.10).



Figure 2.10 : Average pool [49].

**3- Couche entièrement connectée :** La couche entièrement connectée relie toutes les caractéristiques extraites par les couches précédentes. Elle transforme le vecteur aplati en un ensemble de probabilités pour chaque classe grâce à une combinaison linéaire suivie d'une activation (voir la figure 2.11). Cette étape permet au réseau de réaliser un raisonnement global et de prendre une décision finale. Elle est donnée par :

$$y = f(Wx + b)$$

Avec :  $W$  est la matrice de poids,  $x$  l'entrée,  $b$  le biais, et  $f$  la fonction d'activation.

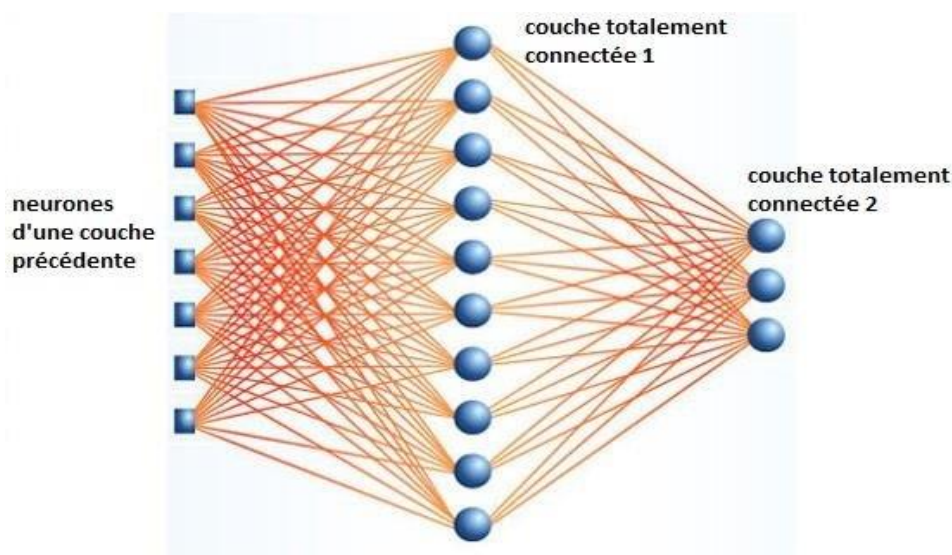


Figure 2.11: La couche entièrement connectée (Fully Connected Layer, FCL) [50].

**4- Fonction d'activation :** Dans un réseau de neurones, la fonction d'activation joue un rôle essentiel : elle introduit de la non-linéarité dans le modèle, [51] ce qui permet au réseau d'apprendre des relations complexes entre les données. Sans fonction d'activation,

un réseau multicouche se comporterait comme un simple modèle linéaire, incapable de modéliser des phénomènes complexes.

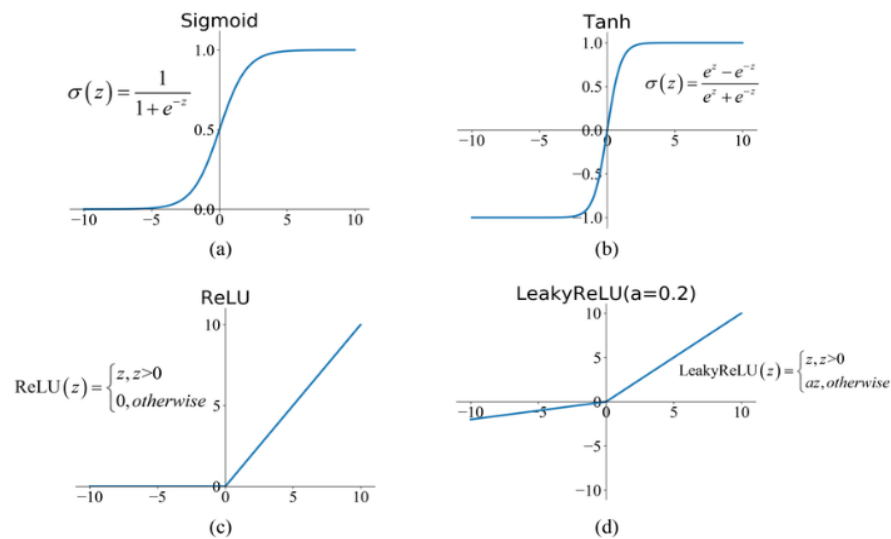
Lorsqu'un neurone reçoit des signaux d'entrée pondérés, il en calcule la somme, puis applique une fonction d'activation à ce total pour produire une sortie. Cette sortie est ensuite transmise aux neurones suivants. Plusieurs fonctions d'activation sont couramment utilisées selon les cas, parmi lesquelles :

**a. La fonction sigmoïde :** est fréquemment utilisée dans les couches de sortie pour les tâches de classification binaire, car elle transforme toute valeur réelle en une sortie comprise entre 0 et 1, ce qui peut être interprété comme une probabilité. [52] Cependant, cette fonction présente un inconvénient majeur : elle tend à ralentir l'apprentissage lorsque les valeurs en entrée sont très élevées ou très faibles. En effet, aux extrémités de sa courbe, la pente devient très faible (voir la figure 2.12 (a)).

**b. La fonction d'activation tanh :** est une amélioration de la fonction sigmoïde, offrant une meilleure non-linéarité. Sa sortie est comprise entre -1 et 1, représentant une version mise à l'échelle de la sigmoïde. Elle peut généralement remplacer la sigmoïde, sauf dans les cas de classification binaire, où des sorties spécifiques de 0 ou 1 sont requises [53] (voir la figure 2.12 (b)).

**c. ReLU (Rectified Linear Unit) :** c'est la fonction d'activation la plus utilisée car elle offre de meilleurs gradients, même pour de grandes valeurs de l'entrée. [54] Cela signifie qu'elle ne souffre pas du problème de ralentissement de l'apprentissage rencontré par sigmoïde et tanh [53] (voir la figure 2.12 (c)).

**d. Leaky ReLU :** est une extension de ReLU qui traite le problème des gradients nuls pour les valeurs d'entrée négatives. Elle introduit une petite pente non nulle pour les valeurs négatives, bien qu'en pratique son impact sur les performances ne soit pas significatif. [55] Par conséquent, ReLU reste la fonction d'activation préférée dans de nombreuses applications (voir la figure 2.12 (d)).



**Figure 2.12 :** Fonction d'activation (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LeakyReLU [56].

- 5- Flattening :** La couche Flatten transforme une entrée multidimensionnelle (par exemple, une image en 2D ou un tenseur 3D issu de convolutions) en un vecteur 1D. Cette étape est nécessaire pour connecter les résultats des couches convolutionnelles et de pooling aux couches entièrement connectées qui suivent. Elle permet donc la transition entre l'extraction de caractéristiques et la classification.
- 6- Softmax :** La fonction Softmax est utilisée en dernière couche d'un réseau pour les tâches de classification multiclasse. Elle convertit un vecteur de scores (ou logits) en un vecteur de probabilités, où chaque valeur représente la probabilité d'appartenance à une classe. Toutes les probabilités sont comprises entre 0 et 1 et leur somme est égale à 1. Le modèle choisit généralement la classe ayant la probabilité la plus élevée.

## 2.5 Détection d'objets à l'aide de YOLO [You Only Look Once]

YOLO (You Only Look Once) est un réseau de neurones spécialisé dans la détection d'objets en temps réel. Comme son nom l'indique, il analyse l'image dans son ensemble en une seule passe à travers le réseau, ce qui lui confère une rapidité remarquable. Cette architecture permet de traiter jusqu'à 30 images par seconde (IPS), rendant YOLO particulièrement adapté aux applications nécessitant une réactivité immédiate, telles que les véhicules autonomes ou la réalité augmentée [57].

Contrairement à d'autres méthodes traditionnelles qui analysent l'image par fragments ou par régions successives, YOLO adopte une stratégie globale : l'image est divisée en une grille, et chaque cellule de cette grille est responsable de prédire des objets avec leurs coordonnées et leurs probabilités d'appartenance à une classe. Grâce à cette approche unifiée

et optimisée, YOLO est capable de détecter plusieurs objets simultanément, avec une grande rapidité et une précision satisfaisante.

C'est cette combinaison de vitesse, simplicité et efficacité qui a rendu YOLO si populaire dans le domaine de la vision par ordinateur (voir la figure 2.13).

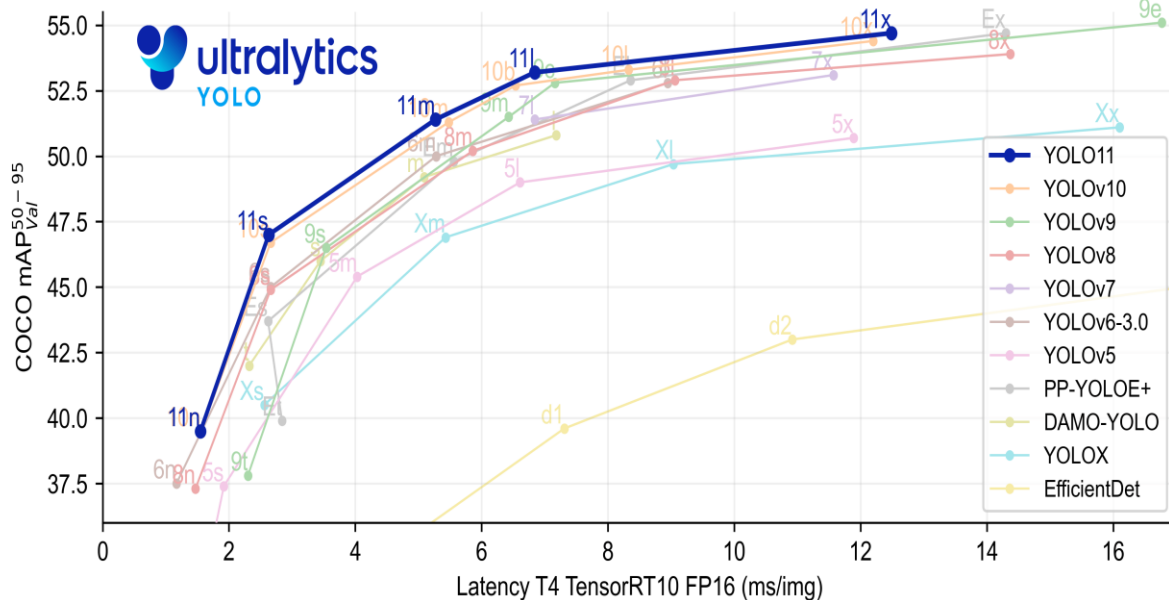


Figure 2.13 : YOLOv11 [58].

### 2.5.1 Algorithme YOLO

YOLO (You Only Look Once) est un détecteur d'objets monocoup (single-stage), c'est-à-dire qu'il effectue la localisation (bounding boxes) et la classification des objets en un seul passage dans le réseau de neurones. Bien que d'autres détecteurs monocoup existent (comme MobileNetSSDv2), YOLO est souvent plus performant en termes de précision et de rapidité.

En formulant la détection comme une tâche de régression directe des boîtes englobantes et des classes, YOLO atteint des vitesses d'inférence élevées avec des modèles souvent compacts, ce qui le rend adapté à l'entraînement rapide et au déploiement sur des dispositifs embarqués à faible puissance de calcul [59].

**-YOLOv1 (2016)** : La première version de YOLO transforme la détection d'objets en un seul passage à travers un réseau. L'image est divisée en une grille, et chaque cellule prédit des boîtes et des classes. C'est rapide mais pas très précis pour les petits objets [60].

**-YOLOv2 (YOLO9000)** : YOLOv2 améliore la précision avec l'ajout des anchor boxes, de la batch normalization, et d'un entraînement multi-échelle. Il introduit aussi YOLO9000, capable

de détecter plus de 9000 classes grâce à un entraînement combiné sur plusieurs bases de données [61].

**-YOLOv3** : Cette version ajoute un backbone plus profond (Darknet-53) et des prédictions à trois échelles différentes. Cela permet de mieux détecter les objets de tailles variées. Elle est plus précise, surtout pour les petits objets [62].

**-YOLOv4** : YOLOv4 intègre de nombreuses techniques modernes comme Mish activation, Cross-stage partial connections (CSP), et des améliorations d'entraînement pour booster les performances sur GPU. Il offre un excellent compromis entre vitesse et précision [63].

**-YOLOv5 (2020)** : Développé par Ultralytics, YOLOv5 n'est pas officiel (pas de Redmon), mais devient très populaire. Il est écrit en PyTorch, facile à utiliser, rapide à entraîner, et propose différentes tailles de modèles (v5s, v5m, v5l, v5x) [64].

**-YOLOv6** : Lancé par Meituan, YOLOv6 cible les applications industrielles. Il utilise un backbone plus efficace et des techniques d'optimisation poussées. Il se concentre sur la rapidité et la précision pour l'inférence en temps réel [65].

**-YOLOv7** : YOLOv7 améliore l'architecture pour intégrer de nouvelles stratégies de formation comme auxiliary heads, coarse-to-fine predictions, et un meilleur équilibre entre légèreté et performance. C'est l'un des plus rapides et puissants à son époque [66].

**-YOLOv8 (2023)** : Toujours développé par Ultralytics, YOLOv8 abandonne l'architecture Darknet au profit d'un modèle plus moderne et modulaire. Il intègre la segmentation, la détection et la classification. Il est très facile à utiliser avec une API claire [67].

**-YOLOv9 (2024)** : YOLOv9 introduit une architecture nommée RT-DETR, inspirée des Transformers mais optimisée pour la vitesse. Il offre de très bonnes performances sans sacrifier la vitesse, notamment pour les applications en edge computing [68].

**-YOLOv10 (2025)** : YOLOv10 combine la rapidité des anciens YOLO avec la précision des Transformers. Il unifie la détection, la segmentation et la pose estimation dans un seul modèle. Il est pensé pour fonctionner sur tous les appareils, du cloud au mobile [69].

**-YOLOv11 (2025)** : YOLOv11 représente l'aboutissement des avancées en matière de détection d'objets en temps réel, combinant la rapidité historique de la série YOLO avec les dernières innovations en intelligence artificielle. Cette version intègre des architectures hybrides mêlant CNN, Transformers et mécanismes d'attention dynamiques, permettant une précision accrue même sur les objets petits ou partiellement occultés. YOLOv11 se distingue



globale du modèle tout en préservant sa capacité à capturer des caractéristiques essentielles.

- **SPPF (Spatial Pyramid Pooling Fast)** : Ce module utilise plusieurs couches de max-pooling pour extraire efficacement des caractéristiques multi-échelles de l'image d'entrée.
- **C2PSA (Cross-Stage Partial with Spatial Attention)** : Ce bloc affine la capacité du modèle à se concentrer sélectivement sur les régions d'intérêt en appliquant une attention spatiale sur les caractéristiques extraites, améliorant ainsi la détection des détails fins des objets.

**b. Neck** : Le neck de YOLOv11 est responsable de l'agrégation et de la fusion des caractéristiques extraites par le backbone. Il comprend :

- **Modules C3k2 supplémentaires** : Utilisés pour une transformation efficace des caractéristiques.
- **Opérations de concaténation et d'upsampling** : Permettent de fusionner les informations à différentes échelles, améliorant la détection des objets de tailles variées.
- **Mécanisme C2PSA** : Intégré pour affiner davantage l'attention spatiale sur les caractéristiques fusionnées, renforçant la précision de la détection.

**c. Head** : La head de YOLOv11 effectue les prédictions finales, incluant les boîtes de délimitation, les masques et les classifications. Elle utilise :

- **Prédictions à trois échelles** : Similaire aux versions précédentes de YOLO, la head génère des prédictions à partir de trois cartes de caractéristiques (P3, P4 et P5), correspondant à différents niveaux de granularité dans l'image. Cette approche assure que les petits objets sont détectés avec plus de détails (P3), tandis que les objets plus grands sont capturés par des caractéristiques de niveau supérieur (P5).

### 2.5.3 Présentation des différentes tailles de modèles YOLOv11

**a. YOLOv11n (YOLOv11 nano)** : Est une version légère et optimisée de l'architecture YOLO de 11<sup>ème</sup> génération, conçue spécifiquement pour les appareils avec des ressources limitées, comme les smartphones, les drones ou les systèmes embarqués. Le suffixe "n" signifie "Nano", indiquant qu'il s'agit d'un modèle compact, avec moins de paramètres et une vitesse d'inférence très rapide, tout en maintenant une précision acceptable [71].

Malgré sa taille réduite, YOLOv11n conserve les principaux avantages des modèles YOLO : une détection en temps réel, un traitement en une seule passe, et la capacité à

localiser et classifier plusieurs objets simultanément. Il est particulièrement adapté aux cas d'usage où la rapidité et l'efficacité énergétique sont essentielles, comme dans les systèmes de détection embarqués pour véhicules ou caméras de surveillance.

**b. YOLOv11s (YOLOv11 Small) :** Est une version compacte et optimisée de la onzième génération de l'architecture YOLO, spécialement conçue pour allier rapidité et efficacité. Grâce à sa structure légère, elle permet une détection d'objets en temps réel tout en consommant peu de ressources, ce qui la rend idéale pour les systèmes embarqués, les appareils mobiles ou les applications industrielles nécessitant une exécution rapide. Bien qu'allégée par rapport aux versions plus grandes, [71] YOLOv11s conserve une bonne précision grâce à l'utilisation de modules d'extraction de caractéristiques performants. Cette version est particulièrement adaptée aux tâches comme la reconnaissance de panneaux de signalisation, où la rapidité d'analyse et la fiabilité sont essentielles.

**c. YOLOv11m (YOLOv11 Medium) :** Est une architecture de détection d'objets basée sur l'apprentissage profond, conçue pour reconnaître efficacement plusieurs objets dans une image en une seule passe. Elle repose sur le principe du traitement global de l'image, en la divisant en une grille et en prédisant simultanément les classes, les positions et les niveaux de confiance des objets détectés. Grâce à une structure bien équilibrée entre performance et efficacité, [71] YOLOv11m permet une détection rapide et précise, adaptée à des contextes variés tels que la reconnaissance des panneaux de signalisation, l'analyse vidéo en temps réel ou la perception visuelle dans les systèmes intelligents.

**d. YOLOv11l (YOLOv11 Large) :** Est une version puissante de l'architecture YOLO de 11ème génération, conçue pour offrir une haute précision dans les tâches de détection d'objets. Grâce à une structure plus profonde et un plus grand nombre de paramètres, ce modèle est capable de capturer des détails fins et de traiter des scènes complexes avec efficacité. Il fonctionne selon le principe "You Only Look Once", en analysant l'image entière en une seule passe pour localiser et classifier les objets. YOLOv11l est particulièrement adapté aux applications exigeantes comme la surveillance avancée, l'analyse vidéo à grande échelle ou la reconnaissance précise de panneaux de signalisation dans des environnements variés [71].

**e. YOLOv11x (YOLOv11 Extra Large) :** Est la version la plus puissante de l'architecture YOLOv11, conçue pour atteindre une précision maximale dans les tâches de détection d'objets. Grâce à une architecture très profonde et un grand nombre de paramètres,

elle est capable de capturer des détails complexes et de traiter des images haute résolution avec une grande fiabilité [71].

Comme les autres modèles de la famille YOLO (You Only Look Once), YOLOv11x fonctionne en une seule passe sur l'image, ce qui permet une détection rapide tout en assurant une excellente qualité des prédictions. Ce modèle est particulièrement adapté aux applications exigeantes comme la détection fine dans des environnements complexes, l'analyse vidéo avancée, ou encore les systèmes d'assistance à la conduite nécessitant une reconnaissance précise des objets.

**Tableau 2.1 : comparatif entre les différentes variantes de l'architecture YOLOv11**

Version	Nom complet	Taille du modèle	Vitesse (FPS)	Précision	Utilisation recommandée
YOLOv11n	Nano	Très légère	Très rapide	Faible à moyenne	Appareils embarqués, IoT, mobiles, temps réel simple
YOLOv11s	Small	Légère	Très rapide	Moyenne	Systèmes embarqués, détection basique rapide
YOLOv11m	Medium	Moyenne	Rapide	Bonne	Usage général, équilibre entre vitesse et précision
YOLOv11l	Large	Grande	Moins rapide	Très bonne	Analyse vidéo détaillée, scènes complexes
YOLOv11x	Extra Large	Très grande	Moins rapide	Excellente	Applications critiques, haute précision nécessaire

#### 2.5.4 Les anchors (ancres)

Les anchors en YOLO (You Only Look Once) sont des boîtes englobantes prédéfinies de différentes tailles et ratios hauteur/largeur. Elles ne servent de points de référence pour aider le modèle à prédire l'emplacement et le type d'objet dans une image. Le modèle ajuste ensuite ces ancres pour mieux correspondre à l'objet détecté. [73] Fonctionnement des ancres (anchors) dans l'architecture YOLO :

- **Boîtes englobantes prédéfinies** : Les ancres sont une série de boîtes englobantes prédéfinies, chacune avec une taille et un ratio hauteur/largeur spécifiques.

- **Points de référence** : Elles sont utilisées comme des points de référence pour le modèle, lui fournissant une base pour prédire les emplacements des objets.
- **Prédiction par ajustement** : Au lieu de prédire directement l'emplacement des boîtes englobantes, le modèle prédit des modifications ou des ajustements à ces ancres. En d'autres termes, il ajuste les dimensions et l'emplacement des boîtes ancres pour les rendre plus proches de la forme et de la taille de l'objet détecté.
- **Grille de prédiction** : Le modèle utilise une grille pour prédire les boîtes englobantes. Chaque cellule de la grille est associée à un ensemble d'ancres, ce qui permet au modèle de prédire différents emplacements et tailles d'objets.

### 2.5.5 Evaluation du model YOLOv11

L'évaluation du modèle YOLOv11 repose sur un ensemble de métriques standardisées permettant de mesurer à la fois la précision et l'efficacité de la détection d'objets. Les principaux indicateurs utilisés sont : le mAP (*mean Average Precision*), la précision, le rappel (recall), le score F1. Ces métriques sont calculées à partir des résultats de la détection, notamment en analysant les vrais positifs, faux positifs et faux négatifs, qui sont synthétisés dans une matrice de confusion

La matrice de confusion (voir la figure 2.15): est un outil fondamental en apprentissage supervisé, utilisé pour évaluer les performances d'un modèle de classification. Elle se présente sous la forme d'un tableau croisé qui compare les prédictions du modèle aux valeurs réelles des données (voir la figure 2.15).

Ce tableau permet non seulement de calculer des métriques clés (telles que la précision, le rappel et la F1-score), mais aussi d'identifier les types d'erreurs que le modèle commet : prédictions correctes, confusions entre classes, ou oublis de certaines catégories. Grâce à cette visualisation détaillée, les praticiens peuvent détecter les faiblesses du modèle, comprendre ses limites, et guider les étapes d'amélioration ou de réentraînement [74].

	Predicted Positive	Predicted Negative	
Actual Positive	<b>TP</b> <i>True Positive</i>	<b>FN</b> <i>False Negative</i>	<b>Sensitivity</b> $\frac{TP}{(TP + FN)}$
Actual Negative	<b>FP</b> <i>False Positive</i>	<b>TN</b> <i>True Negative</i>	<b>Specificity</b> $\frac{TN}{(TN + FP)}$
	<b>Precision</b> $\frac{TP}{(TP + FP)}$	<b>Negative Predictive Value</b> $\frac{TN}{(TN + FN)}$	<b>Accuracy</b> $\frac{TP + TN}{(TP + TN + FP + FN)}$

Figure 2.15 :Matrice de confusion [75].

Tableau 2.2 : Définition des termes TP, TN, FP, FN

Abréviation	Signification	Description
<b>TP</b>	Vrai Positif (True Positive)	Le modèle a prédit positif, et c'était vraiment positif
<b>TN</b>	Vrai Négatif (True Negative)	Le modèle a prédit négatif, et c'était vraiment négatif
<b>FP</b>	Faux Positif (False Positive)	Le modèle a prédit positif, mais c'était en réalité faux (erreur)
<b>FN</b>	Faux Négatif (False Negative)	Le modèle a prédit négatif, alors que c'était vraiment positif

Tableau 2.3 : Exemple concret de TP, TN, FP, FN (application sur les panneaux "Stop")

Situation réelle	Prédiction du modèle	Résultat	Abréviation
Il y a un panneau Stop	Le modèle détecte "Stop"	✓ Correct	<b>TP</b>
Il n'y a pas de panneau	Le modèle dit "Rien"	✓ Correct	<b>TN</b>
Il n'y a pas de panneau	Le modèle dit "Stop"	✗ Faux	<b>FP</b>
Il y a un panneau Stop	Le modèle dit "Rien"	✗ Faux	<b>FN</b>

## 2.6 Conclusion

La détection d'objets, et en particulier celle des panneaux de signalisation, joue un rôle central dans le développement des systèmes de transport intelligents et des véhicules autonomes. Ce chapitre a permis de mettre en lumière l'importance du traitement d'image et des algorithmes d'apprentissage profond dans l'automatisation de la reconnaissance visuelle. Ces technologies permettent non seulement d'améliorer la sécurité routière, mais aussi d'optimiser l'interprétation rapide et fiable de l'environnement routier.

Parmi les méthodes présentées, l'architecture YOLOv11 s'impose comme une solution performante, alliant précision, vitesse d'exécution et traitement en temps réel. Sa capacité à détecter simultanément plusieurs objets dans des conditions variées en fait un outil idéal pour les applications embarquées.

Les fondations théoriques et technologiques exposées dans ce chapitre constituent un socle essentiel pour la conception et l'implémentation de systèmes de détection robustes. Les prochains chapitres s'attacheront à décrire en détail les étapes pratiques de cette mise en œuvre, depuis la collecte des données jusqu'à l'intégration finale dans un système fonctionnel.

---

***Chapitre 03 :***  
***Implémentation, Résultat et***  
***Discussion***

---

### 3.1 Introduction

Dans ce chapitre, nous présentons la mise en œuvre de notre système de détection de panneaux de signalisation basé sur des algorithmes d'apprentissage profond. Notre travail s'appuie principalement sur l'algorithme YOLO (You Only Look Once), plus précisément sa version YOLOv11, afin de concevoir et d'entraîner un modèle capable d'identifier avec précision différents types de panneaux dans divers environnements.

Pour constituer notre jeu de données annoté, nous avons utilisé l'outil Roboflow, qui nous a permis de faciliter le processus de labellisation. L'entraînement du modèle a été réalisé sur Google Colab, tirant parti de ses ressources GPU, tandis que l'environnement de développement VS Code a été utilisé pour le développement, les tests locaux et l'intégration du système.

L'objectif principal de cette phase est de concevoir un système fiable, rapide et efficace pour la détection automatique des panneaux de signalisation, y compris dans des conditions réelles variées.

### 3.2 Plats-forme

#### 3.2.1 Google Colab

Google Colaboratory, communément appelé Google Colab, est un environnement de développement interactif gratuit proposé par Google. Il permet d'écrire et d'exécuter du code Python directement dans un navigateur web, tout en offrant un accès à des ressources matérielles puissantes comme des GPU (Graphics Processing Units) et des TPU (Tensor Processing Units)[34].

Google Colab est particulièrement utilisé dans les domaines de l'apprentissage automatique, du traitement d'images et de l'analyse de données, car il facilite l'entraînement de modèles complexes sans nécessiter de machine locale performante. Il est basé sur Jupyter Notebook, ce qui permet une exécution par cellules, la visualisation instantanée des résultats, ainsi que l'intégration de textes explicatifs en Markdown[34].

Dans le cadre de notre projet, Google Colab a été utilisé pour entraîner le modèle YOLOv11 à partir des données annotées. Son environnement cloud a permis de bénéficier d'une puissance de calcul suffisante pour effectuer l'apprentissage du modèle dans des délais raisonnables, sans contraintes matérielles.

### 3.2.2 Roboflow

Roboflow est une plateforme en ligne spécialisée dans la gestion de données pour les projets de vision par ordinateur. Elle offre une suite d'outils permettant d'importer, annoter, organiser, prétraiter et exporter des ensembles de données pour l'entraînement de modèles de détection d'objets, de classification et de segmentation.

Dans le contexte de notre projet, Roboflow a été utilisé principalement pour annoter manuellement les images, en identifiant et en étiquetant les panneaux de signalisation à détecter. Une fois les images annotées, la plateforme permet d'appliquer des transformations automatiques (comme le redimensionnement, la rotation, l'augmentation de données, etc.) afin d'enrichir le jeu d'entraînement et d'améliorer la robustesse du modèle.

Roboflow prend également en charge l'exportation des données annotées dans différents formats compatibles avec des frameworks populaires comme YOLO, TensorFlow, ou PyTorch, ce qui en fait un outil très pratique et flexible pour le développement de systèmes basés sur l'intelligence artificielle [35].

### 3.3 Logiciels utilisés

**-Visual Studio Code (VS Code) :** Visual Studio Code, souvent abrégé VS Code, est un éditeur de code source léger, open source et multiplateforme développé par Microsoft. Il est largement utilisé dans le développement logiciel grâce à sa rapidité, sa flexibilité et ses nombreuses extensions qui facilitent l'écriture, le test et le débogage du code.

Dans le cadre de notre projet de détection de panneaux de signalisation, VS Code a été utilisé comme environnement de développement principal. Il a permis d'écrire et d'organiser les scripts Python liés à l'interface graphique (Tkinter), au traitement d'images (OpenCV), et à l'intégration du modèle YOLO. Grâce à ses fonctionnalités telles que la coloration syntaxique, l'autocomplétion, le terminal intégré et le contrôle de version (Git), VS Code a offert un cadre de travail efficace et structuré tout au long du développement.

Sa compatibilité avec des environnements virtuels Python et sa prise en charge des notebooks Jupyter en font également un outil adapté aux projets combinant développement local et cloud.

### 3.4 Dataset

Dans notre projet, nous avons utilisé un dataset personnalisé (custom data) constitué de 2200 images de panneaux de signalisation, collectées et annotées spécifiquement pour les besoins de notre étude. L'annotation a été réalisée à l'aide de Roboflow, une plateforme efficace pour la gestion et la labellisation des images. Le dataset a été automatiquement divisé selon les standards de l'apprentissage profond :

- 1543 images pour l'entraînement (train).
- 440 images pour la validation (validation).
- 217 images pour les tests (test).

Afin de tester la robustesse et la généralisation de notre modèle, nous avons également utilisé un deuxième jeu de test indépendant composé de 433 images externes, accompagnées d'un dossier d'annotations (labels/). Ces images supplémentaires comprennent des cas difficiles, tels que des images bruitées (avec du bruit poivre et sel, gaussien, et speckle) ainsi que des images floues. Cette double approche nous a permis d'évaluer de manière plus complète la capacité de notre modèle à détecter les panneaux dans des conditions variées et réalistes.

### 3.5 Détails de l'implémentation

Pour constituer le jeu de données de notre projet, nous avons collecté un total de 2200 images à l'aide de smartphones, en veillant à couvrir un large éventail de situations réalistes. Les panneaux de signalisation ont été capturés sous différents angles de vue et dans diverses conditions d'éclairage : de jour comme de nuit, en plein soleil, à l'ombre, ou dans des environnements à faible luminosité. Nous avons intégré une grande diversité de panneaux en termes de types, tailles, orientations, mais aussi de complexité visuelle. En effet, notre dataset comprend également des cas particuliers tels que des panneaux flous, endommagés, partiellement couverts par des autocollants, ou encore rouillés, afin de confronter notre modèle à des scénarios proches de la réalité du terrain. Cette diversité visuelle s'est révélée essentielle pour entraîner un modèle YOLOv11 capable de généraliser efficacement face à des situations variées, imprévues et potentiellement bruitées. Ainsi, ce jeu de données constitue une base solide et représentative pour l'entraînement et l'évaluation de notre système de détection automatique de panneaux de signalisation.



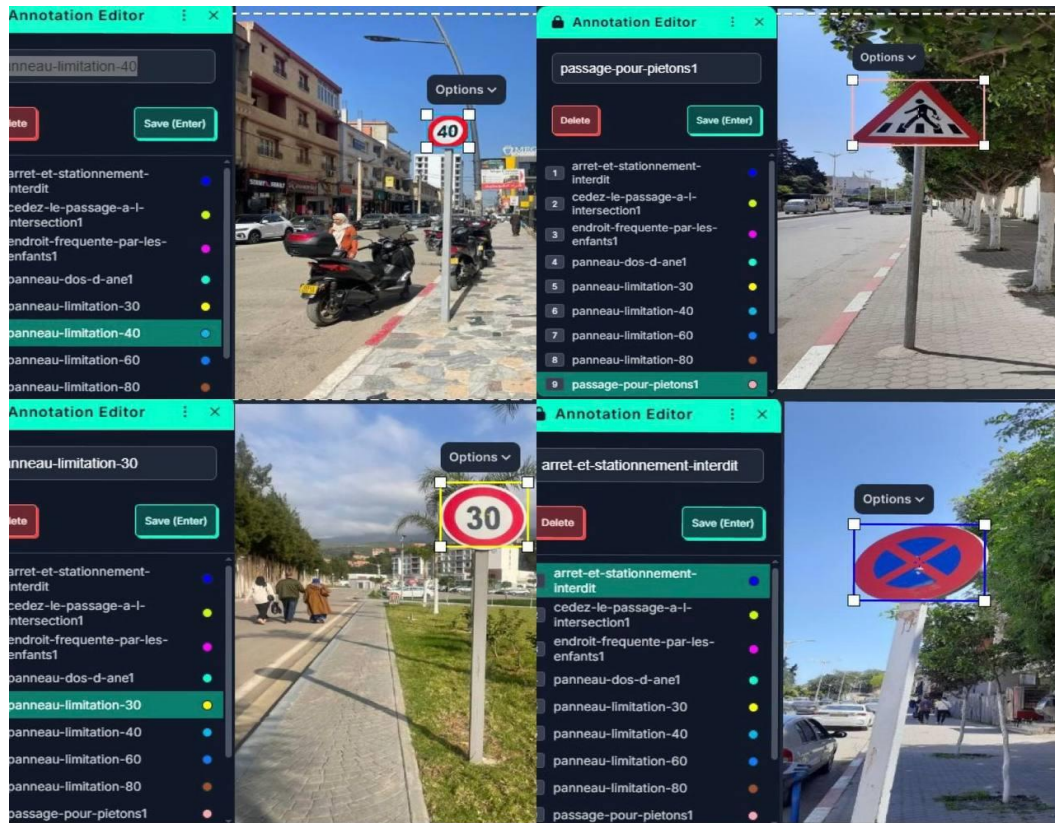
Figure 3.1 : Exemples d'images issues du dataset personnalisé.

### 3.5.1 Annotation du Datasets via Roboflow

L'annotation du dataset a été réalisée à l'aide de la plateforme Roboflow, un outil performant et intuitif pour la gestion et la labellisation d'images. Avant l'utilisation de Roboflow pour l'annotation, toutes les images ont d'abord été uniformément redimensionnées à une taille de 600×600 pixels, afin d'assurer une cohérence dans les dimensions d'entrée et d'optimiser le traitement par les modèles d'apprentissage profond.

Chaque image a ensuite été annotée manuellement en traçant des boîtes englobantes autour des panneaux de signalisation, avec attribution de la classe correspondante. Nous avons défini un total de 12 classes : Stop, Sens interdit, Endroit fréquenté par les enfants, Cédez le passage à l'intersection, Panneau dos d'âne, Panneau limitation de vitesse 30, Panneau limitation de vitesse 40, Panneau limitation de vitesse 60, Panneau limitation de vitesse 80, Rond-point, Arrêt et stationnement interdit, et Passage pour piétons.

Roboflow a permis non seulement de faciliter l'annotation, mais aussi de convertir automatiquement les données au format YOLO et de générer les sous-ensembles d'entraînement, de validation et de test. Cette annotation précise et structurée a constitué une étape cruciale pour garantir l'efficacité et la fiabilité de l'apprentissage du modèle.



**Figure 3.2 :** Procédure d'annotation des panneaux de signalisations à l'aide de Roboflow.

### 3.5.2 Entraînement du modèle YOLOv11

L'entraînement du modèle YOLOv11 a été réalisé à l'aide de Google Colab, une plateforme cloud offrant une puissance de calcul GPU gratuite, parfaitement adaptée aux besoins du deep learning. Après avoir importé le dataset annoté depuis Roboflow au format YOLO, nous avons procédé à la configuration de l'environnement d'entraînement en installant les bibliothèques nécessaires et en important l'architecture YOLOv11. Les hyperparamètres ont été ajustés en fonction des spécificités du dataset, notamment le nombre d'époques, le taux d'apprentissage, la taille des batchs et les dimensions des images.

Afin de comparer les performances et d'identifier le modèle le plus adapté à notre cas d'usage, nous avons entraîné successivement les variantes YOLOv11s, YOLOv11n, YOLOv11m, YOLOv11l et YOLOv11x. Chaque version offre un compromis différent entre vitesse d'inférence et précision, ce qui nous a permis d'observer l'impact du nombre de paramètres sur la qualité des détections. Grâce aux ressources GPU disponibles sur Colab, l'entraînement s'est déroulé efficacement, et le modèle a appris à détecter avec précision les 12 classes de panneaux de signalisation.

Tout au long du processus, nous avons surveillé plusieurs métriques de performance comme la mAP (mean Average Precision), la perte d'entraînement (loss) et le taux de détection, afin d'évaluer la qualité de l'apprentissage, détecter les éventuels cas de surapprentissage, et sélectionner le modèle le plus performant pour les tests finaux.

### 3.6 Interface graphique

L'évaluation du modèle YOLOv11 a été menée à l'aide d'une interface graphique (GUI) développée en Python avec la bibliothèque Tkinter, exécutée sous Visual Studio Code. Cette application permet de tester les performances du modèle sur différents supports : images, vidéos et flux webcam. Elle a été utilisée pour évaluer un total de 433 images, incluant des cas présentant des dégradations visuelles variées telles que du bruit gaussien, du bruit poivre et sel, du speckle ou encore des effets de flou, afin de vérifier la robustesse du modèle dans des conditions complexes et réalistes.

Un module OCR (reconnaissance optique de caractères) a été intégré pour lire et identifier les numéros ainsi que les textes présents sur les panneaux de signalisation, améliorant ainsi la précision globale de l'analyse.

L'interface propose également des fonctionnalités avancées, notamment :

- Le calcul du DPI (résolution en points par pouce), appliqué aux images, vidéos et flux webcam,
- L'affichage en temps réel du FPS (frames per second), pour mesurer les performances en mode dynamique.

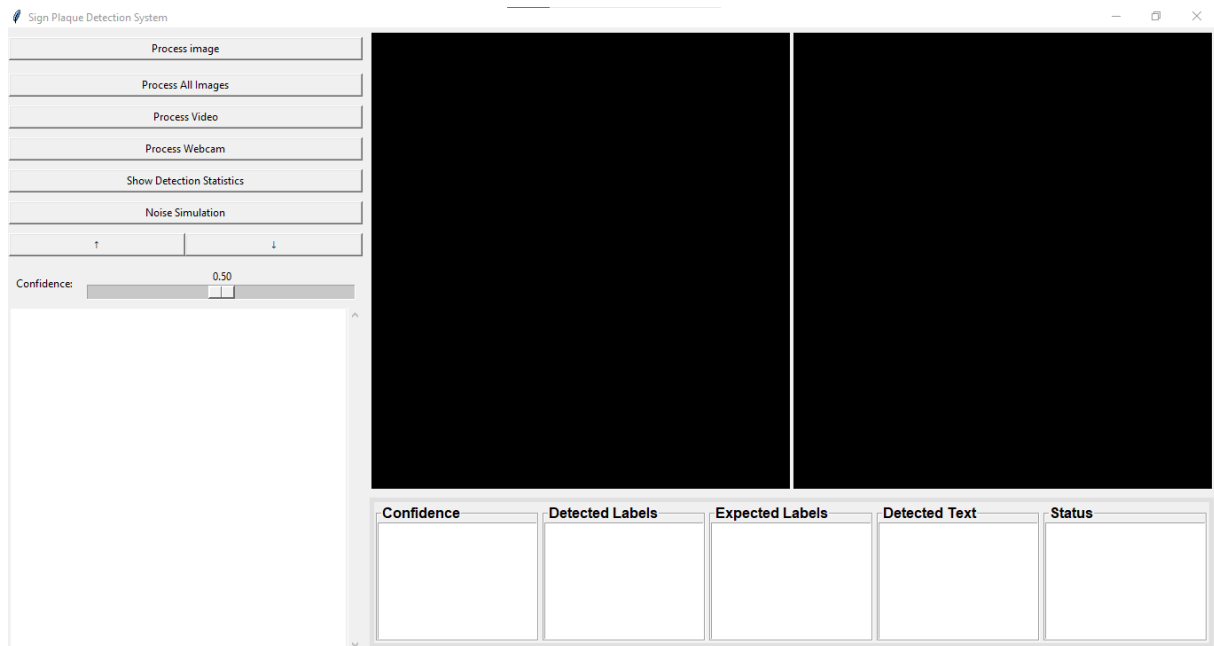
Cette interface graphique offre ainsi une évaluation à la fois qualitative et quantitative du système, aussi bien en conditions contrôlées qu'en conditions proches du réel, tout en facilitant l'analyse et la visualisation des résultats.

#### **-Description de l'interface graphique :**

L'interface graphique développée pour l'évaluation du modèle YOLOv11 intègre une barre d'information dynamique, affichée en bas de l'écran, qui permet de suivre en temps réel les résultats détaillés de chaque détection réalisée sur des images, des vidéos ou des flux en direct via webcam. Pour chaque image ou frame, cette barre affiche :

- Le nom du panneau détecté, selon la classification réalisée par YOLOv11,
- Le score de confiance (confidence) du modèle, exprimé en pourcentage,

- Le statut de la prédiction (Correct ou Incorrect), obtenu en comparant le résultat du modèle avec le label réel attendu,
- Et le résultat de l'identification du panneau à l'aide d'un module OCR, spécialement utilisé pour extraire les chiffres des panneaux de limitation de vitesse (ex. : "30", "40", "80", etc.), ce qui renforce la précision de l'analyse.



**Figure 3. 3** :l'image contenant l'interface graphique Tkinter.

Lors de l'analyse de vidéos ou de flux webcam, la barre affiche également les FPS (frames per second) en temps réel, afin de mesurer les performances du système en conditions dynamiques.

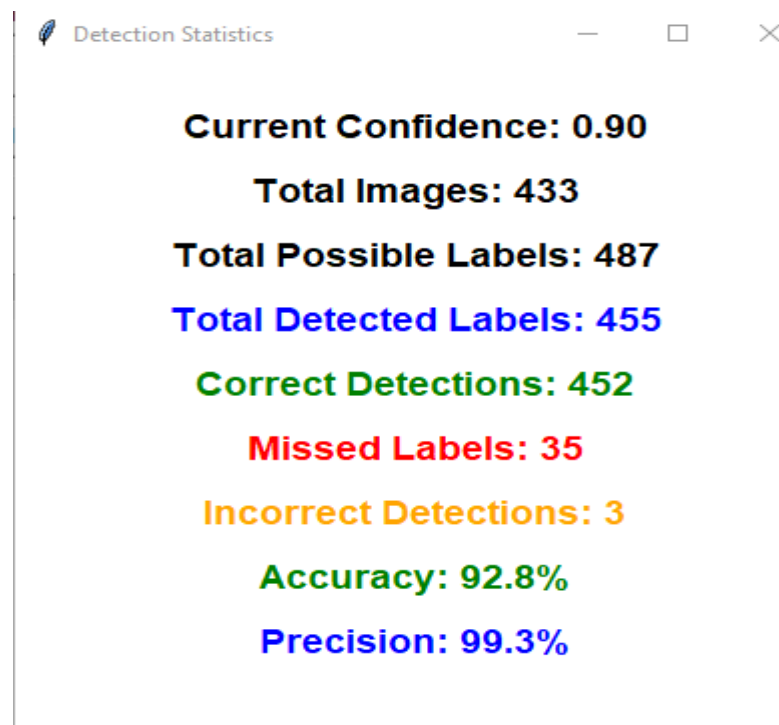
De plus, pour chaque panneau détecté, le système calcule sa taille en pixels grâce à l'exploitation du DPI (dots per inch), ce qui permet d'estimer physiquement la taille relative des objets détectés selon la résolution de l'image ou du flux.

L'application comprend également quatre graphiques de suivi de performance :

1. Un graphe de détection, représentant le nombre total d'objets détectés,
2. Un graphe d'identification, affichant les résultats d'identification textuelle,
3. Un graphe combiné détection + identification, illustrant les cas de succès complets,
4. Un graphe de ratio, qui montre le rapport entre les objets identifiés correctement et ceux simplement détectés.

Enfin, une section intitulée "Show Detection Statistics" fournit une synthèse globale sous forme de tableau statistique avec les indicateurs suivants :

- Nombre total d'objets détectés (Total Detected Labels),
- Nombre total d'objets attendus (Total Possible Labels),
- Nombre de détections correctes (Correct Detections),
- Nombre d'objets manqués (Missed Labels),
- Nombre d'erreurs de détection (Incorrect Detections),
- Ainsi que les deux métriques clés :
  - Précision(*Precision*) : pourcentage de détections correctes parmi toutes les détections effectuées par le modèle.
  - Exactitude(*Accuracy*) : pourcentage de bonnes prédictions parmi toutes les vérités terrain possibles.



**Figure 3.4 :** Affichage des statistiques de détection dans l'interface.

### 3.6.1 Evaluation des performances du modèle YOLO

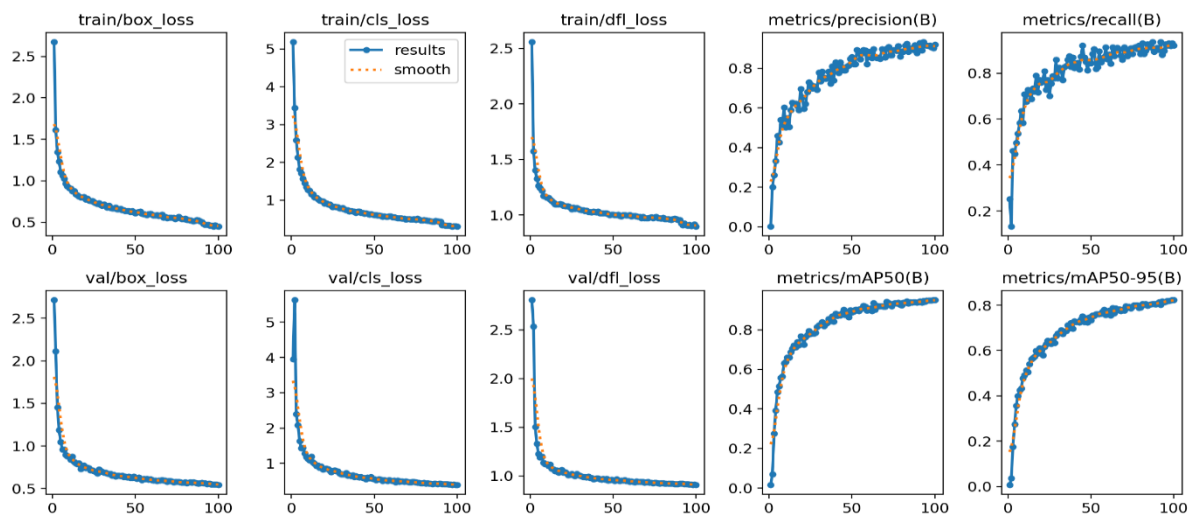
L'évaluation des performances des modèles YOLOv11 dans ses différentes variantes (n, s, m, l, x) a été menée de manière rigoureuse afin d'identifier le compromis optimal entre précision, vitesse et consommation de ressources. Pour chaque version du modèle, des métriques standard ont été calculées, notamment la mAP (mean Average Precision), la

précision, le rappel, ainsi que le temps d'inférence par image. Les tests ont été effectués sur l'ensemble de validation, ainsi que sur un second jeu de test composé de 433 images, incluant des cas complexes (images floues, bruitées et panneaux dégradés). Les modèles plus légers comme YOLOv11n et YOLOv11s ont montré des temps de traitement plus rapides, ce qui les rend adaptés aux environnements temps réel, tandis que les modèles plus lourds comme YOLOv11x ont obtenu de meilleures performances en termes de précision, au prix d'une latence plus élevée. Cette évaluation comparative nous a permis de sélectionner le modèle le plus adapté à nos contraintes d'application tout en maintenant une robustesse satisfaisante en détection.

### 1- Résultat obtenu lors de l'entraînement du modèle YOLOv11:

Les pertes d'entraînement et de validation (box\_loss, cls\_loss, dfl\_loss) sont faibles et convergentes, ce qui indique :

- **Un bon apprentissage sans surentraînement** : le modèle ne mémorise pas simplement le jeu d'entraînement, mais apprend à généraliser efficacement.
- **Une capacité de généralisation solide à de nouvelles images** : le modèle est capable de s'adapter à des données jamais vues auparavant, ce qui est crucial pour des applications



réelles.

**Figure 3.5** : Courbes d'Entraînement et de Validation du Modèle YOLOv11m.

Les résultats finaux obtenus à l'issue de l'entraînement sur 100 époques démontrent les performances exceptionnelles de notre modèle YOLO dans la détection des panneaux de signalisation :

À l'issue de l'entraînement sur 100 époques, le modèle YOLOv11 démontre les performances de notre système dans la détection des panneaux de signalisation :

- **Exactitude et Précision :** Le modèle atteint une précision (précision) élevée, comprise entre 0,95 et 0,97. Cela signifie que 95 % à 97 % des objets détectés sont effectivement corrects, reflétant un taux de faux positifs extrêmement faible. Cette fiabilité est cruciale pour l'identification fiable des panneaux de signalisation.
- **Rappel (Recall) :** Le rappel final est d'environ 0,91, indiquant que le modèle est capable de repérer 91 % des panneaux de signalisation présents dans les images. Ce résultat témoigne d'une très bonne sensibilité, avec très peu d'objets manqués (faux négatifs).
- **Précision Moyenne (mAP) :**
  - mAP@0.5 : 0,96 – excellente capacité du modèle à détecter les objets avec un seuil standard (IoU ≥ 0,5).
  - mAP@0.5:0.95 : 0,84 – le modèle reste performant même avec des critères plus stricts, prouvant sa robustesse et sa régularité sur différents seuils de similarité.

Ces valeurs sont des indicateurs globaux de performance sur toutes les classes d'objets et à différents seuils de similarité (IoU).

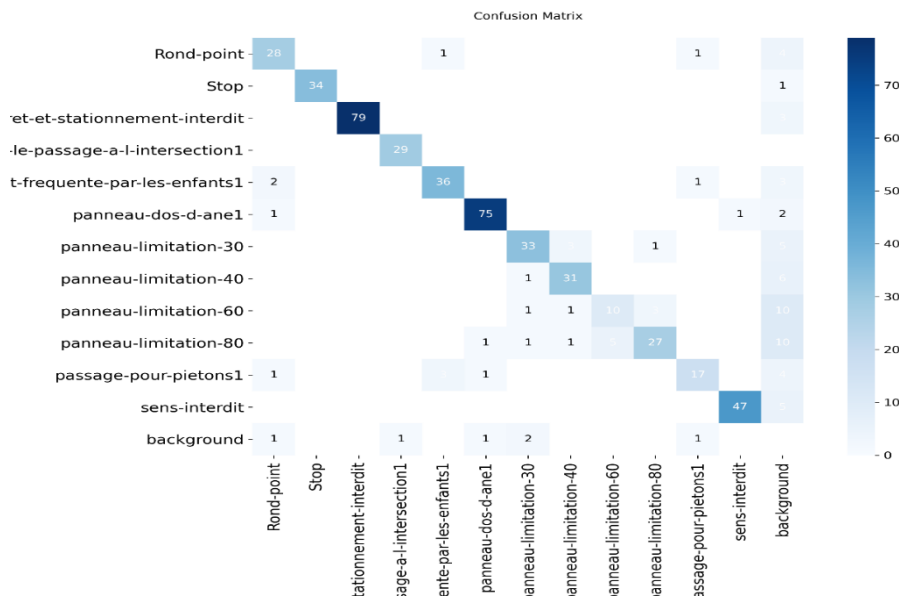


Figure 3.6 : Analyse des prédictions via la matrice de confusion.

-Explication de la matrice :

Cette figure représente la matrice de confusion, un outil d'évaluation des performances d'un modèle de classification, en particulier pour les tâches de reconnaissance d'objets.

- Axe horizontal (colonnes) : les classes prédites par le modèle.
- Axe vertical (lignes) : les classes réelles (ou "vérités terrain").

Chaque cellule indique le nombre d'occurrences où une classe réelle a été prédite comme une autre.

-Analyse de la matrice :

- Les valeurs sur la diagonale (de haut gauche à bas droite) indiquent les prédictions correctes.

Par exemple :

- Le modèle a correctement identifié 79 fois la classe "arret-et-stationnement-interdit".
- Il a correctement identifié 75 fois la classe "panneau-dos-d-ane1".

- Les valeurs hors diagonale indiquent les erreurs de classification, c'est-à-dire que le modèle a confondu une classe avec une autre.

Par exemple :

- La classe "panneau-limitation-30" a été confondue 2 fois avec "panneau-dos-d-ane1".
- La classe "panneau-limitation-80" a été confondue 10 fois avec "panneau-limitation-60".

-Barre de couleur à droite :

Elle indique l'intensité des valeurs dans la matrice : plus la couleur est foncée, plus le nombre d'occurrences est élevé.

## 2- Évaluation des performances du modèle YOLOv11 avec l'interface graphique :

-**Accuracy**, dont la valeur mesurée varie entre 0.81 et 0.98, indiquant un très bon taux de justesse globale,

-**Precision**, estimée entre 0.96 et 0.99, confirmant la haute fiabilité des détections positives effectuées par le modèle.

Cette interface complète et interactive permet une évaluation fine et rigoureuse du modèle YOLOv11, tant sur sa capacité à détecter qu'à identifier correctement les objets, dans des conditions variées et réalistes.

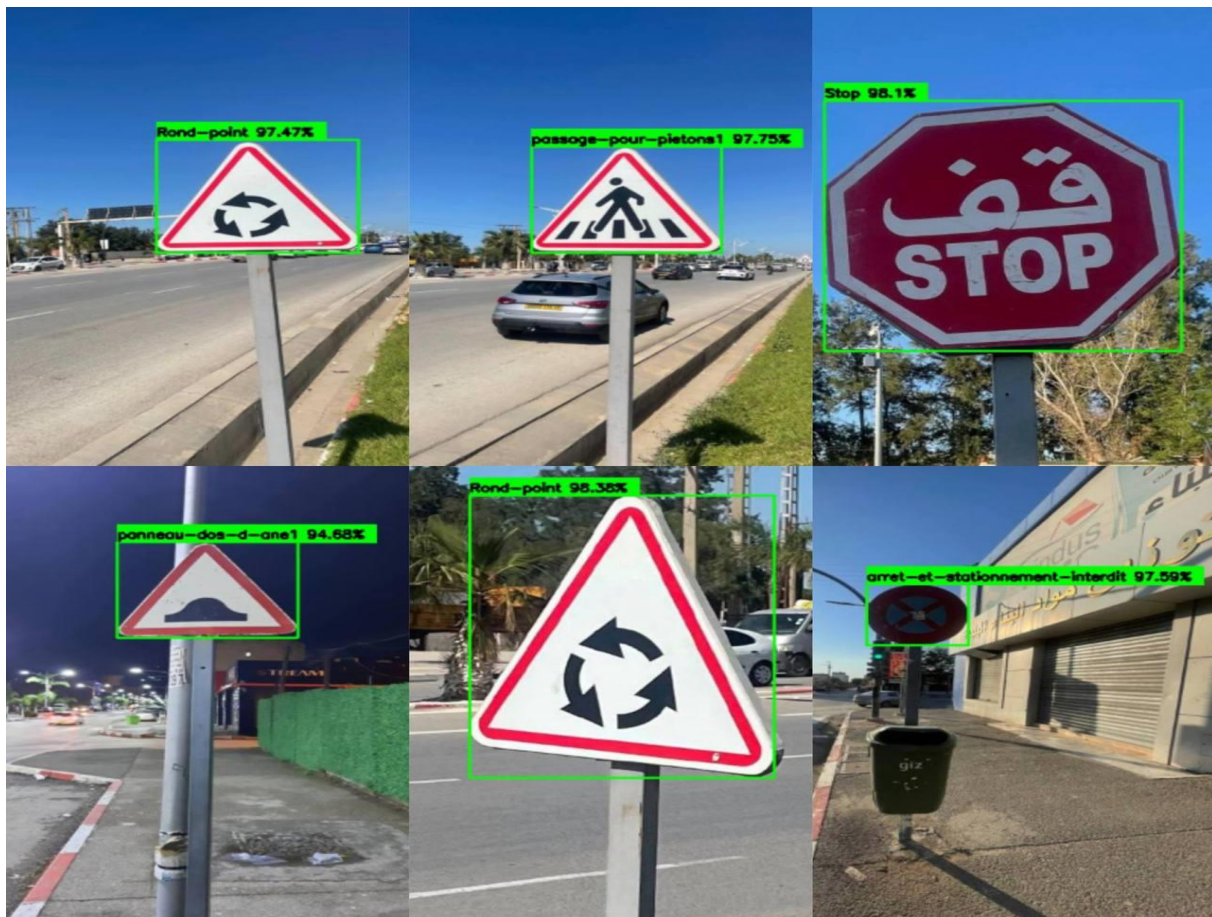


Figure 3.7 : Évaluation des Performances sur Images Nettes en Plein Jour.



Figure 3.8 : Résultats Expérimentaux en Conditions de Faible Luminosité.



Figure 3.9: Tests de Robustesse sur Images Floues : Scènes de Jour et de Nuit.



Figure 3.10: Étude de la Robustesse en Condition Dégradée : Bruit Gaussien, Poivre et Sel, Speckle.

**Tableau 3.1 :Analyse des performances des modèles YOLOv11 sur 433 images à différents seuils de confiance:**

Modèle	Confidence	Total Possible	Total Détecté	Correct	Manqué	Incorrect	Accuracy	Precision
<b>YOLOv11n</b>	<b>0.5</b>	487	484	472	15	12	96.9%	97.5%
	<b>0.6</b>	487	479	470	17	9	96.5%	98.1%
	<b>0.7</b>	487	476	468	19	8	96.1%	98.3%
	<b>0.8</b>	487	473	468	19	5	96.1%	98.9%
	<b>0.9</b>	487	455	452	35	3	92.8%	99.3%
<b>YOLOv11s</b>	<b>0.5</b>	487	493	477	10	16	97.9%	96.8%
	<b>0.6</b>	487	490	477	10	13	97.9%	97.3%
	<b>0.7</b>	487	484	476	11	8	97.7%	98.3%
	<b>0.8</b>	487	475	471	16	4	96.7%	99.2%
	<b>0.9</b>	487	408	407	80	1	83.6%	99.8%
<b>YOLOv11m</b>	<b>0.5</b>	487	491	478	9	13	98.2%	97.4%
	<b>0.6</b>	487	490	478	9	12	98.2%	97.6%
	<b>0.7</b>	487	485	476	11	9	97.7%	98.1%
	<b>0.8</b>	487	474	469	18	5	96.3%	98.9%
	<b>0.9</b>	487	400	398	89	2	81.7%	99.5%
<b>YOLOv11l</b>	<b>0.5</b>	487	490	476	11	14	97.7%	97.1%
	<b>0.6</b>	487	486	474	13	12	97.3%	97.5%
	<b>0.7</b>	487	482	473	14	9	97.1%	98.1%
	<b>0.8</b>	487	475	469	18	6	96.3%	98.7%
	<b>0.9</b>	487	401	398	89	3	81.7%	99.3%
<b>YOLOv11x</b>	<b>0.5</b>	487	490	480	7	10	98.6%	98.0%
	<b>0.6</b>	487	488	479	8	9	98.4%	98.2%
	<b>0.7</b>	487	485	479	8	6	98.4%	98.8%
	<b>0.8</b>	487	482	476	11	6	97.7%	98.8%
	<b>0.9</b>	487	416	414	73	2	85.0%	99.5%

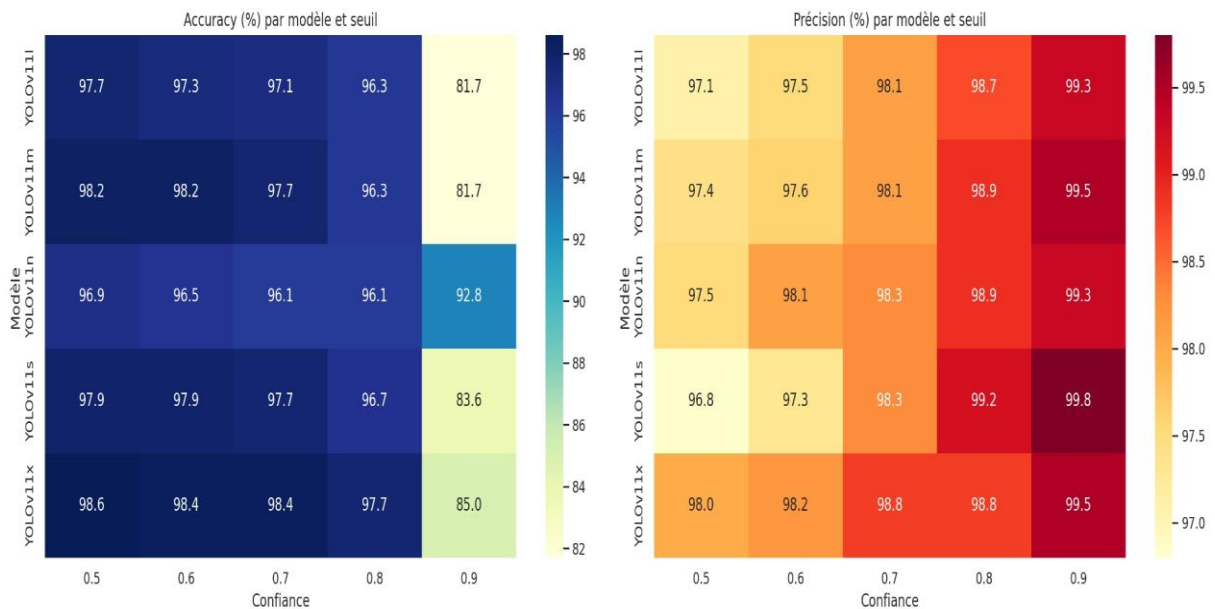
**-Interprétation des colonnes :**

- ❖ **Total Détecté** : nombre total de panneaux de signalisation détectés par le modèle, incluant à la fois les détections correctes et les erreurs (faux positifs + faux négatifs).
- ❖ **Total Possible** : nombre réel de panneaux de signalisation présents dans les images (valeur de référence : 487).
- ❖ **Correct** : nombre de panneaux de signalisation correctement détectés, c'est-à-dire correspondant aux annotations réelles (vrais positifs).
- ❖ **Manqué** : nombre de panneaux de signalisation présents mais non détectés par le modèle (faux négatifs).
- ❖ **Incorrect** : nombre de panneaux de signalisation détectés à tort, c'est-à-dire qui ne correspondent pas à des objets réels (faux positifs).
- ❖ **Accuracy** : proportion de panneaux de signalisation détectés par rapport au nombre total de panneaux présents dans les images.

$$Accuracy = \frac{Total\ Détecté}{Total\ Possible} \times 100$$

- ❖ **Precision** : proportion de panneaux de signalisation correctement identifiées parmi l'ensemble des détections effectuées.

$$Precision = \frac{Correct}{Total\ Détecté} \times 100$$



**Figure 3.11:** comparatif de l'Accuracy et de la Précision (%) des variantes YOLOv11 en fonction du seuil de confiance.

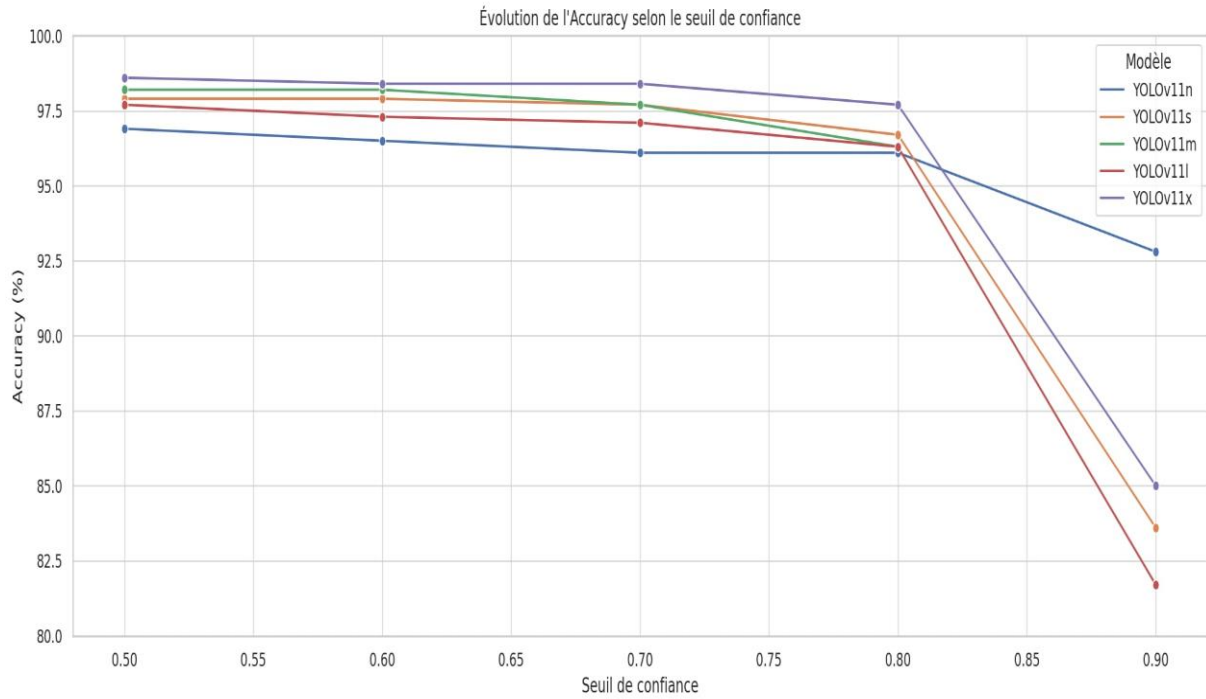


Figure 3.12: Graphe d'évolution de l'Accuracy (%) des modèles YOLOv11 en fonction du seuil de confiance.

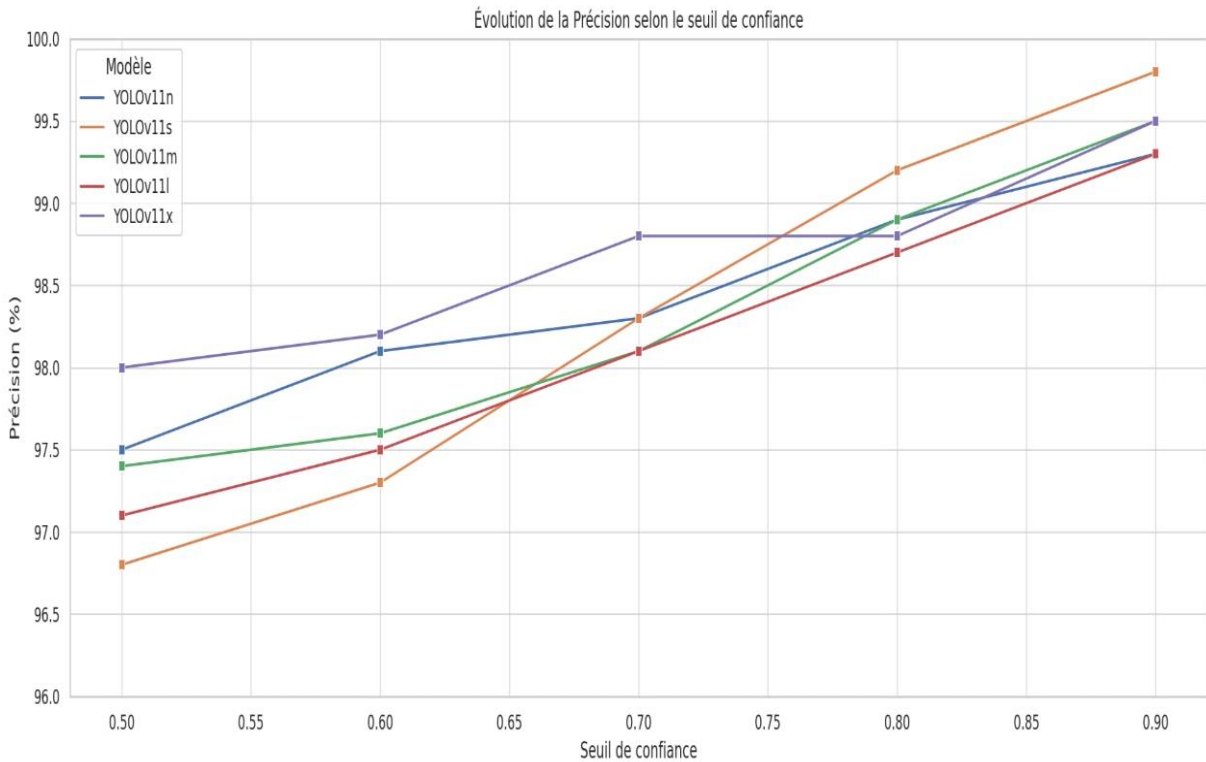


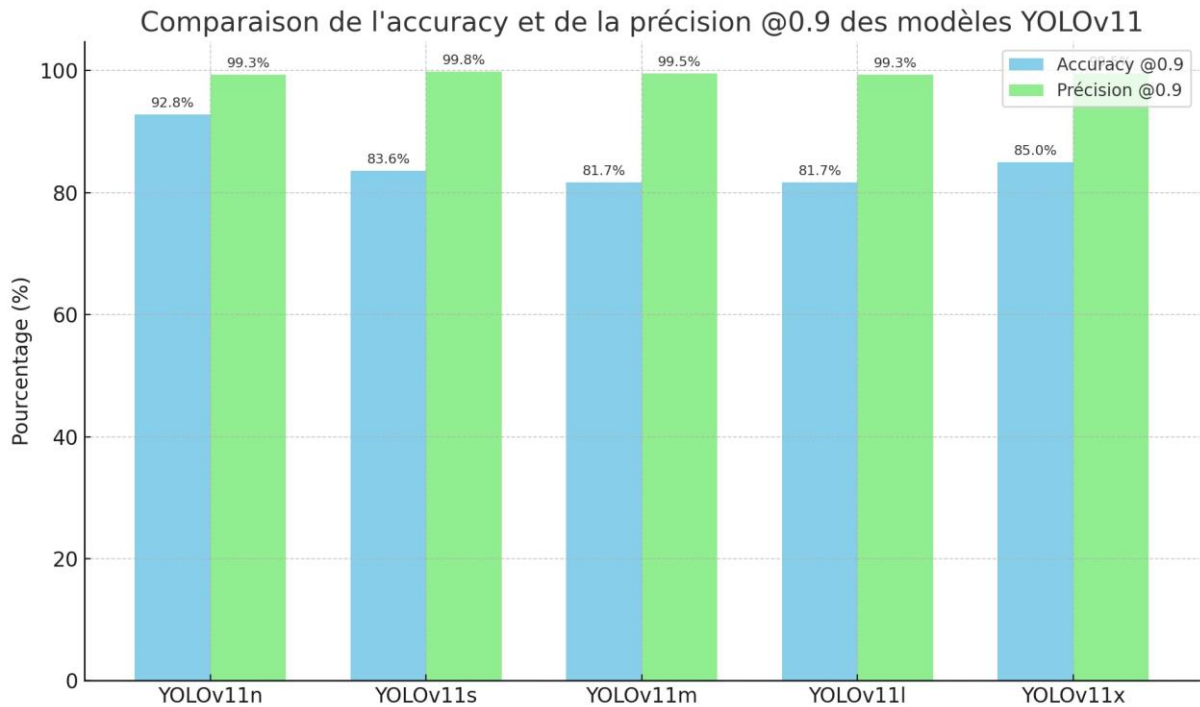
Figure 3.13 : Graphe d'évolution de précision des modèles YOLOv11 en fonction du seuil de confiance.

**-Interprétation des Résultat**

Afin d'évaluer les performances du modèle YOLOv11, plusieurs variantes ont été testées: YOLOv11n, s, m et l. Chaque version présente un compromis différent entre rapidité, précision et stabilité. Le tableau ci-dessous résume les principales observations à un seuil de confiance élevé (0.9) et à des seuils plus modérés (0.6–0.8).

**Tableau 3.2** : Discussion des résultats pour les différentes version du model YOLO11.

<b>Modèle</b>	<b>Précision @0.9</b>	<b>Accuracy @0.9</b>	<b>Robustesse (0.6–0.8)</b>	<b>Remarques principales</b>
<b>YOLOv11n</b>	99.3 %	92.8 %	Bonne jusqu'à 0.8	Léger et rapide, chute de détection à seuil élevé
<b>YOLOv11s</b>	99.8 %	83.6 %	Accuracy >96 %, Préc. >97 %	Bon équilibre, très robuste sur seuils modérés
<b>YOLOv11m</b>	99.5 %	81.7 %	Accuracy >97 % jusqu'à 0.7	Très stable, bon compromis performance/temps
<b>YOLOv11l</b>	99.3 %	81.7 %	Performances élevées globalement	Similaire à YOLOv11m, meilleure couverture à bas seuil
<b>YOLOV11x</b>	99.5 %	85.0%	Accuracy >97 % et Précision >98 %	Très performant, meilleure accuracy globale (98.6 % @0.5), stabilité jusqu'à 0.8, mais sensible au seuil élevé



**Figure 3.14** : Comparaison de l'accuracy et de la précision a seuil de 0.9 des modèles YOLOv11.

Après évaluation des performances des différentes variantes de YOLOv11 à un seuil de confiance élevé (@0.9), nous avons constaté que le modèle **YOLOv11n** offre le **meilleur compromis** entre **accuracy** (92.8 %) et **précision** (99.3 %).

Malgré sa légèreté, il détecte efficacement les panneaux tout en limitant les fausses alertes. Sa rapidité d'exécution et sa faible consommation de ressources en font un choix idéal pour une **intégration dans une application temps réel ou un système embarqué**.

C'est pourquoi nous avons choisi **YOLOv11n** comme **modèle final** pour notre interface graphique.

#### - Discussion des résultats entre les différentes versions du modèle YOLOv11 :

- **YOLOv11x** : Modèle le plus puissant mais très lourd, adapté aux systèmes à hautes performances.
- **YOLOv11l** : Précis mais lent, nécessite plus de ressources pour un gain limité.
- **YOLOv11m** : Bon équilibre, mais accuracy faible à seuil élevé.
- **YOLOv11s** : Léger et rapide, précision élevée mais moins fiable que "n".
- **YOLOv11n** : Très léger, rapide et précis, idéal pour les systèmes embarqués en temps réel.

### 3.6.1 Interprétions Globale avec l'interface graphique

Les images suivantes contiennent quatre graphiques représentant les performances des modèles de détection YOLOv11 (x, s, n, l, m) en fonction du seuil de confiance. Chaque série de graphiques permet d'évaluer la robustesse et la précision de ces variantes du modèle à différents niveaux de confiance. On y observe l'évolution du nombre total d'objets détectés, du nombre d'objets correctement identifiés, de la comparaison entre détection et identification, ainsi que du taux de précision (identification/détection). Ces visualisations permettent de comparer efficacement le comportement de chaque version du modèle selon la qualité du signal de confiance, afin d'identifier le meilleur compromis entre sensibilité et fiabilité.

#### 1. Modèle YOLOv11

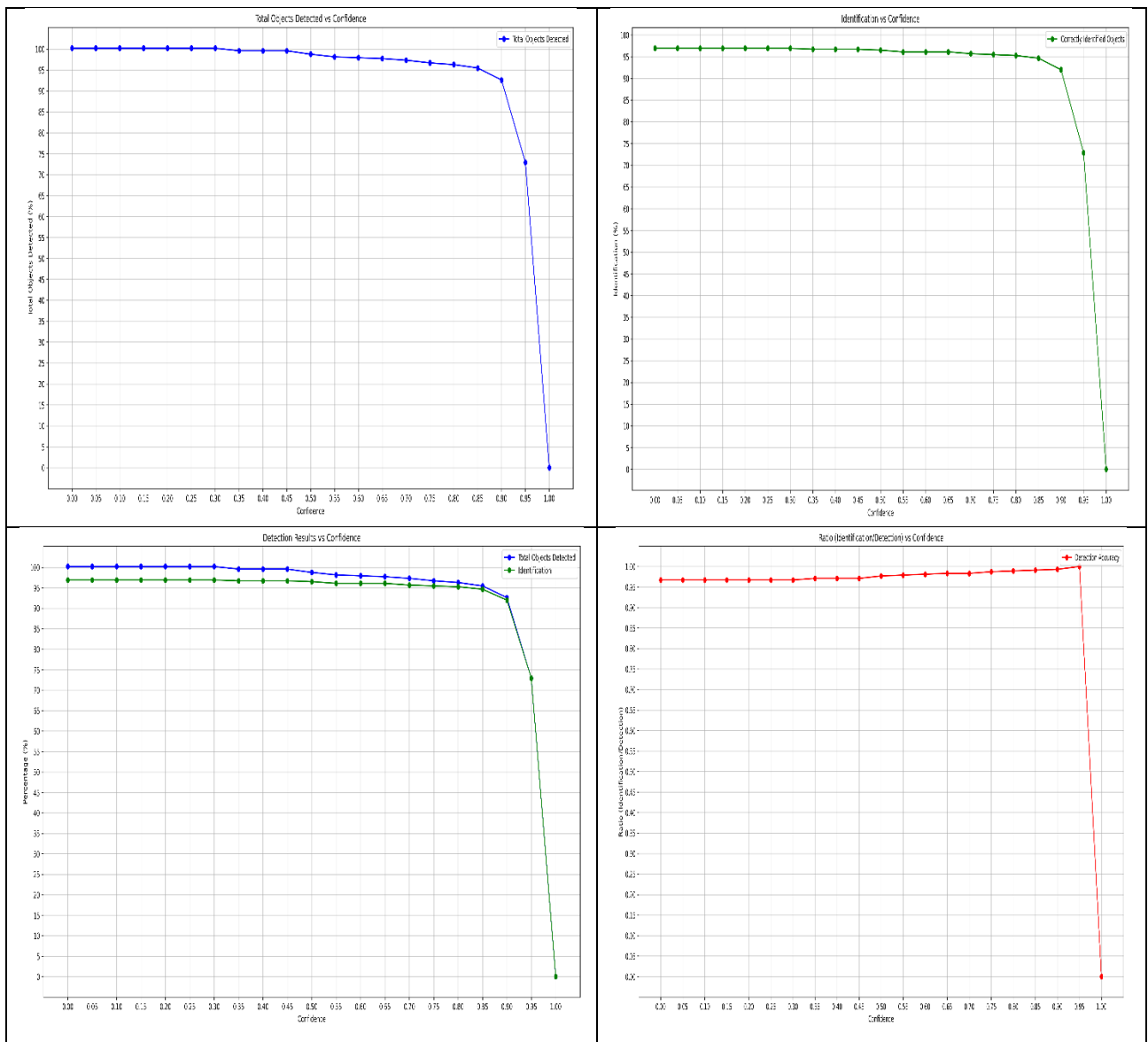


Figure 3.15: Graphe de Résultats pour le Modèle YOLOv11n.

Le graphique ci-dessus illustre les performances du modèle YOLOv11n, une version compacte et rapide du modèle YOLOv11, analysées en fonction de différents seuils de confiance. On observe que le nombre total d'objets détectés reste élevé et stable jusqu'à un seuil de confiance d'environ 80 %, au-delà duquel il chute brutalement. De même, le nombre d'objets correctement identifiés suit une courbe similaire, indiquant que YOLOv11n maintient une bonne capacité d'identification à faible et moyenne confiance. Le taux de précision (rapport identification/détection) reste très proche de 1 jusqu'à un seuil de 85 %, ce qui montre que presque tous les objets détectés sont correctement identifiés. Enfin, la comparaison conjointe des détections et identifications confirme que YOLOv11n est performant à des seuils de confiance modérés, mais montre des limites notables en haute confiance. Ce modèle est donc particulièrement adapté aux contextes nécessitant un traitement rapide avec des ressources limitées, tout en assurant une détection efficace à seuil modéré.

## 2. Modèle YOLOv11s

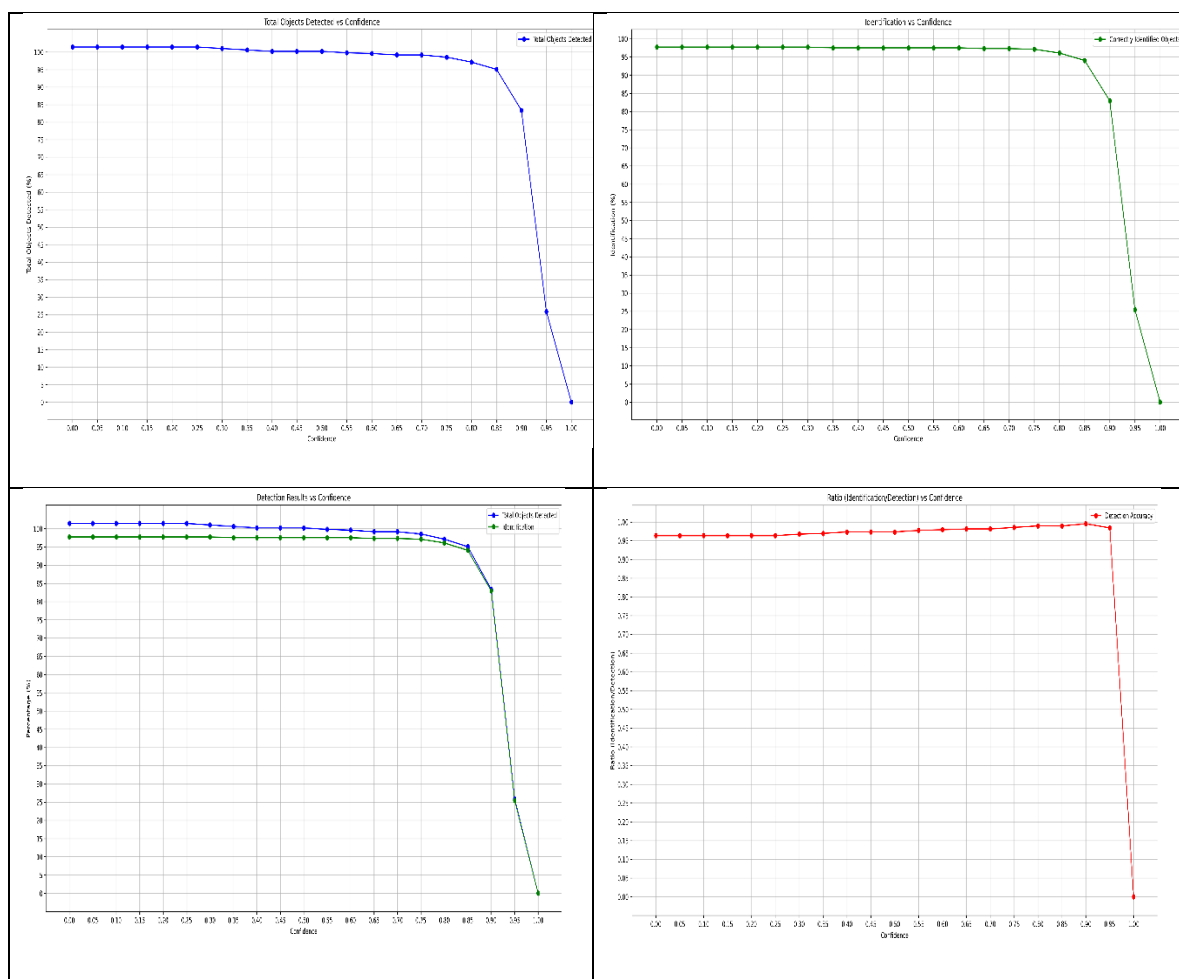


Figure 3.16: Graphiques de Résultats pour le Modèle YOLOv11s.

Le modèle YOLOv11s, version légère de YOLOv11 optimisée pour une exécution rapide avec une précision raisonnable, démontre une stabilité impressionnante dans la détection et l'identification des objets jusqu'à un seuil de confiance d'environ 80 %. Le nombre total d'objets détectés ainsi que celui des objets correctement identifiés restent très proches de 100 %, attestant d'une bonne robustesse du modèle à des seuils faibles et modérés. Toutefois, on observe une chute marquée des performances à partir d'un seuil de 85 %, révélant une sensibilité accrue du modèle aux seuils élevés. Le rapport identification/détection, ou précision de la détection, reste proche de 1 jusqu'au seuil critique, indiquant une grande fiabilité dans la correspondance entre les objets détectés et ceux effectivement bien reconnus. YOLOv11s constitue ainsi un bon compromis entre rapidité et efficacité pour des applications en temps réel, surtout lorsque la tolérance à des seuils de confiance modérés est acceptable.

### 3. Modèle YOLOv11m

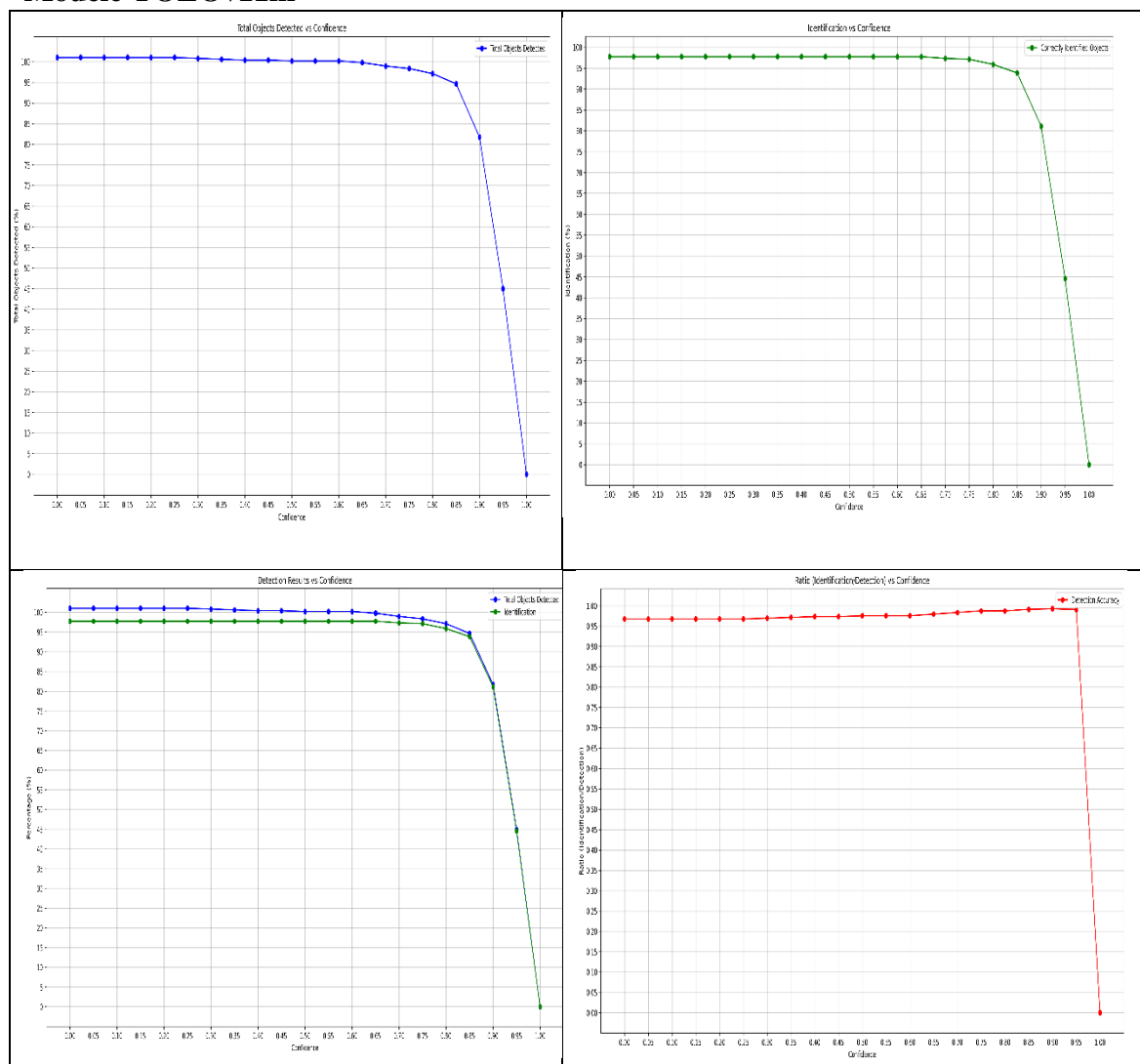


Figure 3.17 : Graphiques de Résultats pour le Modèle YOLOv11m.

Le modèle YOLOv11m, comme illustré par les graphiques, est un modèle de détection d'objets évalué selon différentes métriques en fonction du seuil de confiance. On observe que jusqu'à un seuil de confiance d'environ 0.8, le modèle détecte et identifie près de 100 % des objets avec une très haute précision. Cependant, au-delà de ce seuil, les performances chutent drastiquement, indiquant que des seuils trop élevés entraînent une perte de détection (voir les courbes bleue et verte dans les graphiques du haut et du bas). Le graphique en bas à droite montre que le taux de précision (identification/détection) reste proche de 1 jusqu'à ce que le seuil dépasse 0.9, où il diminue fortement. Cela suggère que YOLOv11m offre un excellent compromis entre quantité de détections et précision jusqu'à un certain niveau de confiance, au-delà duquel la rigueur du seuil filtre trop d'objets, réduisant ainsi la performance globale du système.

#### 4. Modèle YOLOv11

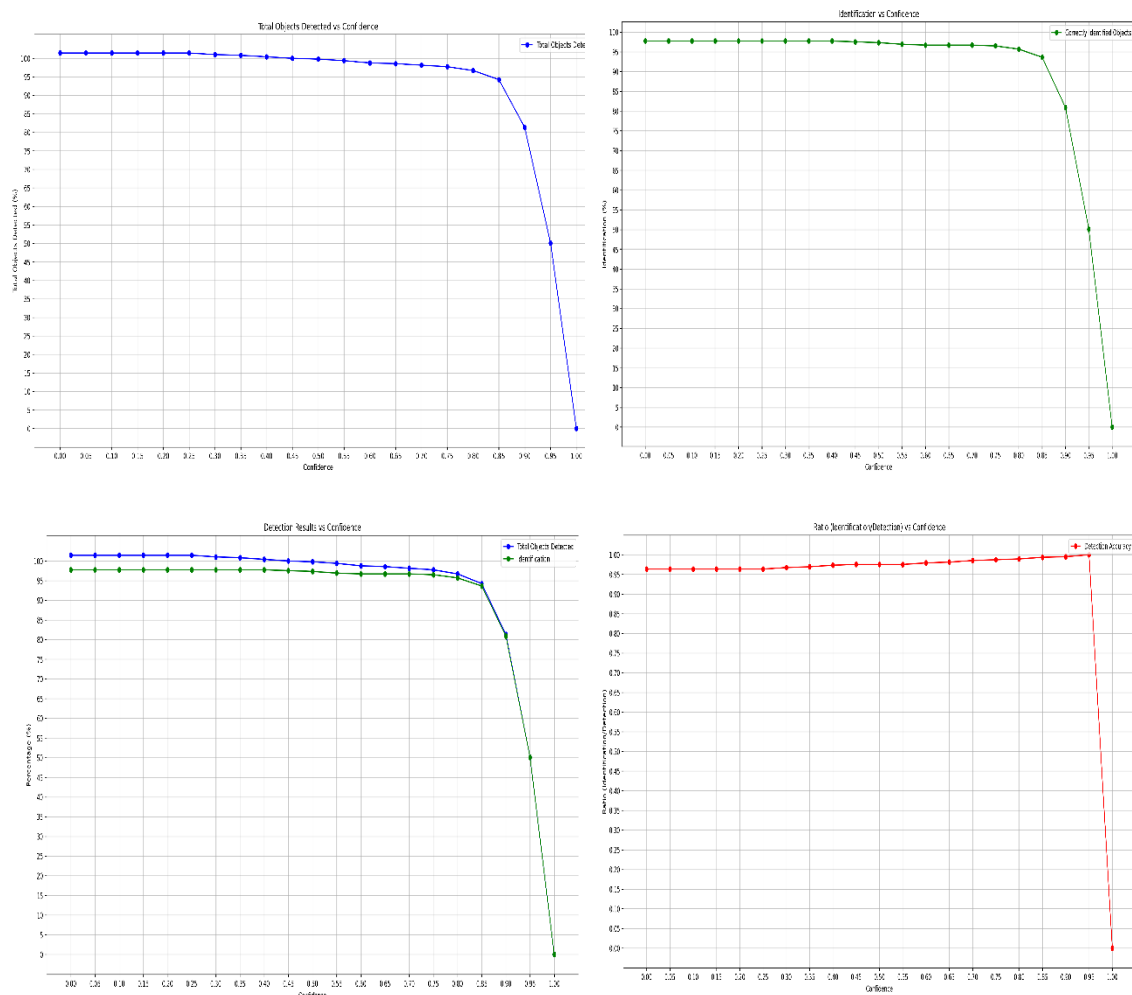


Figure 3.18 : Graphiques de Résultats pour le Modèle YOLOv11.

Le modèle YOLOv11l, tel que représenté dans les graphiques, démontre une performance stable en matière de détection et d'identification des objets sur une large plage de valeurs de confiance, allant jusqu'à environ 0.80. Dans cette zone, le pourcentage d'objets détectés et correctement identifiés reste proche de 100 %, indiquant une grande fiabilité du modèle. Au-delà de ce seuil, une chute rapide des performances est observée, notamment dans l'identification des objets, ce qui montre que le modèle devient plus strict et rejette davantage d'objets avec des scores de confiance plus faibles. Le graphique du bas à droite montre que le rapport identification/détection reste très élevé (près de 1) tant que le nombre d'objets détectés est suffisant, assurant ainsi une cohérence entre détection et identification. Ce comportement montre que YOLOv11l est un modèle précis et rigoureux, bien adapté aux tâches où l'on souhaite contrôler finement la qualité des détections en ajustant le seuil de confiance.

5. Modèle YOLOv11x

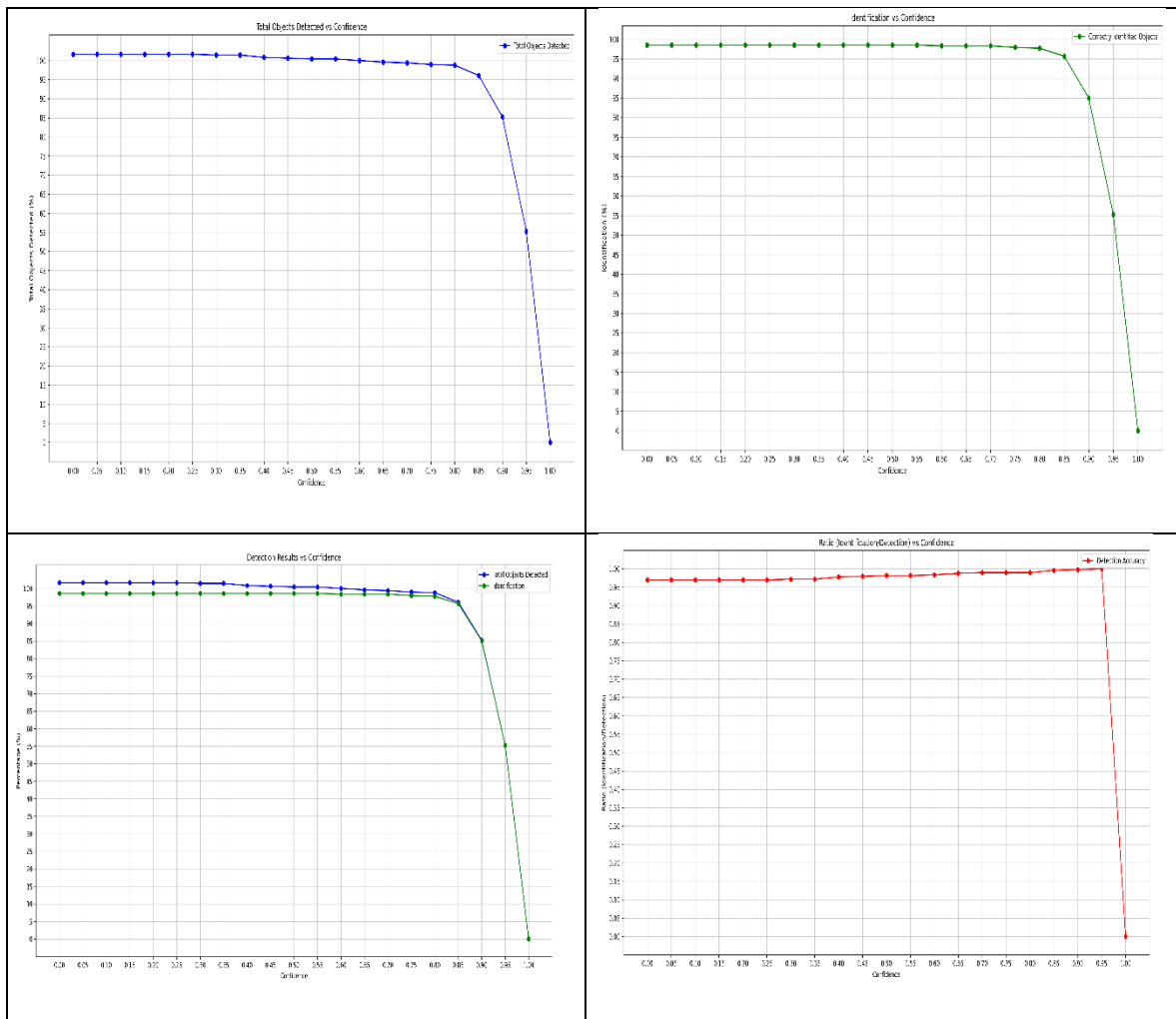


Figure 3.19: Graphiques de Résultats pour le Modèle YOLOv11x.

Le modèle YOLOv11x, illustré dans les graphiques, présente une grande stabilité en matière de détection et d'identification des objets sur une large plage de valeurs de confiance. Jusqu'à un seuil d'environ 0.90, le modèle détecte et identifie presque 100 % des objets avec une grande précision. Les courbes montrent que les performances restent constantes, sans dégradation significative, ce qui témoigne d'un comportement robuste face à l'augmentation du seuil de confiance. Ce n'est qu'au-delà de 0.95 que l'on observe une chute nette des résultats, indiquant un filtrage plus sévère. Le ratio identification/détection reste très élevé ( $\approx 1$ ) sur toute la plage fonctionnelle, ce qui indique une forte cohérence entre les objets détectés et ceux correctement identifiés. Ce modèle est donc très fiable et précis, capable de maintenir une excellente performance même sous des critères de confiance exigeants, ce qui en fait un choix pertinent pour des applications où la rigueur et la justesse des résultats sont prioritaires.

**Tableau 3.4** : Comparaison des modèles YOLOv11 (n, s, m, l, x)

Modèle	Taille / Vitesse	Stabilité jusqu'à seuil (%)	Précision (identification/détection)	Comportement à haut seuil	Usage recommandé
YOLOv11n	Très petit / Très rapide	Jusqu'à 80 %	Près de 1 jusqu'à 85 %	Forte chute après 80 %	Appareils légers, détection rapide
YOLOv11s	Petit / Rapide	Jusqu'à 80–85 %	Près de 1 jusqu'à 85 %	Chute marquée dès 85 %	Temps réel avec bonne précision
YOLOv11m	Moyen / Équilibré	Jusqu'à 80 %	Près de 1 jusqu'à 85 %	Chute rapide après 85 %	Bon compromis précision/vitesse
YOLOv11l	Grand / Plus lent	Jusqu'à 80 %	Excellente stabilité jusqu'à 80 %	Dégradation progressive	Tâches avec bonne puissance de calcul
YOLOv11x	Très grand / Plus lent	Jusqu'à 90–95 %	Très stable ( $\approx 1$ ) jusqu'à 95 %	Chute tardive et brutale	Haute précision, environnements critiques

En tenant compte de l'ensemble des résultats, YOLOv11n émerge comme le meilleur choix pour la détection automatique de panneaux dans un contexte temps réel et embarqué. Bien

qu'il soit le modèle le plus léger, il offre une accuracy exceptionnelle (jusqu'à 92.8 % à un seuil de 0.9) et une précision élevée (99.3 %), surpassant certaines variantes plus lourdes en termes d'efficacité globale.

Sa rapidité d'exécution, sa faible consommation de ressources et ses performances stables en font une solution idéale pour des applications temps réel, telles que les interfaces graphiques interactives, les systèmes embarqués à ressources limitées ou les prototypes embarqués sur caméra ou microcontrôleur.

Parmi toutes les variantes, YOLOv11n se distingue par son équilibre optimal entre légèreté, précision et rapidité, ce qui en fait le modèle le plus adapté à notre projet. En deuxième position, YOLOv11s reste un bon choix intermédiaire, tandis que YOLOv11m et YOLOv11l, bien que plus précis, s'avèrent trop coûteux en ressources. Enfin, YOLOv11x, bien qu'excellent sur le plan de la précision pure, est trop lourd pour les contraintes de notre application ciblée.

### 3.7 Conclusion

Ce chapitre a présenté l'ensemble du travail expérimental réalisé pour développer un système de détection automatique des panneaux de signalisation. À travers l'utilisation de Roboflow pour le traitement des données, Google Colab pour l'entraînement des modèles, et Tkinter pour l'interface utilisateur, nous avons mis en place un pipeline complet et fonctionnel de détection.

Une analyse comparative approfondie des variantes de YOLOv11 (n, s, m, l, x) a été menée, en nous appuyant sur des métriques telles que la précision, l'accuracy, ainsi que les matrices de confusion (TP, FP, FN, TN). Des graphiques explicites ont permis de visualiser les performances de chaque modèle, notamment en termes de prédictions correctes, manquées et incorrectes.

Les résultats ont clairement montré que YOLOv11n présente le meilleur compromis entre efficacité, rapidité et fiabilité. Avec une accuracy de 92,8 % et une précision de 99,3 % à un seuil de confiance élevé, ce modèle léger s'est révélé particulièrement adapté aux applications embarquées et en temps réel.

Par conséquent, YOLOv11n a été retenu comme modèle final pour notre système de détection.

---

## ***Conclusion générale***

---

Ce mémoire a présenté le développement d'un système intelligent de détection automatique des panneaux de signalisation basé sur les réseaux de neurones convolutifs et l'architecture YOLOv11. À travers une approche rigoureuse, nous avons d'abord procédé à la préparation des données, au prétraitement des images, puis à l'entraînement et à l'évaluation de plusieurs variantes du modèle.

Le choix de YOLOv11n s'est révélé particulièrement pertinent, grâce à sa capacité à allier traitement en temps réel, précision fiable et faible charge computationnelle. L'étude comparative des différentes versions du modèle (YOLOv11n, s, m, l, x) a mis en évidence les compromis essentiels entre vitesse d'exécution, précision de détection et robustesse face à **des** perturbations visuelles telles que le bruit, le flou ou les variations d'éclairage. Bien que les versions plus avancées offrent de très bonnes performances, leur complexité les rend peu adaptées à des contextes embarqués. YOLOv11n, quant à lui, s'est montré particulièrement efficace et stable, ce qui en fait le modèle final retenu pour notre système.

Les résultats expérimentaux ont démontré que la solution proposée permet de **détecter** et d'identifier avec fiabilité les panneaux de signalisation, même dans des environnements complexes. Cela met en évidence la pertinence de l'approche combinant **vision** par ordinateur, traitement d'image et apprentissage profond.

Ce travail ouvre la voie à des applications concrètes dans les systèmes embarqués, notamment dans les systèmes d'aide à la conduite, où la rapidité et la fiabilité sont des enjeux majeurs. Il pourra être enrichi par des perspectives telles que l'optimisation du modèle pour des plateformes à très faibles ressources, l'intégration de la reconnaissance de texte, ou encore l'extension de la détection à d'autres objets routiers dans le contexte de la conduite autonome.

## Bibliographique

- [1] Ministère des Transports et de la Mobilité durable du Québec. (2020). *Fiche de connaissances de base sur la signalisation routière*.  
<https://www.transports.gouv.qc.ca/fr/securite-signalisation/signalisation/Documents/fiche-connaissances-base.pdf>
- [2] [https://www.ornikar.com/code/cours/panneaux?srsId=AfmBOoqkSHspeioFJbOQShG1UJxz-ofrzVtfZuwg31\\_GdEtAzIM0IHUN](https://www.ornikar.com/code/cours/panneaux?srsId=AfmBOoqkSHspeioFJbOQShG1UJxz-ofrzVtfZuwg31_GdEtAzIM0IHUN) consulté le 2 mai 2025 à 09 :15
- [3] PermisAPoints.(2020,March9).*Sensinterdit:amendeetpertedepoints*.  
<https://www.permisapoints.fr/infractions/en-circulation/sens-interdit>
- [4] Codedelaroute.io.(2022,January11).*LaSignificationdesPanneauxdeSignalisation Routière*.  
<https://codedelaroute.io/blog/panneaux/#:~:text=signification%20des%20couleurs-1.,l'usage%20de%20la%20route> consulté le 2 mai 2025 à 09 :30
- [5] Codedelaroute.io. (2022, 11 janvier). *La Signification des Panneaux de Signalisation Routière*. <https://codedelaroute.io/blog/panneaux/> consulté le 2 mai 2025 à 09 :45
- [6] Satchoum. (2008, 3 mars). *Contes pour enfants – Jeux éducatifs de Tibous – Jeu des 5 familles de panneaux de signalisation*. Tibous. <https://tibous.over-blog.com/article-17292110.html> consulté le 2 mai 2025 à 10 :00
- [7] Panoloc.(n.d.).*TypeB–Signauxd’interdiction*.Consultéle4mai2025,àl'adresse  
<https://panoloc.fr/boutique/panneaux-de-signalisation/panneaux-permanents/type-b-signaux-dinterdiction/>
- [8] Codedelaroute.io. (2022, 9 mai). *Panneau de Danger | Tout Savoir pour Réussir le Code*.  
<https://codedelaroute.io/blog/panneaux-danger/> consulté le 2 mai 2025 à 10 :30
- [8] Ornikar. (2022, 9 mai). *Les différents panneaux de danger*.  
<https://www.ornikar.com/code/cours/panneaux/danger> consulté le 3 mai 2025 à 11 :00
- [9] Codedelaroute.io. (2023, 1er février). *Panneau de Priorité : Reconnaître la Signalisation*.  
<https://codedelaroute.io/blog/panneau-priorite/> consulté le 3 mai 2025 à 13 :15
- [10] Ornikar. (n.d.). *Panneaux de fin d'interdiction : signification*. Consulté le 7 mai 2025, à l'adresse <https://www.ornikar.com/code/cours/panneaux/interdiction/fin> consulté le 3 mai 2025 à 14 :15
- [11] Panoloc.(n.d.).*TypeB–Signauxd’obligation*.Consultéle7mai2025,àl'adresse  
<https://panoloc.fr/boutique/panneaux-de-signalisation/panneaux-permanents/type-b-signaux-dobligation/>  
 consulté le 3 mai 2025 à 15 :15

- [12] Ornikar. (n.d.). *Les panneaux de danger temporaire*. Consulté le 7 mai 2025, à l'adresse <https://www.ornikar.com/code/cours/panneaux/danger/temporaire>
- [13] Pixers. (n.d.). *Sticker Panneaux de signalisation temporaire*. Consulté le 7 mai 2025, à l'adresse <https://pixers.fr/stickers/panneaux-de-signalisation-temporaire-33178224>
- [14] Alpha Signa. (2024, 23 juillet). *Signalisation routière : couleurs et signification*. Consulté le 8 mai 2025, à l'adresse [https://alphasigna.com/tvcmsblog/post/29/signalisation-routiere-couleurs-et-signification.html?page\\_type=post](https://alphasigna.com/tvcmsblog/post/29/signalisation-routiere-couleurs-et-signification.html?page_type=post)
- [15] Wikipédia. (n.d.). *Panneau routier d'indication d'un lieu aménagé pour le stationnement gratuit en France*. Consulté le 8 mai 2025, à l'adresse [https://fr.wikipedia.org/wiki/Panneau\\_routier\\_d%27indication\\_d%27un\\_lieu\\_aménagé\\_pour\\_le\\_stationnement\\_gratuit\\_en\\_France](https://fr.wikipedia.org/wiki/Panneau_routier_d%27indication_d%27un_lieu_aménagé_pour_le_stationnement_gratuit_en_France)
- [16] Wikipédia. (n.d.). *Panneau d'indication d'une route à accès réglementé en France*. Consulté le 9 mai 2025, à l'adresse [https://fr.wikipedia.org/wiki/Panneau\\_d%27indication\\_d%27une\\_route\\_à\\_accès\\_réglementé\\_en\\_France](https://fr.wikipedia.org/wiki/Panneau_d%27indication_d%27une_route_à_accès_réglementé_en_France)
- [17] Khamadja, M., & Benierbah, S. (n.d.). *Traitement d'images (Version 1.0)*. Université des Frères Mentouri Constantine 1, Faculté des Sciences de la Technologie, Département d'Électronique, Master Systèmes des Télécommunications. Consulté le 9 mai 2025, à l'adresse <https://fac.umc.edu.dz/fstech/cours/Electronique/Master%20ST%C3%A9lectronique.com/CoursImageProcessing1.pdf>
- [18] Académie de Toulouse. (n.d.). *Photofiltre 7 : Résolution, définition d'une image*. Consulté le 10 mai 2025, à l'adresse <https://edu1d.ac-toulouse.fr/politique-educative-31/site-ressources31/files/04-photofiltre7-resolution-definition-simplifie.pdf>
- [19] Hadj Abdelkader, K., & Bounouahine, B. (2008). *Réduction du bruit des images par filtrage spatial* (Mémoire de fin d'études, Université Saad Dahleb de Blida, Faculté des sciences de l'ingénieur, Département d'Aéronautique). Consulté le 10 mai 2025, à l'adresse <https://di.univ-blida.dz/jspui/bitstream/123456789/4147/1/004-2008.pdf>
- [20] Chavel, P. H., Kuan, D. T., Sawchuk, A. A., & Strand, T. C. (1984). *Techniques de réduction de speckle*. *Journal of the Optical Society of America*, 1(1), 31–42.

<https://doi.org/10.1364/JOSA.1.000031>

- [21] Université PSL. (n.d.). *Résolution et DPI/PPP*. Consulté le 10 mai 2025, à l'adresse <https://explore.psl.eu/fr/se-former/tutoriels/resolution-et-dpippp>
- [22] <https://www.youtube.com/watch?v=wPlf2gdJANY> consulté le 12 mai 2025 à 14 :23
- [23] TheCodingMachine.(2024,17décembre).*Reconnaissanceoptiquedecaractères(OCR), quelle technologie choisir en 2024 ?* Consulté le 12 mai 2025, à l'adresse <https://thecodingmachine.com/reconnaissance-optique-de-caracteres-ocr-2024/>
- [24] Mahajan, A. (2023, 28 octobre). *EasyOCR : un guide complet*. Medium. Consulté le 12 mai 2025, à l'adresse <https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168>
- [25] Roboflow. (2024, 16 octobre). *Comment utiliser EasyOCR*. Consulté le 13 mai 2025, à l'adresse <https://blog.roboflow.com/how-to-use-easyocr/>
- [26] Mubeen,S.,Brahmani,J.,Kalyan,D. P., Jagirdar,A.,&Kumar,A. P. (2022,novembre). *Optical Character Recognition Using Tesseract. International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 10(11), 672–675. <https://doi.org/10.22214/ijraset.2022.47414>
- [27] Rosebrock, A. (2020, 25 mai). *Tesseract OCR : localisation et détection de texte*. PyImageSearch. Consulté le 16 mai 2025, à l'adresse <https://pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/>
- [28] I.Goodfellow,Y.Bengio,andA.Courville,\*DeepLearning\*.MITPress,2016.
- [29] S.Haykin,\*NeuralNetworksandLearningMachines\*,3rded.,Pearson,2009.
- [30] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", *\*Psychological Review\**, vol. 65, no. 6, pp. 386–408, 1958.
- [31] C.M.Bishop,\*PatternRecognitionandMachine Learning\*.Springer,2006. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *\*NeurIPS\**, 2012.
- [32] Maurice, B. (2018, 15 septembre). *Fonctionnement du neurone artificiel*. Deeply Learning. Consulté le 20 mai 2025, à l'adresse <https://deeplylearning.fr/cours-theoriques-deep-learning/fonctionnement-du-neurone-artificiel/>
- [33] F. Rosenblatt, « The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain », *Psychological Review*, vol. 65, no 6, pp. 386–

- 408, 1958.
- [34] S. Russel et P. Norvig, *Artificial Intelligence: A Modern Approach*, 3<sup>e</sup> éd., Upper Saddle River, NJ, USA: Pearson, 2010, pp. 734–736.
- [35] Jaiswal, S. (2025, 5 avril). *Perceptrons multicouches dans l'apprentissage automatique : un guide complet*. DataCamp. Consulté le 21 mai 2025, à l'adresse <https://www.datacamp.com/fr/tutorial/multilayer-perceptrons-in-machine-learning>
- [36] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY, USA: Springer, 2006. [En ligne]. Disponible : <https://doi.org/10.1007/978-0-387-45528-0> Consulté le 22 mai 2025 à 15 :00
- [37] Boughareb, D. (2014, juillet). *Architecture d'un réseau de neurones artificiels* [Figure C.2]. Dans *Mémoire : Recherche d'information multicritères* (Figure C.2). Consulté le 22 mai 2025, à l'adresse ResearchGate
- [39] S. Haykin, *Neural Networks and Learning Machines*, 3<sup>e</sup> éd., Upper Saddle River, NJ, USA: Pearson Education, 2009. Consulté le 22 mai 2025 à 16 :20
- [40] Jaiswal, S. (2025, 5 avril). *Perceptrons multicouches dans l'apprentissage automatique : Un guide complet*. DataCamp. Consulté le 22 mai 2025, à l'adresse <https://www.datacamp.com/fr/tutorial/multilayer-perceptrons-in-machine-learning>
- [41] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », *arXiv preprint arXiv:1409.1556*, 2014.
- [42] Vakalopoulou, M., Christodoulidis, S., Burgos, N., Colliot, O., & Lepetit, V. (2023). *Figure 18. A basic CNN architecture* [Illustration]. Dans *Machine Learning for Brain Disorders* (Chapitre 3). Humana. Consulté le 7 juin 2025, à l'adresse NCBI Bookshelf (id NBK597497)
- [43] Trehan, D. (2020, 24 juillet). *Convolutional Neural Networks for Dummies. Towards AI*. Consulté le 22 mai 2025, à l'adresse <https://pub.towardsai.net/convolutional-neural-networks-for-dummies-afd7166cd9e?gi=d02d7cc41b77>
- [44] Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », *Nature*, vol. 521, no 7553, pp. 436–444, 2015. [En ligne]. Consulté le 22 mai 2025, Disponible : <https://doi.org/10.1038/nature14539>
- [45] DataScientest. *Convolutional Neural Network (CNN): qu'est-ce que c'est?* Disponible en ligne : <https://datascientest.com/convolutional-neural-network> (consulté le 7 juin 2025).

- [46] Raut,A.S.(2023,13mars).*PaddinginNeuralNetworks:WhyandHow?* GoPenAI.  
<https://blog.gopenai.com/padding-in-neural-networks-why-and-how-b076ab0a4fc2>  
 Consulté le 23 mai 2025 à 9 :20
- [47] N.MurrayetF.Perronnin,«Generalizedmaxpooling»,dans *Proc.IEEEConferenceon Computer Vision and Pattern Recognition*, 2014, pp. 2473–2480. Consulté le 22 mai 2025 à 12 :30
- [48] GeeksforGeeks.(2025,2avril).*CNN/IntroductiontoPoolingLayer*.  
<https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>  
 Consulté le 23 mai 2025 à 10 :00
- [49] GeeksforGeeks.(2025,2avril).*CNN/IntroductiontoPoolingLayer*.Disponibleen ligne : <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>
- [50] Ayachi,R.,Said,Y.,&Atri,M.(2018).*StudyandImplementationofDeepLearning Algorithms for Object Detection*. Thèse de doctorat, Université de Monastir, Tunisie. Disponible en ligne : [https://www.researchgate.net/figure/fully-connected-layers\\_fig22\\_337447566](https://www.researchgate.net/figure/fully-connected-layers_fig22_337447566). Consulté le 21 mai 2025 à 10 :05
- [51] Goodfellow,I.,Bengio,Y.,&Courville,A.(2016). *DeepLearning*.MITPress.  
<https://www.deeplearningbook.org> Consulté le 24 mai 2025 à 12 :10
- [52]S.Haykin,*NeuralNetworksandLearningMachines*,3eéd.,PearsonEducationIndia, 2009. Consulté le 24 mai 2025 à 12:51
- [53] I.Goodfellow,Y.Bengio,etA.Courville,*DeepLearning*,MITPress,2016.[Enligne]. Disponible:<https://www.deeplearningbook.org/> Consulté le 24 mai 2025 à 13 :50
- [54] X.Glorot,A.Bordes,etY.Bengio,«DeepSparse RectifierNeuralNetworks»,dans *Proc.14thInternationalConferenceonArtificialIntelligenceandStatistics(AISTATS)*, 2011, pp.315–323.[Enligne].Disponible :<https://proceedings.mlr.press/v15/glorot11a.html> Consulté le 24 mai 2025 à 12 :40
- [55]A.L.Maas,A.Y.Hannun,andA.Y.Ng,"Rectifiernonlinearitiesimproveneural network acoustic models," in *Proc. ICML*, 2013. Consulté le 25 mai 2025 à 11 :10
- [56] Feng, J., He, X., Teng, Q., Li, Y., &al. (2019). *Reconstruction of porous media*

*from extremely limited information using conditional generative adversarial networks. Scientific Reports*, 9(1), 13797. <https://doi.org/10.1038/s41598-019-50345-1> Consulté le 25 mai 2025 à 14 :10

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788. Consulté le 26 mai 2025 à 12 :10

[58]

Ultralytics.(2024).*Ultralytics YOLO11*. Disponible en ligne : <https://docs.ultralytics.com/fr/models/yolo11/> Consulté le 26 mai 2025 à 12 :50

[59] J.Redmon,S.Divvala,R.Girshick,etA.Farhadi,«YouOnlyLookOnce:Unified, Real-Time Object Detection », dans *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [En ligne]. Disponible : <https://arxiv.org/abs/1506.02640> Consulté le 26 mai 2025 à 13 :10

[60] J. Redmon, S. Divvala, R. Girshick, et A. Farhadi, «*You Only Look Once: Unified, Real Time Object Detection*», dans *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [En ligne]. Disponible : <https://arxiv.org/abs/1506.02640> Consulté le 26 mai 2025 à 15 :10

[61] J. Redmon, A. Farhadi, «*YOLO9000: Better, Faster, Stronger*», dans *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271. [En ligne]. PDF sur le site de la conférence CVPR 2017 (YOLOv2 inaugurée dans cet article) Consulté le 26 mai 2025 à 15 :30

[62] J.Redmon,A.Farhadi,«*YOLOv3:AnIncrementalImprovement*»,arXiv,2018.[En ligne]. Disponible : arXiv:1804.02767 (version PDF original) Consulté le 26 mai 2025 à 12 :15

[63]A.Bochkovskiy,C.-Y.Wang,H.-Y.M.Liao,«*YOLOv4:OptimalSpeedandAccuracy of Object Detection* », arXiv, 2020. [En ligne]. Disponible : arXiv:2004.10934 Consulté le 27 mai 2025 à 10 :00

- [64] G.Jocher(*Ultralytics*),«*YOLOv5byUltralytics*»,logiciel,2020.[Enligne].DOI: 10.5281/zenodo.3908559 , Consulté le 30 mai 2025 à 11 :10
- [65]C.Li,L.Li,H.Jiang,etal.,«*YOLOv6:ASingle-StageObjectDetectionFrameworkfor Industrial Applications* », arXiv, 2022. [En ligne]. Disponible : arXiv:2209.02976 Consulté le 30 mai 2025 à 9 :30
- [66] C.-Y.Wang,A.Bochkovskiy,H.-Y.M.Liao,«*YOLOv7:Trainablebag-of-freebiesets new state-of-the-art for real-time object detectors* », arXiv/CVPR, 2022/2023. [En ligne].  
Disponible: arXiv:2207.02696 Consulté le 30 mai 2025 à 11 :10
- [67] M.Yaseen,«*What isYOLOv8:AnIn-DepthExplorationofthe InternalFeaturesofthe Next-GenerationObjectDetector*»,arXivpreprintarXiv:2408.15857,août2024.[Enligne].  
Disponible : <https://arxiv.org/abs/2408.15857> Consulté le 5 juin 2025 à 10 :10
- [68] Chien-YaoWangetH.-Y.M.Liao,«*YOLOv9:LearningWhatYouWanttoLearnUsing Programmable Gradient Information* », arXiv preprint arXiv:2402.13616, 2024, 1page.  
[En ligne]. Disponible : <https://arxiv.org/abs/2402.13616> Consulté le 5 juin 2025 à 11 :30
- [69] AoWang,HuiChen,LihaoLiu,KaiChen,ZijiaLin,JungongHan,etGuiguangDing, «*YOLOv10:Real-TimeEnd-to-EndObjectDetection*»,arXivpreprintarXiv:2405.14458, 2024, (NeurIPS 2024). [En ligne]. Disponible : <https://arxiv.org/abs/2405.14458> Consulté le 6 juin 2025 à 13 :17
- [70] Rahima Khanam et MuhammadHussain, « *YOLOv11: An Overview of the Key ArchitecturalEnhancements*»,arXivpreprintarXiv:2410.17725,23octobre2024.[Enligne].  
Disponible : <https://arxiv.org/abs/2410.17725> Consulté le 8 juin 2025 à 10 :13

[71] R. Khanam et M. Hussain, « *YOLOv11 : Aperçu des principales améliorations architecturales* », *arXiv preprint arXiv:2410.17725*, 23 octobre 2024. [En ligne]. Disponible : <https://arxiv.org/abs/2410.17725> Consulté le 8 juin 2025 à 09 :10.

[72] Rao, S.N. (2024, 22 octobre). *YOLOv11 Explained: Next-Level Object Detection with Enhanced Speed and Accuracy*. Medium. <https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71> Consulté le 7 juin 2025 à 10 :23.

[73] Yuanyi Zhong, Jianfeng Wang, Jian Peng, et Lei Zhang, « *Anchor Box Optimization for Object Detection* », *arXiv preprint arXiv:1812.00469*, 2 décembre 2018. [En ligne]. Disponible : <https://arxiv.org/abs/1812.00469> Consulté le 7 juin 2025 à 09 :17.

[74] Scikit-learn, « *Model Evaluation Documentation* », disponibles sur : [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html), consulté le 12 juin 2025 à 14 :20.

[75] Wainstein, D. (2020, 29 juillet). *Understanding the Confusion Matrix*. Medium. Disponible en ligne : <https://medium.com/@danyal.wainstein1/understanding-the-confusion-matrix-b9bc45ba2679> Consulté le 14 juin 2025 à 10 :12.