



Mémoire de Master

Filière Électronique
Spécialité Électronique Des Systèmes Embarqués

Présenté par

EULMI Kahina

&

MOUZAOUI Abdallah

Détection et identification de plaques d'immatriculation de véhicules en utilisant l'apprentissage profond

Proposé par : -Pr. Namane Abderrahmane

-Dr. Aouinane Bilel

Année Universitaire 2024-2025

Dédicace

Je dédie ce travail de fin d'études :

À moi-même,

À ma famille,

À Ma chère mère,

À Mon cher père,

À mes chères sœurs,

À mon binôme,

À toute ma promotion,

À tous mes amis,

*À tous ceux qui ont contribué, de près ou de loin, à
l'accomplissement de ce travail.*

Eulmi Kahina

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,

À la mémoire de mon père,

Tu n'es plus là pour partager avec moi cette étape importante, mais chaque mot de ce travail, chaque effort, chaque réussite, porte l'empreinte de ce que tu m'as appris. Ton courage, ta sagesse, ton humilité et ton amour ont façonné la personne que je suis devenue. Tu as semé en moi les graines de la persévérance et du respect, que je cultive encore aujourd'hui avec fierté.

Repose en paix, Papa. Ce travail est pour toi.

À ma mère,

Je tiens à exprimer ma profonde gratitude à ma mère,

Celle qui a toujours été mon pilier et qui m'a soutenu avec amour, patience et courage.

Merci pour ta présence, tes sacrifices et ton inspiration au quotidien.

À ma famille,

À toute ma famille, ma source de force, d'amour et d'équilibre.

Je dédie ce travail à ceux qui m'ont entouré, soutenu et encouragé à chaque étape de ce parcours, vos prières et votre patience ont été pour moi un refuge et une inspiration. Dans les moments de doute comme dans ceux de joie, vous avez toujours été là.

Merci à chacun d'entre vous frères, sœurs, oncles, tantes, cousins, cousines pour votre amour inconditionnel. Ce succès est aussi le vôtre.

À ma femme,

À toi mon épouse, ton amour est ma force et ton sourire et ta présence mon équilibre je te dis merci d'avoir été à mes côtés à chaque étape de ce chemin

À mes amies et mon binôme,

Pour leur écoute, leur bonne humeur et leur amitié fidèle qui m'ont aidé à traverser les défis avec sérénité. 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.

Mouzaoui abdallah

Remerciements

En préambule à ce mémoire, nous remercions Allah, le Tout-Puissant et Miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à exprimer notre profonde gratitude à notre promoteur, le Professeur NAMANE Abderrahmane, qui nous a fait l'honneur de diriger ce travail. Nous le remercions sincèrement pour ses précieux conseils, sa disponibilité et son accompagnement tout au long de cette étude.

Nos remerciements s'adressent également à notre co-promoteur, Dr AOUIANE Bilel, pour son soutien, ses orientations pertinentes et l'attention qu'il a accordée à notre travail.

Nous remercions vivement tous les membres du jury, en particulier Dr REGUIEG et Dr ZAIR, pour leur précieux temps ainsi qu'à leur évaluation méticuleuse.

Ces remerciements vont également au corps professoral et administratif de l'Université de Blida 1, et plus particulièrement au département d'électronique, pour la richesse et la qualité de leur enseignement, ainsi que pour les efforts déployés afin de garantir une formation de haut niveau à leurs étudiants.

Nous n'oublions pas nos parents, pour leur soutien indéfectible, leur patience et leurs sacrifices tout au long de notre parcours.

Enfin, nous adressons nos plus sincères remerciements à nos proches et amis, qui nous ont toujours encouragés et soutenus au cours de la réalisation de ce

ملخص: في هذا المشروع، تم تطوير نظام ذكي مخصص للكشف والتعرف التلقائي على لوحات ترقيم المركبات انطلاقاً من الصور. يتم تنظيم العملية على عدة مراحل: كشف وتحديد موقع اللوحة بدقة، بالإضافة إلى الأحرف المكونة لها، تليها مرحلة التعرف التلقائي على هذه الأحرف. تم إنشاء قاعدة بيانات مكونة من صور حقيقية، وتم عنونها باستخدام أداة Roboflow سريعة وفعالة لتعليم الصور. يُستخدم نموذج YOLO كتعلم مُراقب للكشف عن الأجسام الموجودة في الصور. تم اقتراح وتدريب نموذج YOLO11(n,s,m,l,x) على منصة Google Colab، مما يسمح بتقييم مقارن من حيث السرعة والدقة. تم استكشاف استراتيجيتين: استخدام YOLO11 فقط للكشف والتعرف وحقت نسبة تعرف صحيحة بلغت 91.4%، ونهج هجين يجمع بين YOLO11 للكشف ووحدة OCR للتعرف (تسجيل نسبة تعرف بلغت 54.1% باستخدام Tesseract و 62.4% باستخدام EasyOCR).

كلمات المفاتيح: YOLO11، OCR، كشف لوحات أرقام المركبات، التعلم العميق، التعرف على الأحرف.

Résumé : Dans ce projet, un système intelligent destiné à la détection et à l'identification automatique des plaques d'immatriculation de véhicules à partir d'images a été développé. Le processus est structuré en plusieurs phases : détection et localisation précise de la plaque ainsi que les caractères que constituent cette plaque, suivi d'une reconnaissance automatique de ces caractères. Une base de données composé d'images réelles a été créé et annotée à l'aide de l'outil Roboflow, qui offre une interface rapide et efficace pour l'annotation d'images. Le modèle YOLO, basé sur l'apprentissage supervisé, a été utilisé pour détecter les objets présents dans les images. Le modèle YOLO11(n, s, m, l, x) est proposé et entraîné sur Google Colab, permettant une évaluation comparative en termes de rapidité et de précision. Deux stratégies sont explorées : l'utilisation exclusive de YOLO11 pour la détection et la reconnaissance (taux de reconnaissance atteignant 91,4 %), et une approche hybride combinant YOLO11 pour la détection et un module OCR pour la reconnaissance (taux de reconnaissance 54,1% pour Tesseract et 62,4% pour EasyOCR).

Mots clés : YOLO11 ; OCR ; Détection de plaques d'immatriculation ; Apprentissage profond ; Reconnaissance de caractères.

Abstract : In this project, an intelligent system was developed for the automatic detection and recognition of vehicle license plates from images. The process is structured into several phases: detection and precise localization of the plate, extraction of the characters it contains, followed by the automatic recognition of these characters. A real-world image dataset was created and annotated using the Roboflow tool, which provides a fast and efficient interface for image labeling. The YOLO model, based on supervised learning, was used to detect objects within images. Specifically, the YOLO11 model (in its n, s, m, l, x variants) was proposed and trained on Google Colab, enabling comparative evaluation in terms of speed and accuracy. Two strategies were explored: the exclusive use of YOLO11 for both detection and recognition(achieving recognition rate of 91.4%), and a hybrid approach combining YOLO11 for detection with an OCR module for recognition (reaching 54.1% with Tesseract and 62.4% with EasyOCR).

Keywords: YOLO11; OCR; License plate detection; Deep learning; Character recognition.

YOLO : You Only Look Once

YOLOv1 : You Only Look Once Version 1

YOLOv11 : You Only Look Once Version 11

YOLOv12 : You Only Look Once Version 12

OCR : Optical Character Recognition

ALPR : Automatic License Plate Recognition

ANPR : Automatic Number Plate Recognition

IA : Intelligence Artificielle

RNA : Reseaux Neuronaux Artificiels

R-CNN : Regions With Convolutional Neural Network

CNN : Convolutional Neural Network

ROI : Region Of Interest

SVM : Support Vector Machine

SPPF : Spatial Pyramid Pooling Fast

ReLU : Rectified Linear Unit

IOU : Intersection Over Union

NMS : Non-Maximum Suppression

OBB : Oriented Bounding Box

FPN : Feature Pyramid Network

PAFPN : Path Aggregation Feature Pyramid Network

C2F : Cross Stage Partial Avec 2 Convolutions Et Fusion De Caractéristiques

C3K2 : C3 block With Kernel Size 2

C2PSA : Convolutional Block With Parallel Spatial Attention

OpenCV : Open Source Computer Vision Library

GPU : Graphics Processing Unit

SSD : Solid-State Drive

Listes des acronymes et abréviations

RAM: Random Access Memory

NVIDIA: NVIDIA Corporation (entreprise spécialisée dans les GPU)

NumPy: Numerical Python

PyTorch: Python Torch (bibliothèque de deep learning basée sur Torch)

CTC: Connectionist Temporal Classification

CRNN: Convolutional Recurrent Neural Network

mAP: mean Average Precision

LSTM: Long Short-Term Memory

JPEG: Joint Photographic Expert Group

PNG: Portable Network Graphics

BMP: Bitmap Image File

HP: Hewlett-Packard (entreprise informatique)

YOLOv11n: You Only Look Once version 11 nano

YOLOv11s: You Only Look Once version 11 small

YOLOv11m: You Only Look Once version 11 medium

YOLOv11l: You Only Look Once version 11 large

YOLOv11x: You Only Look Once version 11 extra-large

TKinter: TK interface

Introduction générale.....	1
Chapitre 1 : Généralités sur les plaques d'immatriculation et leur détection par traitement d'image	
1.1 Introduction	3
1.2 Plaques d'immatriculation à travers le monde	3
1.2.1 Historique et évolution	3
1.2.2 Caractéristiques générales des plaques	4
1.2.3 Diversité des formats selon les pays	4
1.2.4 Contraintes et enjeux.....	5
1.3 Système d'immatriculation en Algérie	5
1.3.1 Structure d'une plaque algérienne	5
1.3.2 Problèmes spécifiques aux plaques algériennes.....	6
1.3.3 Enjeux de la reconnaissance automatique en Algérie.....	7
1.4 Techniques de traitement d'image pour la détection de plaques.....	7
1.4.1 Introduction au traitement d'image	7
1.4.2 Chaîne typique d'un système ALPR	8
1.4.3 Méthodes classiques de détection	10
1.4.4 Méthodes modernes basées sur l'IA.....	14
1.4.5 YOLO pour la détection de plaques.....	14
1.4.6 Problèmes rencontrés et techniques de traitement.....	15
1.4.7 YOLO11 pour la détection des plaques d'immatriculation	16
1.5 Conclusion.....	17
Chapitre 2 : Intelligence artificielle et détection de matricules	
2.1 Introduction.....	18
2.2 Apprentissage profond et réseaux de neurones convolutifs	18
2.2.1 Architecture de base du CNN	19
2.3 Algorithme YOLO	25
2.3.1 Architecture de YOLO	26
2.3.2 Processus de détection d'objets avec YOLO	27
2.3.3 Évolution des modèles YOLO.....	30
2.4 Algorithme YOLO11	31
2.4.1 Architecture de YOLO11	32
2.4.2 Déclinaisons Spécialisées de YOLO11.....	34
2.4.3 Déclinaisons dimensionnelles du YOLO11 : Versions n, s, m, l, x	35

2.4.4 Technique d'évaluation du model YOLO	36
2.5 Choix du YOLO11 pour la détection des plaques d'immatriculation.....	37
2.6 Conclusion.....	39
Chapitre 3 : Implémentation et Résultats	
3.1 Introduction	40
3.2 Environnement de travail.....	41
3.2.1 Poste de travail local.....	41
3.2.2 Plateforme d'entraînement Cloud (Google Colaboratory)	41
3.2.3 Logiciels et Langages	42
3.3 Préparation des données	42
3.3.1 Constitution des jeux de données	42
3.3.2 Annotation des images avec Roboflow	46
3.3.3 Organisation en ensembles d'entraînement, validation et test	48
3.4 Interface graphique avec tkinter	48
3.5 Choix du modèle YOLO11l : un compromis optimal entre précision et rapidité.....	49
3.5.1 Détection des plaques avec YOLO11l	54
3.5.2 Entraînement.....	54
3.5.3 Résultats obtenus et interprétations.....	55
3.6 Détection et reconnaissance des plaques avec YOLO11l.....	58
3.6.1 Entraînement.....	58
3.6.2 Résultats obtenus et interprétations.....	59
3.7 Détection des plaques avec YOLO11l et reconnaissance avec OCR	64
3.7.1 Reconnaissance avec EasyOCR	64
3.7.2 Reconnaissance avec Tesseract	65
3.8 Étude comparative.....	66
3.9 Conclusion	68
Conclusion Générale	69

Figure 1.1 : Apparition de la plaque d'immatriculation, France 1903	3
Figure 1.2 : Les plaques d'immatriculation à travers le monde	4
Figure 1.3 : Exemple de plaque d'immatriculation algérienne	5
Figure 1.4 : variabilité des plaques d'immatriculation algérienne	6
Figure 1.5 : Exemple de plaque d'immatriculation algérienne usée.	6
Figure 1.6 : Exemple de plaque d'immatriculation algérienne non conforme.....	6
Figure 1.7 : Schéma synoptique d'un système de vision artificielle.....	7
Figure 1.8 : Acquisition d'image par caméra.....	8
Figure 1.9 : Image avec et sans bruit.....	9
Figure 1.10 : Détection de plaque.	9
Figure 1.11 : Reconnaissance de matricule	10
Figure 1.12 : Méthodes classiques de détection.....	10
Figure 1.13 : Conversion RVB vers niveau de gris	11
Figure 1.14 : Application du filtre de GAUSS	11
Figure 1.15 : Application de l'algorithme de Canny	12
Figure 1.16 : Application de la binarisation locale	13
Figure 1.17 : Les caractères reconnus par OCR	14
Figure 1.18 : Processus de détection et reconnaissance de matricule avec YOLO.....	15
Figure 2.1 : Architecture de réseau de détection d'objets à base de CNN, appliquée à la détection de plaques d'immatriculation.	18
Figure 2.2 : Architecture des réseaux neuronaux convolutifs.....	19
Figure 2.3 : Couche de Convolution	20
Figure 2.4 : Exemple de la formule $\text{ReLU}(x) = \max(0, x)$	20
Figure 2.5 : Graphe de la fonction ReLU	21
Figure 2.6 : Graphe de la fonction d'activation Softmax	22
Figure 2.7 : Graphe de la fonction d'activation Sigmoid.....	23
Figure 2.8 : Couches de pooling.....	23
Figure 2.9 : Couches d'aplatissement (flattening).....	24
Figure 2.9 : Couche entièrement connectée.....	25
Figure 2.11 : Modèle de YOLO	25
Figure 2.12 : Architecture de YOLO	26
Figure 2.13 : Processus de détection d'objets avec YOLO	27
Figure 2.14 : Exemple d'une boîte englobante	28

Figure 2.15 : Modèle YOLO simplifié avec une grille de 3 x 3, trois classes, produit un vecteur de huit valeurs	28
Figure 2.16 : Deux objets détectés dans la même grille	29
Figure 2.17 : Calcul de l'Intersection over Union	30
Figure 2.18 : Application du NMS	30
Figure 2.19 : Evolution des modèles de YOLO à travers les années	31
Figure 2.20 : Architecture originale du YOLO11	32
Figure 2.21 : Architecture du modèle YOLO11	34
Figure 2.22 : YOLO11 offre plusieurs tâches de vision par ordinateur	35
Figure 2.23 : Graphique de comparaison des performances de YOLO11 avec différents précédents modèles	35
Figure 2. 24 : Structure de la matrice de confusion	36
Figure 2.25 : YOLO11 pour la détection des plaques d'immatriculation	39
Figure 3.1 : Image avec matricule incliné.....	44
Figure 3.2 : Image avec matricule arrière très proche et salle.....	44
Figure 3.3 : Image avec matricule en face.....	45
Figure 3.4 : Exemple avec faible luminosité.....	45
Figure 3.5 : Exemple avec forte luminosité.....	46
Figures 3.6 : Exemples des différentes polices.....	46
Figure 3.7 : Exemple annotation pour la détection seulement.....	47
Figure 3.8 : Exemple annotation pour la détection et reconnaissance.....	47
Figure 3.9 : Affichage de l'image originale et annotée dans l'interface Tkinter.....	49
Figure 3.10 :Taux de détections et reconnaissances correctes avec YOLO11n (255 images test).....	51
Figure 3.11 : Taux de détections et reconnaissances correctes avec YOLO11s (255 images test).....	51
Figure 3.12 : Taux de détections et reconnaissances correctes YOLO11m (255 images test).....	52
Figure 3.13 : Taux de détections et reconnaissances correctes avec YOLO11l (255 images test).....	53

Figure 3.14 : Taux de détections et reconnaissances correctes avec YOLO11x (255 images test).....	53
Figure 3.15 : Détection de la plaque avec YOLO11l	54
Figure 3.16 : Résultats obtenus avec colab.....	54
Figure 3.17 : Résultats obtenus avec YOLO11l pour la détection (une classe).....	55
Figure 3.18 : Matrice de confusion obtenu avec YOLO11l pour la détection (une classe).....	56
Figure 3.19 : Taux de détections correctes avec YOLO11l pour la détection avec 255 images test.....	57
Figure 3.20 : Détection et reconnaissance de la plaque avec YOLO11l.....	58
Figure 3.21 : Résultats obtenus avec colab.....	59
Figure 3.22 : Résultats obtenus avec YOLO11l pour la détection et reconnaissance (13 classes).....	59
Figure 3.23 : Matrice de confusion obtenu avec YOLO11l pour la détection (13 classes)	60
Figure 3.24 : Taux de détections et reconnaissances correctes avec YOLO11l avec 255 images test.....	61
Figure 3.25 : Taux de détections et reconnaissances correctes avec YOLO11l par rapport à la détection avec 255 images test.....	62
Figure 3.26 : Rapport des résultats de détections et reconnaissances correctes avec yolo11-L sur résultats de la détection avec YOLO11l (avec 255 images test).....	62
Figure 3.27 : Exemple de détection et reconnaissance de la plaque avec YOLO11l.....	63
Figure 3.28 : Détection des plaques avec YOLO11l et reconnaissance avec OCR.....	64

Tableau 3.1 : Résultats avec YOLO11l pour la détection et reconnaissance avec 255 images test.....	63
Tableau 3.2 : Résultats avec YOLO11l pour la détection et EasyOCR pour reconnaissance avec 255 images test.....	65
Tableau 3.3 : Résultats avec YOLO11l pour la détection et Tesseract pour reconnaissance avec 255 images test.....	66
Tableau 3.4 : Comparaison des 3 approches.....	67

Introduction générale

Introduction générale

Dans un monde en constante évolution, la gestion automatisée du trafic, la sécurité routière et la régulation des accès aux zones sensibles sont devenues des enjeux majeurs. Les systèmes de reconnaissance automatique des plaques d'immatriculation (Automatic License Plate Recognition, ALPR) constituent une solution technologique centrale dans ces domaines. En effet, les matricules des véhicules jouent un rôle crucial dans l'identification formelle des automobiles, permettant ainsi d'assurer un suivi fiable dans des contextes variés : péages automatiques, parkings intelligents, systèmes de vidéosurveillance, gestion de la circulation, ou encore contrôle des véhicules volés ou non assurés.

Le caractère unique de chaque plaque permet d'associer un véhicule à son propriétaire, à ses caractéristiques techniques ou à son historique administratif. Cette propriété rend la détection et la reconnaissance des plaques particulièrement précieuses pour les applications en sécurité publique et privée.

Notre projet s'inscrit dans cette évolution technologique. Il vise à développer une solution ANPR basée sur l'apprentissage profond et plus précisément sur le modèle YOLO11 (You Only Look Once). Le YOLO11 est une architecture de détection d'objets puissante et rapide, capable de détecter et d'identifier directement les plaques (avant et arrière) ainsi que les caractères qu'elles contiennent, en une seule passe. Cette approche unifiée permet de fusionner la détection et la reconnaissance dans un seul modèle, optimisant ainsi la rapidité du système, ce qui est indispensable pour une application embarquée ou temps réel.

Développer un système capable de localiser, segmenter et reconnaître automatiquement les caractères d'une plaque d'immatriculation dans des conditions réelles (variabilité d'éclairage, angles de vue, flou de mouvement, obstructions partielles, etc.) est un défi majeur. Cela nécessite la combinaison de techniques avancées en traitement d'image et en intelligence artificielle.

Pour renforcer la fiabilité du système, nous avons intégré une phase complémentaire de reconnaissance par OCR (easyocr et tesseract), permettant de comparer ou valider les caractères détectés par YOLO. Cette approche comparative hybride offre une double sécurité sur le résultat final, tout en améliorant les performances globales du système.

Le mémoire est structuré en trois chapitres suivis d'une conclusion générale :

- Le chapitre 1 : introduit les plaques d'immatriculation dans la vie réelle ainsi que les bases du traitement d'image, nécessaires à la localisation et à la préparation des plaques pour la reconnaissance.
- Le chapitre 2 : présente les concepts clés de l'IA, notamment les RNA (réseaux de neurones artificiels), le fonctionnement du modèle YOLO11 pour la détection en temps réel.
- Le chapitre 3 : expose le travail, de l'implémentation du système ANPR jusqu'aux résultats expérimentaux, avec une évaluation des performances et une interface logicielle développée.

Chapitre 1

Généralités sur les plaques d'immatriculation et leur détection par traitement d'image

1.1 Introduction

Les plaques d'immatriculation jouent un rôle crucial dans l'identification des véhicules, facilitant la gestion du trafic, la sécurité routière, et l'application des lois. Avec l'essor des technologies de traitement d'images et d'intelligence artificielle, la reconnaissance automatique des plaques d'immatriculation (Automatic License Plate Recognition - ALPR) est devenue une composante essentielle des systèmes de transport intelligents.

Ce chapitre présente une vue d'ensemble des systèmes d'immatriculation à travers le monde, avec un accent particulier sur le système algérien, et explore les techniques de traitement d'image utilisées pour la détection des plaques.

1.2 Plaques d'immatriculation à travers le monde

1.2.1 Historique et évolution

L'histoire des plaques d'immatriculation remonte à la fin du 19^e siècle. La France fut le premier pays à introduire un système d'immatriculation des véhicules en 1893, suivi par le Royaume-Uni en 1903 et l'Allemagne en 1906. Ces systèmes visaient à identifier les véhicules pour des raisons de sécurité et de réglementation [1] (Voir figure 1.1).



Figure 1.1 Apparition de la plaque d'immatriculation, France 1903 [2].

1.2.2 Caractéristiques générales des plaques

Les plaques d'immatriculation varient considérablement selon les pays, mais elles partagent des caractéristiques communes [15] (Voir figure 1.2):



Figure 1.2 Les plaques d'immatriculation à travers le monde [3].

- **Format** combinaisons alphanumériques uniques.
- **Disposition** une ou deux plaques par véhicule, généralement à l'avant et à l'arrière.
- **Couleurs** souvent un fond blanc ou jaune avec des caractères noirs, bien que cela puisse varier.
- **Éléments additionnels** certains pays incluent des symboles nationaux ou régionaux, des codes de pays, ou des hologrammes de sécurité.

1.2.3 Diversité des formats selon les pays

- **Europe** les pays de l'Union européenne ont adopté un format standardisé avec une bande bleue sur le côté gauche contenant le code du pays, suivi d'une combinaison alphanumérique.
- **Amérique du Nord** aux États-Unis et au Canada, chaque État ou province a son propre modèle de plaque, souvent avec des images ou des slogans spécifiques.
- **Asie** au Japon, les plaques sont codées par couleur pour indiquer le type de véhicule, tandis qu'en Chine, elles incluent des caractères chinois et des codes régionaux.
- **Afrique** la diversité est grande, avec des systèmes variant considérablement d'un pays à l'autre [15].

1.2.4 Contraintes et enjeux

Les systèmes d'immatriculation doivent répondre à plusieurs défis :

- **Sécurité** prévenir la falsification et l'utilisation frauduleuse des plaques.
- **Standardisation** assurer une reconnaissance facile par les systèmes automatisés.
- **Adaptabilité** accommoder les besoins spécifiques des différentes régions et types de véhicules.

1.3 Système d'immatriculation en Algérie

Le système d'immatriculation en Algérie permet d'identifier chaque véhicule de manière unique à l'aide d'une plaque normalisée. Il a évolué pour répondre aux exigences de gestion, de sécurité et de traçabilité. La structure de ces plaques suit un format bien défini [6].

1.3.1 Structure d'une plaque algérienne

En Algérie, le système d'immatriculation a évolué au fil du temps pour répondre aux besoins croissants en matière de gestion des véhicules. Des réformes ont été introduites pour moderniser le processus. Les plaques algériennes suivent un format spécifique [6] (Voir figure 1.3) :

- **Premier groupe** 5 chiffres, 6 chiffres au maximum, qui correspond au numéro de série du véhicule.
- **Deuxième groupe** composé de 3 chiffres : le premier chiffre permet d'identifier le type de véhicule (1 véhicule léger, 2 pour un poids lourd, 3 voiture commerciale, 4 bus, 5 tracteurs de semi-remorque, 6 tracteurs, 7 engins, 8 remorques, 9 motos), les deux suivants renvoient à l'année de mise en circulation du véhicule.
- **Troisième groupe** 2 chiffres qui identifient la wilaya d'immatriculation (De 01 à 58 actuellement).

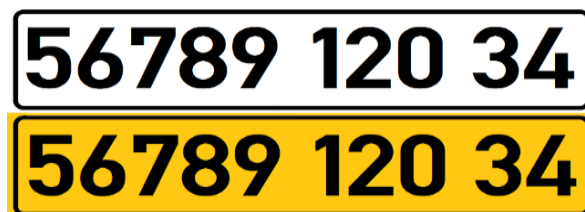


Figure 1.3 Exemple de plaque d'immatriculation algérienne [6].

1.3.2 Problèmes spécifiques aux plaques algériennes

Plusieurs défis sont associés aux plaques d'immatriculation en Algérie :

- **Variabilité** diversité dans la taille, la police et la qualité des plaques (Voir figure 1.4).



Figure 1.4 variabilité des plaques d'immatriculation algérienne [7].

- **Usure** plaques endommagées ou illisibles en raison des conditions climatiques ou du manque d'entretien (Voir figure 1.5).



Figure 1.5 Exemple de plaque d'immatriculation algérienne usée.

- **Non-conformité** utilisation de plaques non réglementaires ou falsifiées (Voir figure 1.6).



Figure 1.6 Exemple de plaque d'immatriculation algérienne non conforme [4].

1.3.3 Enjeux de la reconnaissance automatique en Algérie

Des initiatives sont en cours pour moderniser le système d'immatriculation et intégrer des technologies avancées. La mise en œuvre de systèmes ALPR en Algérie présente plusieurs objectifs [15] :

- **Sécurité routière** : identification rapide des véhicules en infraction.
- **Gestion du trafic** : surveillance en temps réel des flux de véhicules.
- **Lutte contre la fraude** : détection des plaques falsifiées ou non conformes.

1.4 Techniques de traitement d'image pour la détection de plaques

1.4.1 Introduction au traitement d'image

Dans le contexte des systèmes de reconnaissance automatique de plaques d'immatriculation (ALPR, *Automatic License Plate Recognition*), le traitement d'image joue un rôle fondamental. Les techniques classiques, telles que la conversion en niveaux de gris, la réduction du bruit par filtrage (Gaussian blur, median blur), le seuillage, la détection de contours (comme l'algorithme de Canny) ou les opérations morphologiques (érosion, dilatation), étaient historiquement utilisées pour localiser les plaques et segmenter les caractères (Voir figure 1.7).

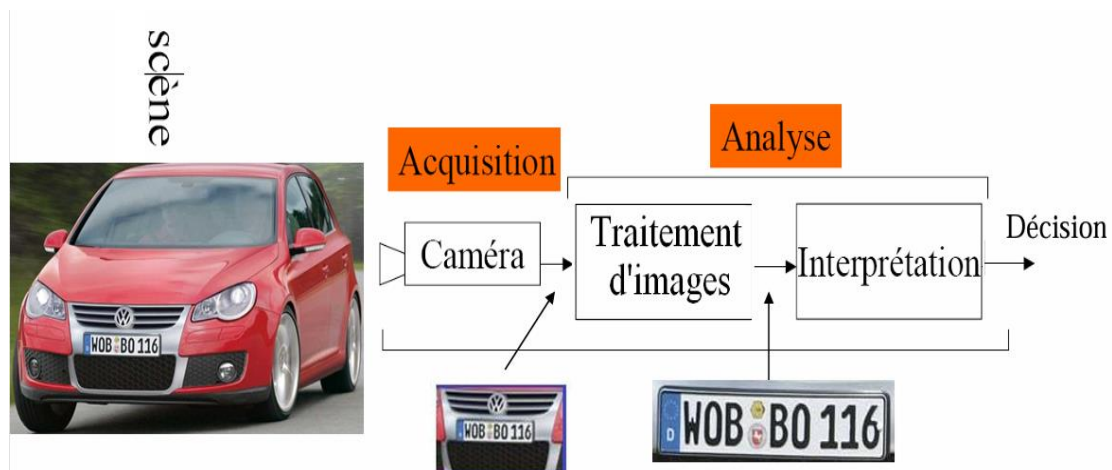


Figure 1.7 Schéma synoptique d'un système de vision artificielle [5].

1.4.2 Chaîne typique d'un système ALPR

Un système ALPR fonctionne en plusieurs étapes successives, allant de la capture d'image jusqu'à l'extraction des caractères de la plaque. Chaque étape joue un rôle essentiel pour assurer la précision du système, même dans des conditions variées. Un système ALPR standard comprend plusieurs étapes :

a. Acquisition de l'image Capturer des images ou vidéos des véhicules à l'aide de caméras, souvent équipées d'éclairage infrarouge pour fonctionner dans diverses conditions d'éclairage.

Les caméras peuvent être fixes (sur des portiques, feux de circulation) ou mobiles (embarquées sur des véhicules) (Voir figure 1.8).



Figure 1.8 Acquisition d'image par caméra [8].

b. Prétraitement Amélioration de la qualité de l'image capturée par des techniques telles que la conversion en niveaux de gris, la réduction du bruit, l'amélioration du contraste et la normalisation de l'intensité lumineuse. Ces étapes facilitent les processus ultérieurs de détection et de reconnaissance (Voir figure 1.9).



Figure 1.9 Image avec et sans bruit [9].

- c. Détection de la plaque** Identifier et localiser la plaque d'immatriculation dans l'image. Des algorithmes de détection d'objets, tels que YOLO ou Faster R-CNN, sont souvent utilisés pour cette tâche (Voir figure 1.10).



Figure 1.10 Détection de plaque.

- d. Reconnaissance des caractères** Après avoir localisé la plaque et extrait la région contenant les caractères, il est nécessaire d'identifier chaque caractère individuellement. Cette étape est cruciale pour convertir l'information visuelle en texte lisible par machine.
- Utiliser des algorithmes de Reconnaissance Optique de Caractères pour identifier les caractères segmentés.

- Des techniques basées sur l'apprentissage profond, comme les réseaux de neurones convolutifs (CNN), sont couramment employées (Voir figure 1.11).



Figure 1.11 Reconnaissance de matricule [11].

1.4.3 Méthodes classiques de détection

Les méthodes classiques de détection des plaques d'immatriculation suivent une chaîne de traitement d'image bien définie. Voici les principales étapes (Voir figure 1.12).

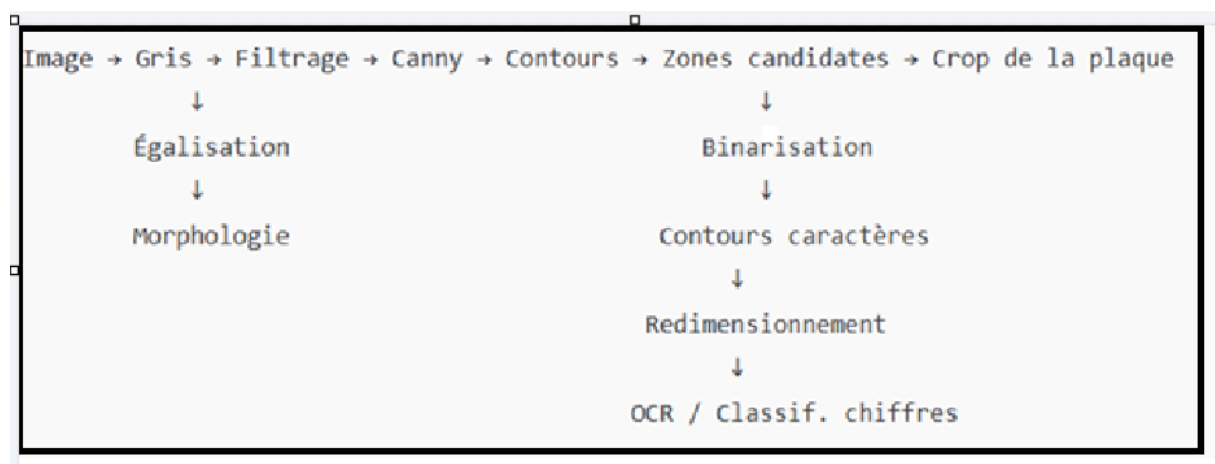


Figure 1.12 Méthodes classiques de détection.

- Conversion en niveaux de gris** Le traitement commence généralement par la conversion de l'image en niveaux de gris. Cette étape simplifie les données en supprimant les informations de couleur, ce qui permet de se concentrer uniquement sur l'intensité lumineuse. Cela réduit la complexité de l'image et facilite les étapes suivantes, tout en conservant les détails essentiels pour la détection des objets [5] (Voir figure 1.13).



Figure 1.13 Conversion RVB vers niveau de gris [12].

- b. Filtrage (réduction de bruit)** Un filtrage est appliqué, souvent à l'aide d'un flou gaussien ou médian, pour réduire le bruit. Cela permet d'éliminer les détails fins non pertinents et d'améliorer la stabilité de la détection des bords. Un traitement sans filtrage peut conduire à des détections erronées à cause de textures ou autres parasites dans l'image [5] (Voir figure 1.14).



Figure 1.14 Application du filtre de GAUSS [12].

- c. Détection des contours (Canny)** La détection des contours, réalisée typiquement avec l'algorithme de Canny qui est reconnu pour sa capacité à détecter les bords avec précision, permet ensuite de repérer les contours nets dans l'image. Cette étape est cruciale pour détecter les formes géométriques qui pourraient correspondre à des plaques d'immatriculation, qui sont en général rectangulaires avec des contours bien définis [21] (Voir figure 1.15).



Figure 1.15 Application de l'algorithme de Canny [13].

- d. Extraction des contours et détection des zones candidates** À partir de l'image des bords, une détection de contours est effectuée pour identifier toutes les formes fermées. Les contours sont ensuite analysés pour extraire les zones candidates, c'est-à-dire les régions dont la taille, la forme, et le rapport largeur/hauteur sont compatibles avec une plaque. Les régions ne correspondant pas aux critères géométriques sont rejetées [13].
- e. Découpage de la plaque** Une fois la plaque est localisée, on effectue un rognage pour extraire uniquement la zone contenant la plaque d'immatriculation (ROI ; region of interest). Cette image est ensuite soumise à un rehaussement du contraste, souvent par égalisation d'histogramme, afin d'augmenter la visibilité des caractères, surtout dans des conditions d'éclairage variables [13].
- f. Extraction des contours des caractères**
- **Binarisation :** Pour faciliter l'extraction des caractères, on applique ensuite une binarisation de l'image. Cette opération, aussi appelée seuillage, elle transforme les pixels en noir ou blanc, ce qui simplifie la séparation des caractères du fond. Elle peut être réalisée par un seuillage global (par exemple, méthode d'Otsu) ou adaptatif (selon l'éclairage local) [22] (Voir figure 1.16).

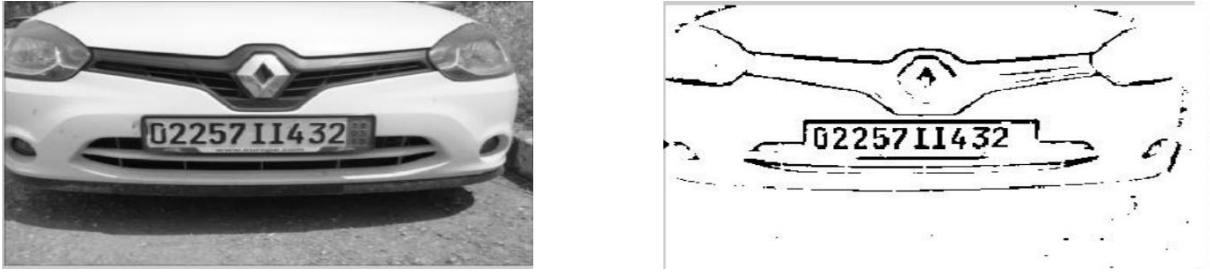


Figure 1.16 Application de la binarisation locale [12]

- **Opérations morphologiques** Des opérations morphologiques telles que la fermeture ou l'ouverture peuvent ensuite être appliquées pour renforcer la structure des caractères ou supprimer de petites imperfections. Ces opérations améliorent la qualité de la segmentation en unifiant les traits discontinus ou en supprimant les petits bruits [23].
- g. **Détection des contours des caractères** L'étape suivante consiste à détecter les contours des caractères à l'intérieur de la plaque binarisée. Chaque caractère est isolé sous forme de petit rectangle, en filtrant selon leur taille et leur position. Cela permet d'extraire individuellement les chiffres et lettres qui composent la plaque.
- k. **Redimensionnement** Les caractères extraits sont ensuite redimensionnés à une taille uniforme (par exemple 28x28 pixels ou plus), ce qui est indispensable pour les étapes de reconnaissance, surtout lorsqu'on utilise un classifieur entraîné ou un OCR.
- l. **Reconnaissance et Classification des chiffres** Enfin, chaque caractère est analysé par une méthode de reconnaissance, qui peut être un système OCR comme Tesseract ou un modèle de classification supervisée (comme un SVM ou un réseau de neurones convolutif). Cette dernière étape permet de convertir les images de caractères en texte lisible, reconstituant ainsi le numéro de la plaque [20] (Voir figure 1.17).

	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		}	~		
€					†	‡		Š	Œ	Ž																š	œ	ž	Ÿ			
	ı	ϕ	£	≈	¥	§	©	«		®	°	±				μ	¶											»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

Figure 1.17 Les caractères reconnus par OCR [13].

Ces méthodes peuvent être efficaces dans des conditions contrôlées mais sont sensibles aux variations d'éclairage et aux occlusions.

1.4.4 Méthodes modernes basées sur l'IA

L'évolution technologique rapide, notamment dans le domaine de l'apprentissage automatique, a permis de franchir un cap décisif dans le traitement des images. Contrairement aux méthodes classiques fondées sur des règles fixes, les approches modernes basées sur l'intelligence artificielle permettent aux systèmes d'apprendre directement à partir des données. Ces techniques s'adaptent mieux aux variations complexes des scènes réelles (éclairage, angles de vue, obstruction partielle, etc.) et améliorent considérablement la précision des résultats. L'avènement de l'intelligence artificielle a révolutionné la détection des plaques :

- **Réseaux de neurones convolutifs (CNN)** capables d'apprendre des caractéristiques complexes à partir de données d'entraînement [18].
- **Modèles YOLO (You Only Look Once)** algorithmes de détection d'objets en temps réel qui identifient les plaques en une seule passe [19].

Ces approches offrent une meilleure robustesse face aux variations d'environnement et permettent une détection plus rapide et précise.

1.4.5 YOLO pour la détection de plaques

YOLO est particulièrement adapté pour la détection de plaques en raison de sa vitesse et de sa précision. Il divise l'image en une grille et prédit simultanément les classes et les boîtes englobantes pour chaque cellule.

Des versions récentes, comme YOLO11, offrent des améliorations en termes de performance et de légèreté, facilitant leur déploiement sur des dispositifs embarqués (Voir figure 1.18).

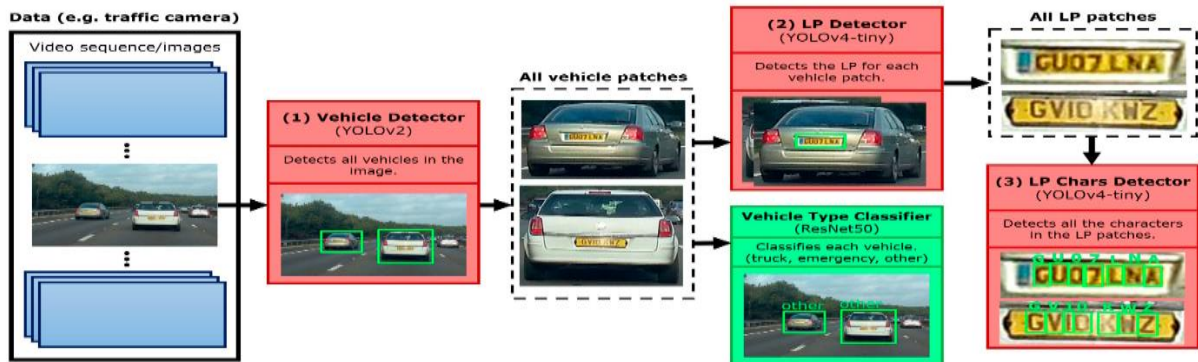


Figure 1.18 Processus de détection et reconnaissance de matricule avec YOLO [10].

1.4.6 Problèmes rencontrés et techniques de traitement

La détection automatique des plaques d'immatriculation présente plusieurs défis, notamment :

- **Variabilité des plaques** différences de taille, de police et de format entre les plaques, même au sein d'un même pays.
- **Conditions d'éclairage** images prises de nuit, en contre-jour, ou avec des reflets peuvent altérer la qualité de la détection.
- **Occlusions et salissures** présence de boue, de poussière ou d'autres objets obstruant partiellement la plaque.
- **Angles de prise de vue** plaques photographiées sous des angles obliques ou à distance variable.

Pour surmonter ces obstacles, plusieurs techniques de traitement d'image sont employées :

- **Prétraitement des images** application de filtres pour réduire le bruit, améliorer le contraste, et corriger l'éclairage.
- **Redressement des perspectives** utilisation de transformations géométriques pour corriger les distorsions dues aux angles de prise de vue.
- **Utilisation de modèles robustes** entraînement de modèles de détection sur des jeux de données variés pour améliorer la généralisation.

1.4.7 YOLOv11 pour la détection des plaques d'immatriculation

Le modèle YOLOv11 (You Only Look Once version 11) représente une avancée significative dans le domaine de la détection d'objets en temps réel. Développé par Ultralytics, YOLO11 introduit plusieurs améliorations architecturales par rapport à ses prédécesseurs, C3k2 améliorant l'extraction des caractéristiques, SPPF permettant une meilleure capture des informations contextuelles à différentes échelles, C2PSA améliorant la détection des petits objets, tels que les caractères des plaques, en se concentrant sur les régions pertinentes de l'image. , YOLO11 offre une combinaison optimale de vitesse et de précision [16], le rendant particulièrement adapté à la détection des plaques d'immatriculation [8].

Bien que YOLO11 ne soit pas pré-entraîné spécifiquement pour la détection des plaques d'immatriculation, il peut être facilement adapté à cette tâche grâce à l'entraînement sur des jeux de données appropriés.

1.5 Conclusion

Ce chapitre a présenté une vue d'ensemble des systèmes d'immatriculation à travers le monde, avec un accent particulier sur le système algérien, en soulignant les spécificités et les défis associés. Il a également exploré les techniques de traitement d'image utilisées pour la détection des plaques, en mettant en lumière les avancées offertes par les modèles modernes tels que YOLO11. Ces éléments posent les bases pour le développement d'un système de reconnaissance automatique des plaques d'immatriculation, qui sera détaillé dans les chapitres suivants.

Chapitre 2

Intelligence artificielle et détection de matricules

2.1 Introduction

La reconnaissance automatique des plaques d'immatriculation (Automatic License Plate Recognition - ALPR) combine des techniques avancées de traitement d'image et d'intelligence artificielle, en particulier celles issues de l'apprentissage profond. L'apparition des réseaux de neurones convolutifs (CNN) a permis des progrès significatifs dans la détection et la reconnaissance visuelle, améliorant la précision et la rapidité des systèmes ALPR. Parmi les modèles de détection d'objets, la famille YOLO (You Only Look Once) s'est imposée comme une solution efficace pour la détection en temps réel. Ce chapitre présente les fondements des CNN, les principes de la détection d'objet, et se focalise sur le modèle YOLO, avec une attention particulière portée à sa version avancée YOLOv11 qui constitue l'outil central utilisé dans ce projet.

2.2 Apprentissage profond et réseaux de neurones convolutifs

L'apprentissage profond est une sous-catégorie de l'intelligence artificielle qui utilise des réseaux de neurones artificiels profonds pour extraire automatiquement des caractéristiques complexes à partir des données brutes, notamment les images. Depuis la percée majeure d'AlexNet en 2012, qui a obtenu un succès significatif dans la classification d'images, les réseaux convolutifs (CNN) sont devenus la norme pour les tâches de vision par ordinateur. Les CNN sont particulièrement efficaces pour la détection des objets dans des images, car ils exploitent des filtres convolutifs capables de capturer des motifs locaux invariants (bords, textures, formes). Cette propriété est cruciale pour la détection fiable des plaques d'immatriculation, notamment la localisation de la plaque, la segmentation des caractères et leur reconnaissance même dans des conditions variées (angle, éclairage, bruit) (voir figure 2.1). [24]

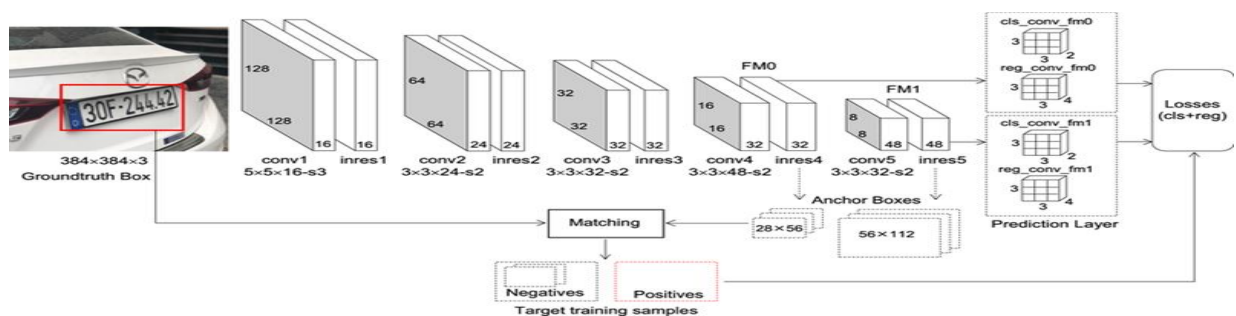


Figure 2.1 Architecture de réseau de détection d'objets à base de CNN, appliquée à la détection de plaques d'immatriculation [18].

2.2.1 Architecture de base du CNN

Les réseaux de neurones convolutifs s'appuient sur une architecture hiérarchique de couches spécialisées dans le traitement d'images. L'image en entrée passe d'abord par des couches convolutives, qui appliquent des filtres pour détecter des motifs locaux comme les bords et textures. Une fonction d'activation non linéaire, typiquement la ReLU, est ensuite appliquée pour permettre l'apprentissage de relations complexes entre les données. Les couches de pooling, souvent du max pooling, suivent afin de réduire la taille des données tout en conservant les informations essentielles. Cette réduction permet non seulement de limiter le surapprentissage, mais aussi d'améliorer l'efficacité computationnelle. Ces blocs convolution/activation/pooling peuvent être répétés plusieurs fois, ce qui permet au réseau d'apprendre des représentations de plus en plus abstraites et riches. Les données sont ensuite aplaties (flatten) puis envoyées vers des couches entièrement connectées, responsables de la classification ou de la localisation. Cette hiérarchie permet aux CNN d'apprendre progressivement les caractéristiques des images, depuis les formes simples jusqu'aux objets complexes, et s'adapte particulièrement bien à des tâches comme la reconnaissance automatique de plaques d'immatriculation (voir figure 2.2) [26].

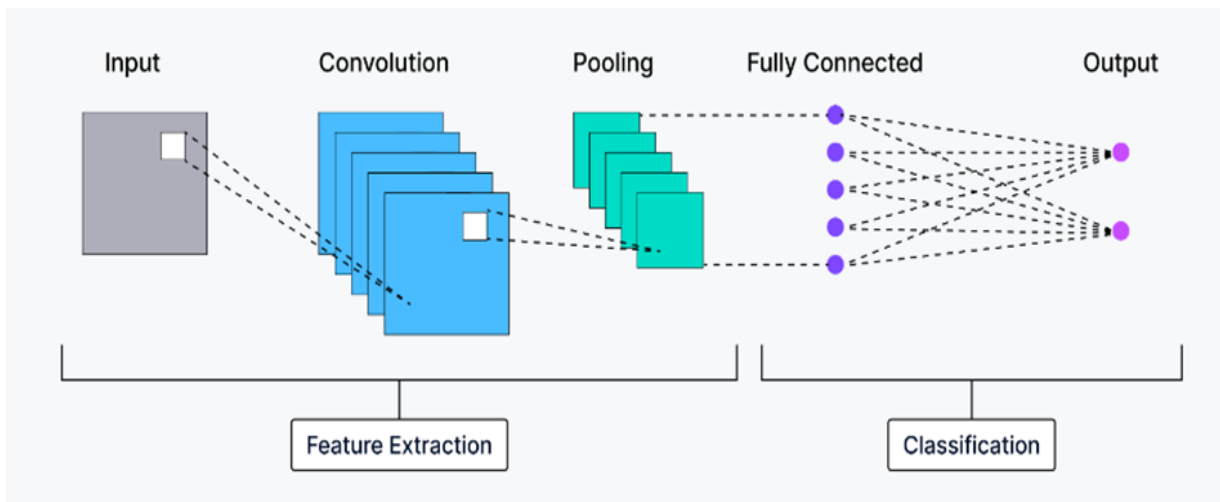


Figure 2.2 Architecture des réseaux neuronaux convolutifs [25].

a. Couche de Convolution (Convolutional Layer)

La couche de convolution est la base des CNN. Elle applique des filtres glissants (ou noyaux) sur l'image d'entrée pour extraire automatiquement des caractéristiques locales telles que les bords ou textures. Chaque filtre parcourt l'image et calcule un produit scalaire entre ses poids et les pixels, produisant une carte de caractéristiques (feature map). Cette opération permet de détecter des motifs importants tout en réduisant le nombre de paramètres comparé à un réseau dense [26] (voir figure 2.3).

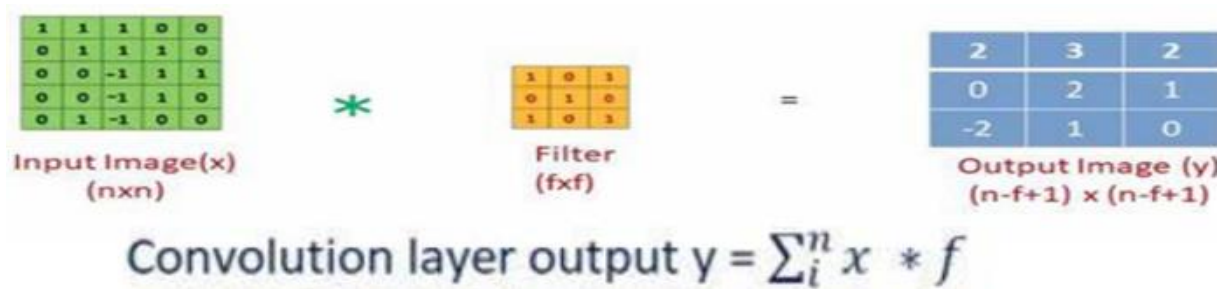


Figure 2.3 Couche de Convolution [27].

b. Fonction d'activation

- (ReLU et Leaky ReLU)

La fonction ReLU (Rectified Linear Unit) introduit de la non-linéarité dans le réseau, ce qui permet au modèle d'apprendre des relations complexes. Elle transforme chaque valeur en sortie de convolution selon la formule (voir 2.1). Autrement dit, elle remplace les valeurs négatives par zéro, tout en laissant passer les positives, ce qui accélère et stabilise l'apprentissage (voir la figure 2.4 et 2.5) [28].

$$\text{ReLU}(x) = \max(0, x) \quad (2.1)$$



Figure 2.4 Exemple de la formule $\text{ReLU}(x) = \max(0, x)$ [27].

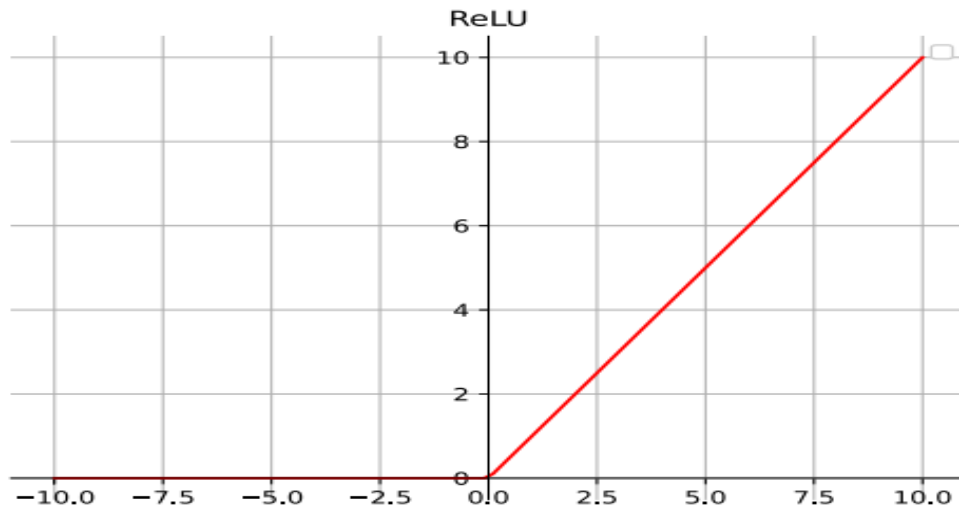


Figure 2.5 Graphe de la fonction ReLU [26].

Cependant, une limitation de ReLU est qu'elle peut conduire à des "neurones morts" : si un neurone reçoit des valeurs négatives en entrée pendant tout l'entraînement, son gradient devient nul et il cesse d'apprendre. Pour atténuer ce problème, on peut utiliser la fonction Leaky ReLU (voir 2.2), une variante qui permet une petite pente pour les valeurs négatives :

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{si } x \geq 0 \\ \alpha x & \text{si } x < 0 \end{cases} \quad (2.2)$$

Avec généralement $\alpha=0,01$. Cette fonction conserve un léger flux de gradient même pour les entrées négatives, ce qui réduit le risque de blocage du neurone [44].

- **Fonction d'activation Tanh**

La fonction Tanh (tangente hyperbolique) est une fonction d'activation non linéaire qui transforme les valeurs d'entrée en un intervalle compris entre -1 et 1 selon la formule (voir 2.3).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

Centrée en zéro, elle facilite l'apprentissage en réduisant les biais de distribution des données. Toutefois, elle reste sensible au problème du vanishing gradient, ce qui limite son usage dans

les réseaux profonds. Tanh est encore utilisée dans certains cas, notamment dans les réseaux récurrents ou pour des données normalisées [44].

- **Fonction d'activation de sortie (Softmax / Sigmoid)**

La phase finale d'un réseau de neurones convolutif (CNN) est assurée par une fonction d'activation de sortie, généralement Softmax ou Sigmoid, en fonction du type de tâche de classification. La fonction de sortie transforme les valeurs produites par la dernière couche du réseau (souvent une couche dense) en probabilités interprétables.

- **Softmax** est utilisée pour la classification multi classe : elle convertit un vecteur de scores en probabilités normalisées dont la somme est égale à 1. Elle met en évidence la classe la plus probable (voir figure 2.6).

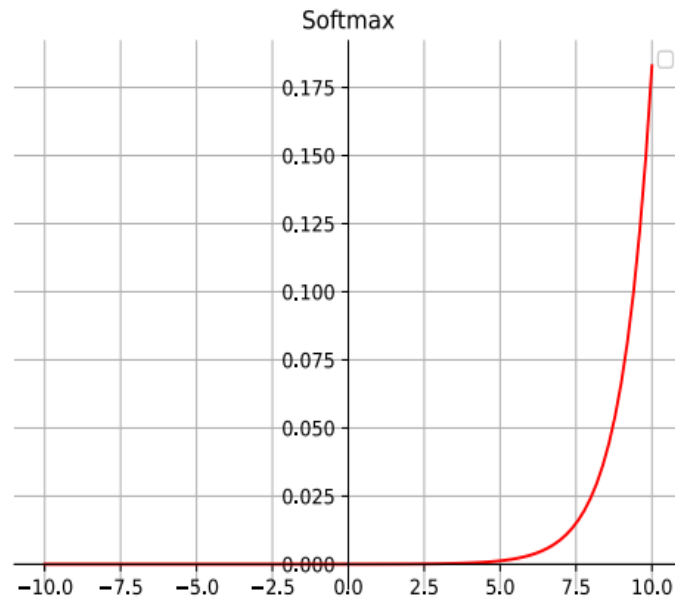


Figure 2.6 Graphe de la fonction d'activation Softmax [26].

- **Sigmoid** cette onction est utilisée pour la classification binaire ou multi-étiquette ,elle applique la fonction logistique à chaque sortie, produisant une probabilité indépendante entre 0 et 1 pour chaque classe (voir figure 2.7).

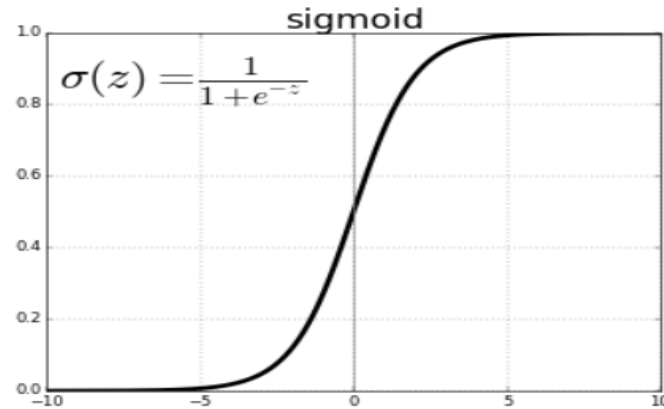


Figure 2.7 Graphe de la fonction d'activation Sigmoid [33].

Ces fonctions permettent à un modèle d'interpréter ses sorties comme des prédictions de classes [26].

c. Couche de Pooling (Pooling Layer)

La couche de pooling sert à réduire la taille des cartes de caractéristiques tout en conservant les informations importantes. La méthode la plus courante est le max pooling, qui sélectionne la valeur maximale dans chaque sous-région. Cela permet de réduire la complexité computationnelle, de limiter le surapprentissage et de rendre le modèle invariant aux petites translations dans l'image (voir figure 2.8) [29].

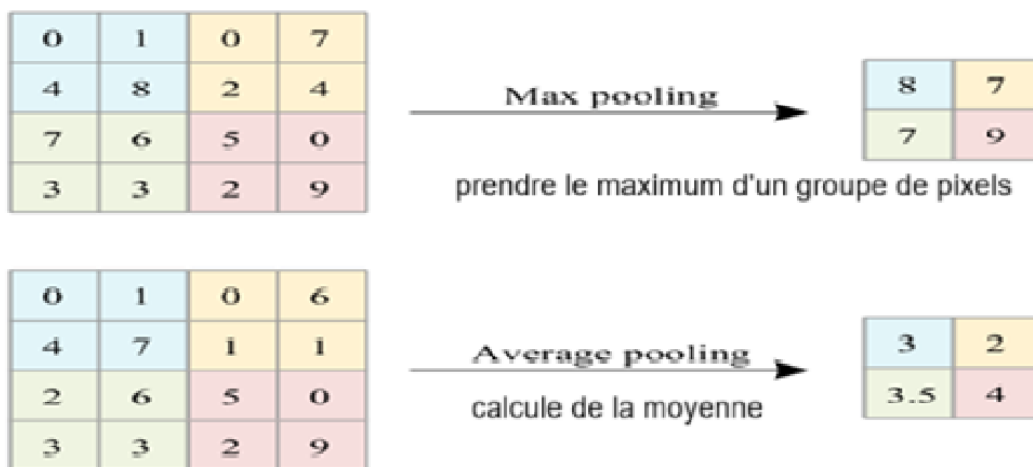


Figure 2.8 Couches de pooling [26].

d. Couches d'aplatissement (Flattening)

L'étape d'aplatissement consiste à transformer les cartes de caractéristiques multidimensionnelles (générées par les couches convolutives et de pooling) en un vecteur unidimensionnel. Ce vecteur peut ensuite être traité par les couches entièrement connectées (Fully Connected Layers) pour effectuer la prédiction finale. L'aplatissement ne modifie pas les données mais change leur forme pour les rendre compatibles avec les couches denses (voir figure 2.9) [30].

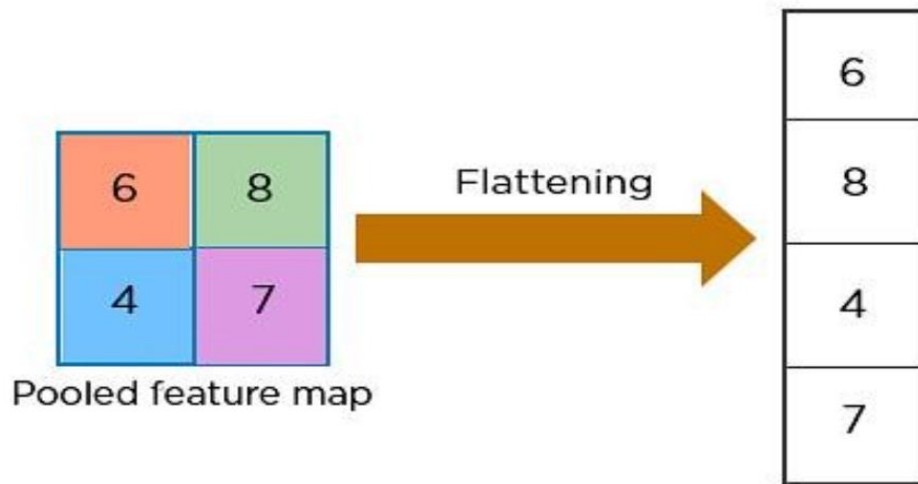


Figure 2.9 Couches d'aplatissement (flattening) [31].

e. Couche entièrement connectée (Fully Connected Layer, Dense Layer)

La couche entièrement connectée (ou dense) relie tous les neurones de la couche précédente à chaque neurone de cette couche. Elle traite les caractéristiques extraites lors des étapes précédentes pour produire une sortie finale comme une probabilité de classe (via Softmax ou Sigmoid). C'est dans ces couches que le réseau apprend à combiner les informations locales en une décision globale (voir figure 2.10) [26].

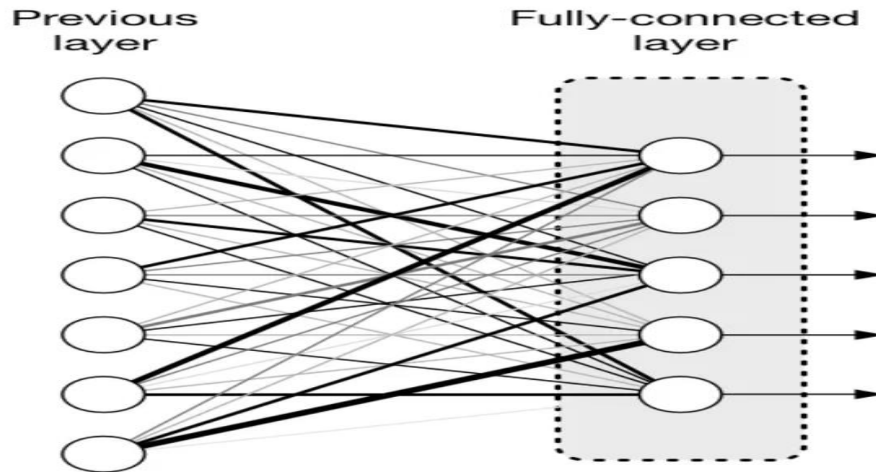


Figure 2.10 Couche entièrement connectée [32].

2.3 Algorithme YOLO

L'évolution rapide des méthodes de vision par ordinateur a permis de développer des approches de détection d'objets de plus en plus performantes. Parmi elles, YOLO (You Only Look Once) s'est imposée comme l'un des algorithmes les plus efficaces pour la détection en temps réel. Proposé initialement par Redmon et al. [19] en 2016, YOLO effectue la détection d'objets en une seule passe à travers le réseau de neurones. Contrairement aux méthodes traditionnelles telles que R-CNN et ses variantes, qui décomposent le processus en plusieurs étapes (proposition de régions, classification, ajustement de boîte), YOLO traite l'image entière d'un seul coup, ce qui améliore considérablement la vitesse d'inférence (voir figure 2.11).

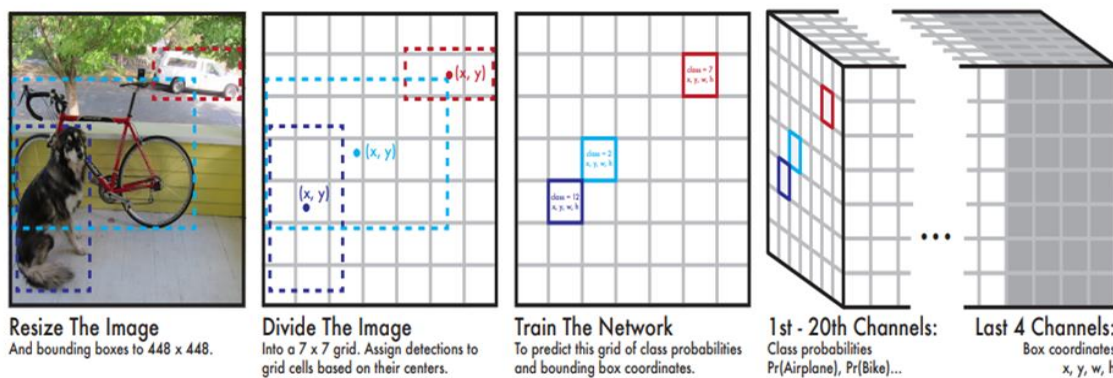


Figure 2.11 Modèle de YOLO [19].

2.3.1 Architecture de YOLO

L'architecture de YOLO est basée sur un réseau neuronal convolutif unique qui divise l'image d'entrée en une grille $S \times S$ et, pour chaque cellule de cette grille, prédit un certain nombre de boîtes englobantes avec des scores de confiance et des probabilités de classes. Cette conception permet à YOLO de traiter des images à une vitesse impressionnante adaptée aux applications temps réel (voir figure 2.12) [19].

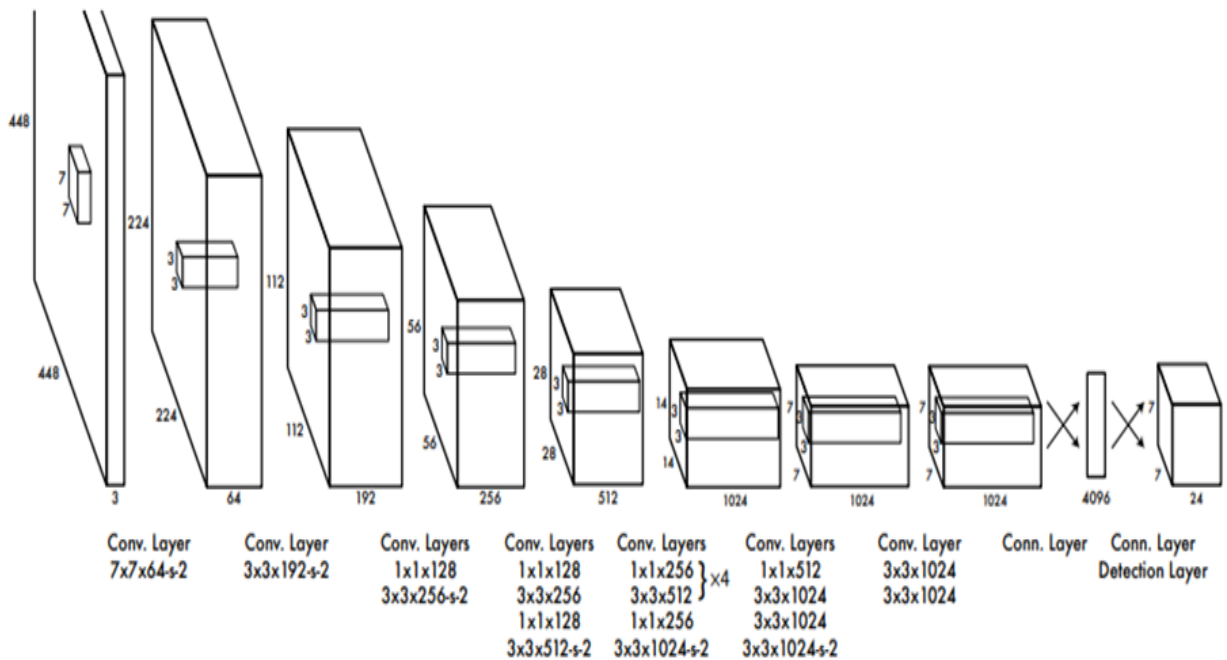


Figure 2.12 Architecture de YOLO [19].

2.3.2 Processus de détection d'objets avec YOLO

L'algorithme YOLO (You Only Look Once) est une méthode de détection d'objets en temps réel qui traite l'image entière en une seule passe à travers un réseau de neurones convolutifs (CNN). Il divise l'image en une grille de taille $S \times S$, où chaque cellule est responsable de la détection des objets dont le centre tombe à l'intérieur de cette cellule. Chaque cellule prédit plusieurs boîtes de délimitation (bounding boxes) et leurs scores de confiance associés, qui reflètent à la fois la probabilité de présence d'un objet et la précision de la boîte via le calcul de l'Intersection over Union (IoU) entre la boîte prédite et la boîte réelle. YOLO utilise des boîtes d'ancrage (anchor boxes), qui sont des boîtes de délimitation prédéfinies avec différentes échelles et rapports d'aspect. Ces boîtes permettent au modèle de mieux s'adapter à la variabilité des objets dans les images. L'utilisation de l'IoU permet de mesurer la qualité de chaque prédiction en comparant les boîtes de sortie aux véritables annotations (voir figure 2.13) [36].

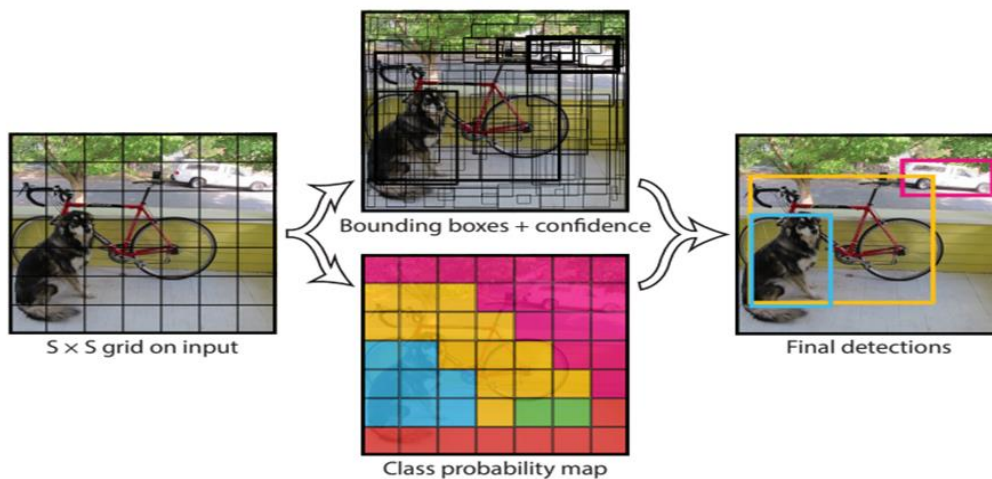


Figure 2.13 Processus de détection d'objets avec YOLO [19].

a. Boîtes englobantes (Bounding Box)

Une boîte englobante est un rectangle qui encadre un objet d'intérêt à l'intérieur d'une image ou d'une cellule de la grille. Elle permet de localiser spatialement l'objet détecté en indiquant sa position, sa taille et sa classe (ex. : voiture, chien, fumée...) (voir figure 2.14) [36].

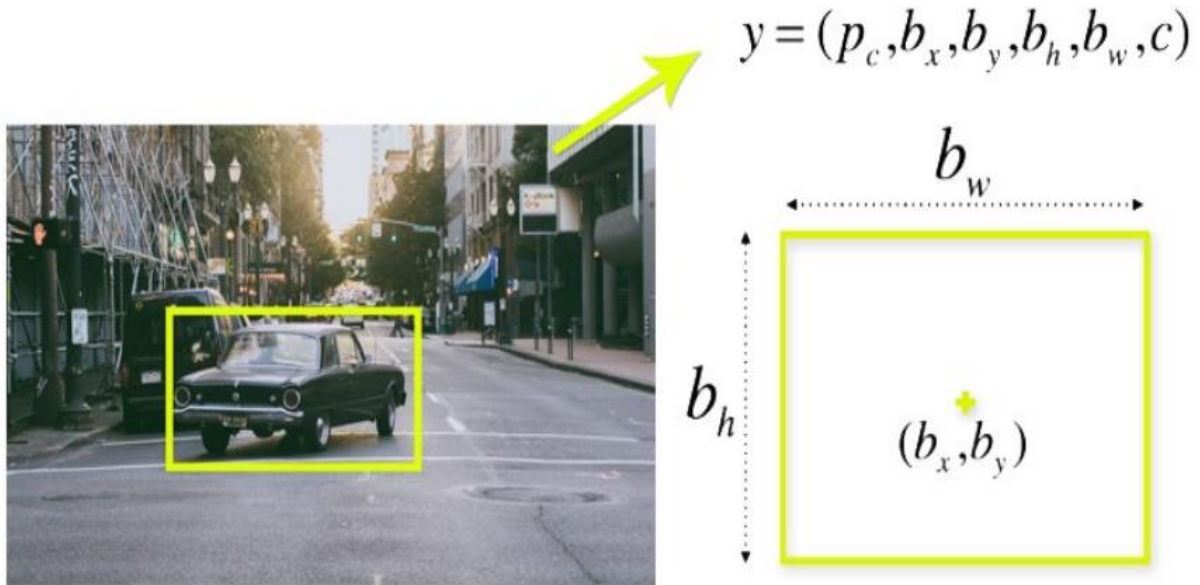


Figure 2.14 Exemple d'une boîte englobante [36].

L'algorithme YOLO utilise une régression unique de boîte englobante (bounding box regression) pour prédire tous ces éléments à la fois. Cela signifie que, plutôt que d'isoler la tâche de classification de celle de localisation comme dans d'autres approches traditionnelles, YOLO les traite de manière conjointe et directe, en une seule passe du réseau (voir figure 2.15) [36].

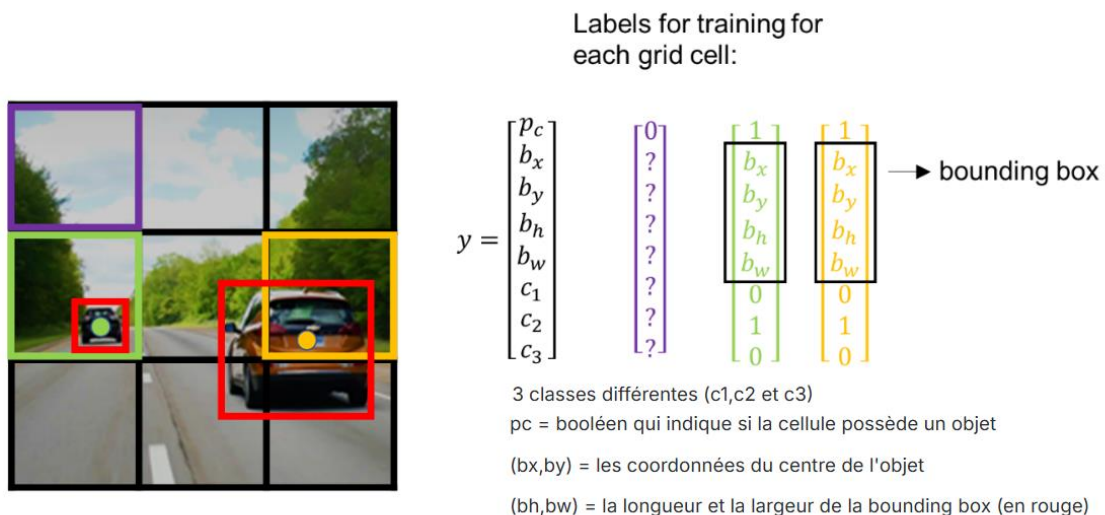


Figure 2.15 Modèle YOLO simplifié avec une grille de 3 x 3, trois classes, produit un vecteur de huit valeurs [37].

b. Boîte d’ancrage

Les boîtes d’ancrage (ou anchor boxes) sont des boîtes prédéfinies utilisées pour améliorer la détection d’objets de différentes formes, tailles et rapports d’aspect dans une même cellule de la grille. Ces boîtes permettent de prédire plusieurs objets dans une même cellule, en associant chaque objet à l’ancre la plus appropriée. Le modèle apprend à ajuster chaque boîte d’ancrage à l’objet réel à travers une régression des décalages relatifs (position, largeur, hauteur) et à prédire un score de confiance et une classe. Cette approche permet à YOLO d’être plus flexible et précis, notamment dans les scènes contenant plusieurs objets proches ou de tailles variées (voir figure 2.16) [38].



Figure 2.16 Deux objets détectés dans la même grille [39].

c. Intersection over Union

L’Intersection over Union (IoU) est une métrique essentielle en détection d’objets, utilisée pour évaluer la qualité des prédictions faites par un modèle tel que YOLO. Elle mesure le recouvrement entre une boîte prédite par le modèle et la boîte réelle (ou ground truth) annotée dans les données. L’IoU est définie comme le rapport entre l’aire de l’intersection des deux boîtes et l’aire de leur union (voir figure 2.17) [38].

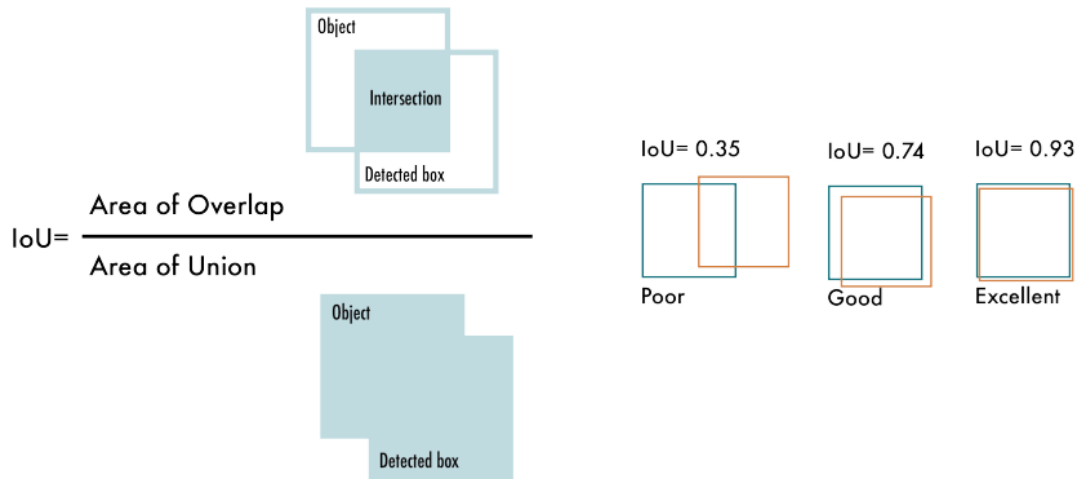


Figure 2.17 Calcul de l'Intersection over Union [38].

L'IoU est également utilisée pour l'application de Non-Maximum Suppression (NMS), une technique qui permet d'éliminer les prédictions redondantes (multiples boîtes pour un même objet) en ne gardant que celle avec le meilleur score de confiance et une IoU suffisamment faible avec les autres (voir figure 2.18) [38].

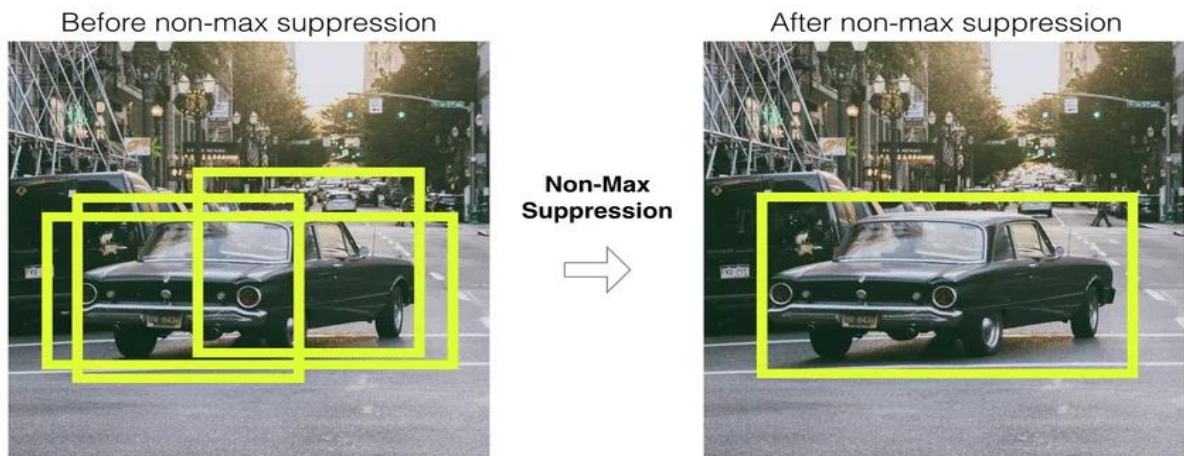


Figure 2.18 Application du NMS [40].

2.3.3 Évolution des modèles YOLO

YOLO a connu plusieurs évolutions successives (YOLOv1 à YOLOv12), avec à chaque version des améliorations sur la vitesse d'inférence, précision de détection, la gestion des objets

de petite taille, et la robustesse générale du modèle. YOLOv11 représente aujourd'hui l'une des versions les plus avancées, intégrant des techniques issues des dernières recherches d'apprentissage profond et de vision par ordinateur, ce qui en fait un choix de premier plan pour les applications industrielles et embarquées, telles que la surveillance, la conduite autonome ou la reconnaissance de plaques d'immatriculation (voir figure 2.19) [34].

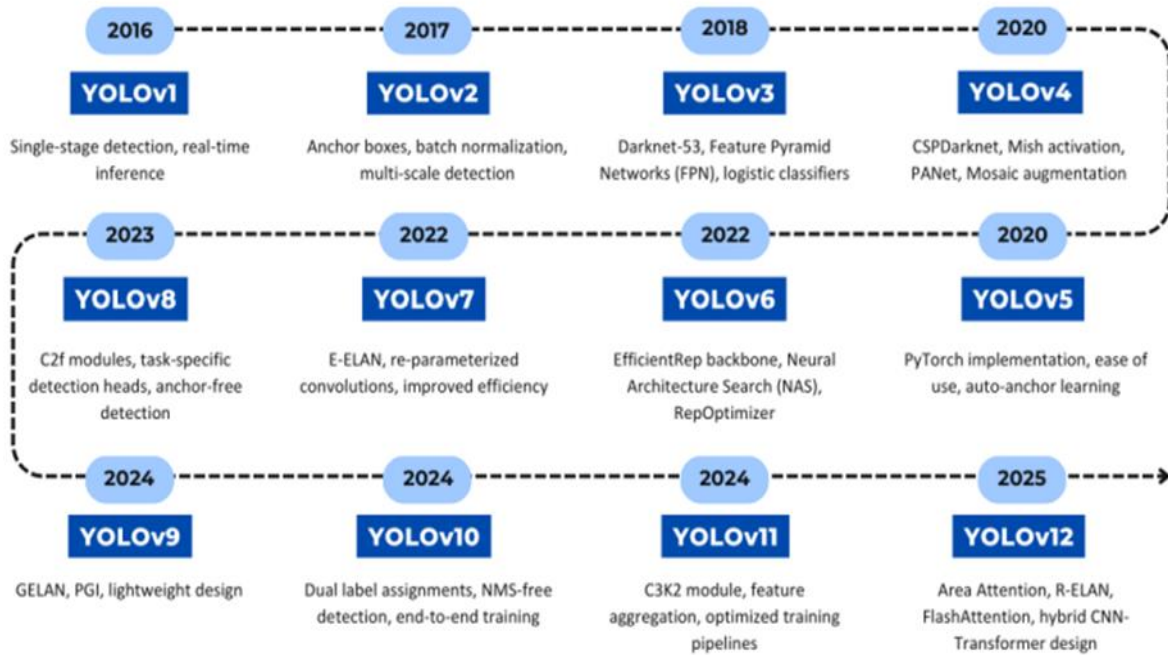


Figure 2.19 Evolution des modèles de YOLO à travers les années [35].

2.4 Algorithme YOLO11

YOLOv11 (You Only Look Once version 11) est une version récente et avancée de la famille des modèles YOLO, conçue pour la détection d'objets en temps réel avec un haut niveau de précision et une efficacité computationnelle accrue. Il s'inscrit dans la continuité des évolutions apportées depuis YOLOv1, en intégrant les meilleures pratiques issues des dernières recherches en vision par ordinateur et en apprentissage profond. YOLO11 étend également ses capacités à des tâches variées de vision par ordinateur, notamment la détection d'objets, la segmentation d'instances, l'estimation de pose et la détection d'objets orientés (OBB). Le modèle offre une gamme de tailles, du nano à l'extra-large, permettant une adaptation flexible aux contraintes spécifiques des dispositifs embarqués ou des environnements de calcul intensif [16]. YOLO11 est particulièrement pertinent pour les tâches de reconnaissance automatique de

plaques d'immatriculation (ALPR), où la détection rapide et fiable de petits objets (comme les caractères) est essentielle. Il offre également une meilleure adaptabilité aux environnements complexes, grâce à sa capacité à gérer des variations d'échelle, d'orientation et de luminosité [8].

2.4.1 Architecture de YOLO11

Le modèle YOLOv11 représente une évolution significative dans la série des modèles YOLO, combinant vitesse de traitement et précision grâce à une architecture profondément optimisée. Inspiré de ses prédécesseurs, YOLO11 adopte une structure modulaire composée de trois éléments principaux : le Backbone, le Neck et le Head. Chaque module joue un rôle spécifique dans le traitement des images d'entrée et dans l'extraction de prédictions précises (voir figure 2.20) [16].

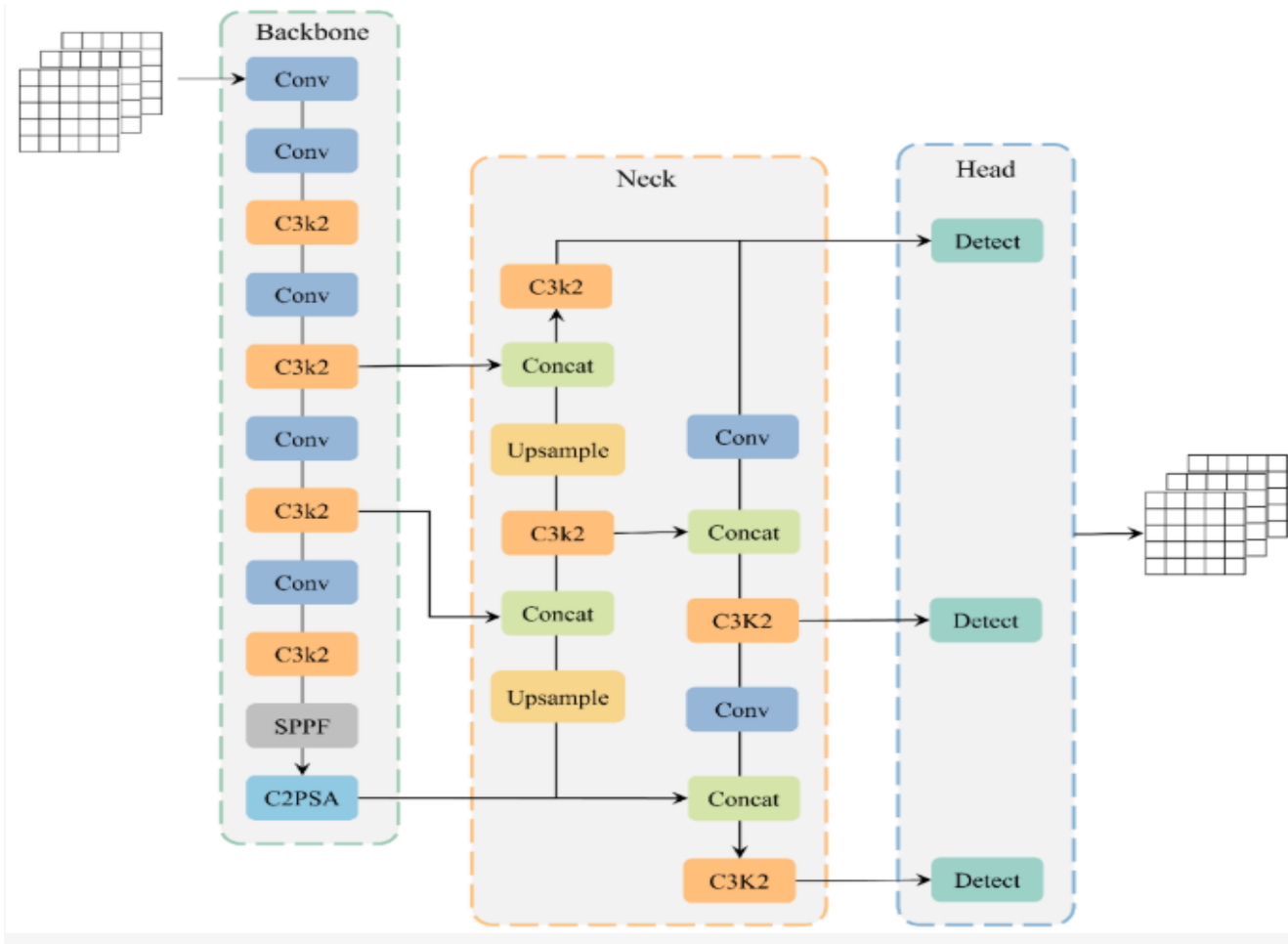


Figure 2.20 Architecture originale du YOLO11 [42].

a. Backbone

Le Backbone est la première partie du modèle YOLO11. Il a pour rôle de transformer l'image d'entrée en un ensemble de représentations profondes appelées cartes de caractéristiques, utiles pour la détection. Le backbone est composé de :

- Plusieurs blocs Conv (couches de convolution) pour extraire les motifs visuels.
- Des modules C3k2, qui sont des blocs convolutifs profonds, afin d'améliorer la propagation du gradient et réduire la redondance.
- Un bloc SPPF (Spatial Pyramid Pooling - Fast), qui permet d'extraire des informations contextuelles à plusieurs échelles tout en conservant une faible complexité.
- Une couche C2PSA, qui représente une attention spatiale améliorée, renforçant les zones pertinentes de l'image pour la détection.

Ce backbone permet donc d'extraire efficacement les caractéristiques visuelles tout en préservant la rapidité d'exécution, un point essentiel pour les applications en temps réel comme la reconnaissance de plaques d'immatriculation [16].

b. Neck

Le Neck assure la fusion et la réorganisation des cartes de caractéristiques extraites par le Backbone, afin de mieux détecter des objets de différentes tailles.

- Le Neck repose sur des blocs C3k2, des Concat (concaténations) et des opérations Upsample (suréchantillonnage).
- Il combine des informations provenant de plusieurs niveaux de profondeur, ce qui est crucial pour localiser des objets de petite taille comme les chiffres des plaques.
- Cette structure multi-échelle, inspirée des FPN (Feature Pyramid Networks) et PAFPN, améliore la richesse des représentations transmises au Head.

Le Neck agit donc comme un pont entre l'extraction de caractéristiques et la prédiction finale [16].

c. Head

Le Head est la dernière partie du modèle. Il est chargé de produire les prédictions finales, à savoir :

- Les boîtes englobantes (bounding boxes).
- Les scores de confiance.

- Les classes d'objet détectées (par exemple : plaque avant, plaque arrière, chiffres, etc.).

YOLO11 utilise plusieurs branches de détection en parallèle, ce qui permet de faire des prédictions à différentes résolutions. Cela améliore la précision, notamment pour des objets de tailles variables dans une même image (voir figure 2.21) [16].

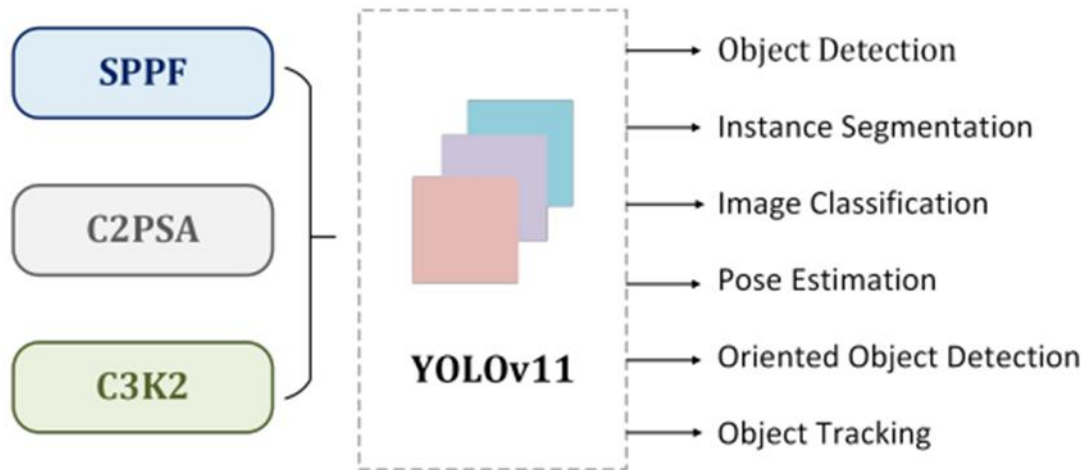


Figure 2.21 Architecture du modèle yolo11 [16].

2.4.2 Déclinaisons Spécialisées de YOLO11

L'architecture modulaire de YOLO11 permet une adaptation fluide à divers types de tâches de vision par ordinateur. En plus de la version de base dédiée à la détection d'objets, plusieurs déclinaisons ont été développées pour répondre à des cas d'usage spécifiques : segmentation d'instances, estimation de pose, classification d'image, et détection orientée (bounding boxes rotatives). Ces variantes conservent les blocs fondamentaux du modèle principal (C3k2, SPPF, C2PSA), tout en incorporant des couches supplémentaires adaptées à chaque tâche (voir figure 2.22) [41].

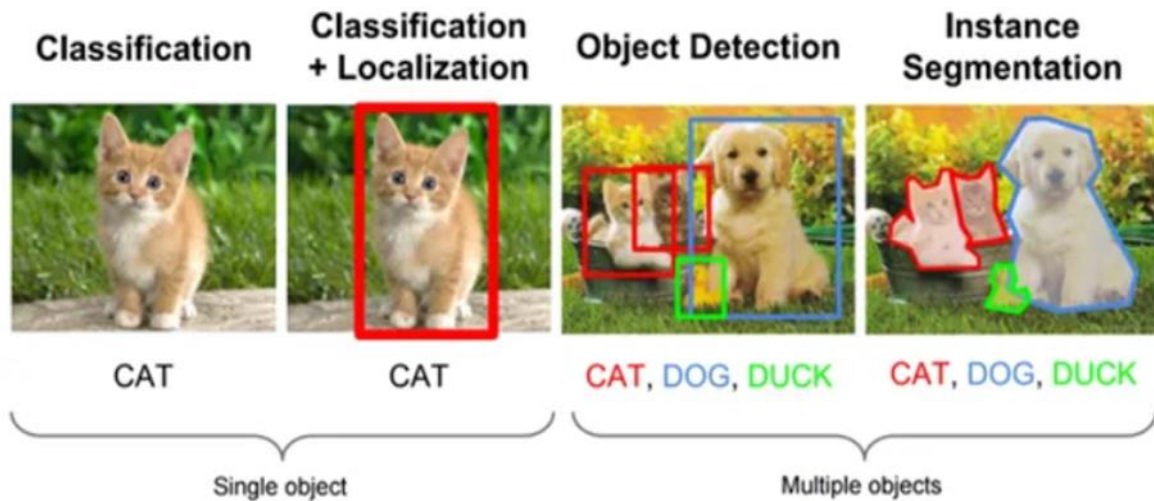


Figure 2.22 YOLO11 offre plusieurs tâches de vision par ordinateur [27].

2.4.3 Déclinaisons dimensionnelles du YOLO11 : Versions n, s, m, l, x

Afin d'adapter le modèle YOLOv11 à une grande diversité de cas d'usage, Ultralytics propose cinq versions graduelles , YOLO11n (Nano), s (Small), m (Medium), l (Large) et x (Extra-Large). Ces déclinaisons partagent la même architecture fondamentale (incluant les blocs C3k2, C2f et SPPF), mais varient en complexité selon la profondeur des couches, le nombre de canaux, et les paramètres d'apprentissage. Cette modularité permet d'équilibrer la précision, la vitesse d'inférence et la consommation de ressources, selon les exigences du contexte d'utilisation, allant de l'embarqué léger au traitement haute performance [41] (voir figure 2.23).

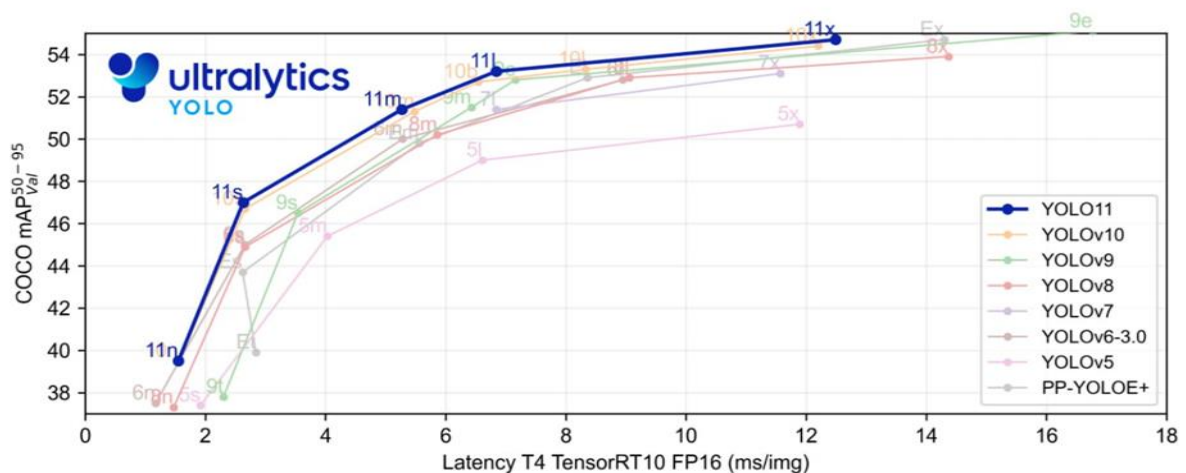


Figure 2.23 Graphique de comparaison des performances de YOLO11 avec différents précédents modèles [8].

2.4.4 Technique d'évaluation du model YOLO

L'évaluation d'un modèle de détection d'objets tel que YOLO11 repose sur des indicateurs de performance spécifiques, adaptés à la fois à la classification des objets et à leur localisation spatiale dans l'image. Ces métriques permettent de quantifier la capacité du modèle à identifier correctement les objets d'intérêt (par exemple, les plaques d'immatriculation ou les caractères alphanumériques) tout en évitant les fausses détections.

a. Matrice de confusion

La matrice de confusion est un outil fondamental en apprentissage automatique pour visualiser les performances d'un modèle. Elle permet d'évaluer les performances en détaillant les prédictions effectuées par rapport aux vraies valeurs, les erreurs et les faiblesses du modèle. Elle est particulièrement utile pour les modèles de classification (binaire ou multi classe) afin de mesurer la précision, le rappel, la spécificité et d'autres métriques de performance.

La structure de la matrice de confusion est représentée visuellement sous la forme d'un tableau réparti en quatre groupes (voir figure 2.24) [43].

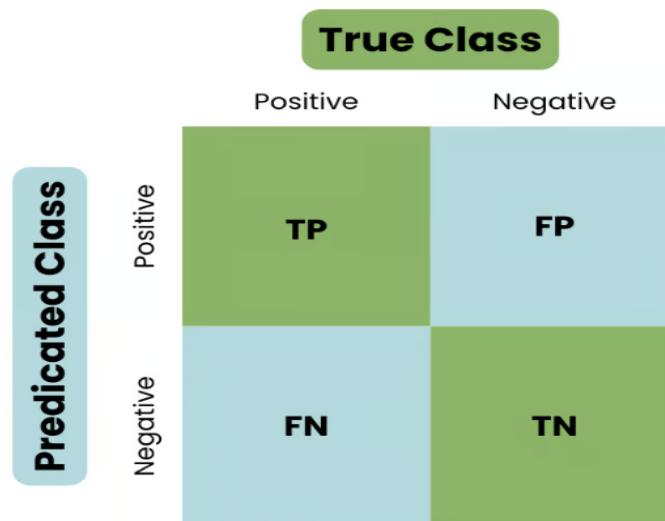


Figure 2.24 Structure de la matrice de confusion [14]

- **Vrai positif [True Positive TP]** le modèle détecte correctement un objet.
- **Faux positif [False Positive FP]** un objet ne figurant pas sur l'image est détecté.
- **Faux négatif [False negative FN]** un objet n'est pas détecté dans la réalité du terrain.
- **Vrai négatif [True Negative TN]** le modèle prédit correctement la classe négative.

b. Mesures de performance

À partir de la matrice de confusion, plusieurs métriques de performance peuvent être dérivées pour évaluer la qualité du modèle. Voici les plus courantes :

- **Précision**

La proportion des prédictions positives correctes par rapport à toutes les prédictions positives faites (voir 2.4).

$$\text{Precision} = \frac{VP}{VP+FP} \quad (2.4)$$

- **Recall (Rappel ou Sensibilité)**

Mesure la capacité du modèle à détecter les cas positifs (voir 2.5).

$$\text{Recall} = \frac{VP}{VP+FN} \quad (2.5)$$

- **Accuracy (Précision globale)**

Mesure la proportion des bonnes prédictions sur l'ensemble des données (voir 2.6).

$$\text{Accuracy} = \frac{VP+VN}{VP+VN+FP+FN} \quad (2.6)$$

- **F1-Score (F-mesure)**

Moyenne harmonique entre précision et rappel. Utile en cas de classes déséquilibrées (voir 2.7).

$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.7)$$

2.5 Choix du YOLO11 pour la détection des plaques d'immatriculation

Le modèle YOLO11 présente plusieurs avantages majeurs pour la détection automatique des plaques d'immatriculation (ANPR – Automatic Number Plate Recognition). Tout d'abord, sa rapidité d'inférence exceptionnelle permet une détection en temps réel, ce qui est crucial pour les systèmes embarqués ou les applications de vidéosurveillance. Grâce à son architecture optimisée, intégrant des blocs légers et efficaces comme C2f, C3k2, et C2PSA, YOLO11 offre une très bonne précision même dans des environnements complexes, tels que les routes

encombrées ou les conditions d'éclairage variables. Il excelle également dans la détection des objets de petite taille, comme les caractères des plaques, souvent difficiles à distinguer avec d'autres architectures. De plus, sa flexibilité, avec des variantes de taille allant de nano à xlarge, permet son déploiement aussi bien sur des machines puissantes que sur des dispositifs à ressources limitées. En combinant rapidité, robustesse, et précision, YOLO11 constitue ainsi un choix particulièrement adapté pour les systèmes ANPR modernes (voir figure 2.25) [8].



Figure 2.25 YOLOv11 pour la détection des plaques d'immatriculation [17].

2.6 Conclusion

Ce chapitre a permis de poser les bases théoriques nécessaires à la compréhension des techniques d'intelligence artificielle appliquées à la détection automatique des plaques d'immatriculation. Nous avons d'abord présenté les principes fondamentaux de l'apprentissage profond et des réseaux de neurones convolutifs (CNN), avant d'explorer en détail le modèle YOLO et son évolution jusqu'à la version YOLO11. Grâce à ses performances en temps réel, sa capacité à détecter des objets de petite taille avec une grande précision, et son architecture optimisée, YOLO11 se révèle particulièrement adapté à la détection des plaques d'immatriculation dans des environnements variés. Forts de ces fondements, nous aborderons dans le prochain chapitre la mise en œuvre pratique de notre système ANPR (Automatic Number Plate Recognition).

Chapitre 3

Implémentation et résultats

3.1 Introduction

Ce chapitre présente le travail personnel réalisé pour le développement d'un système complet de reconnaissance automatique des plaques d'immatriculation, basé sur les réseaux de neurones convolutifs et les techniques modernes de vision par ordinateur. L'objectif de ce projet est d'explorer différentes approches de détection et de reconnaissance des plaques, en s'appuyant principalement sur le modèle YOLO11, tout en comparant les performances obtenues. Le système a été implémenté en trois étapes successives, chacune correspondant à une architecture et une stratégie de traitement distinctes :

- **Première approche détection des plaques uniquement avec YOLO11**

Dans cette première phase, le modèle YOLOv11 est entraîné pour détecter uniquement les plaques d'immatriculation (avant et arrière) sur des images ou des vidéos. Cette étape permet de localiser avec précision les zones contenant les plaques, mais sans effectuer la reconnaissance des caractères.

- **Deuxième approche détection et reconnaissance des caractères avec YOLO11**

Ici, YOLO11 est utilisé à double niveau : d'abord pour détecter les plaques, puis pour détecter et reconnaître les chiffres (de 0 à 9) présents sur les plaques extraites. Cette méthode repose sur l'apprentissage supervisé avec des annotations précises, afin de permettre une reconnaissance directe des caractères à l'aide du même modèle.

- **Troisième approche détection avec YOLO11 et reconnaissance avec un OCR (EasyOCR et Tesseract).**

Dans cette dernière approche, YOLO11 est utilisé uniquement pour détecter les plaques, tandis que la reconnaissance des caractères est assurée par OCR (EasyOCR et Tesseract). Cette combinaison permet d'évaluer la performance d'un pipeline mixte, associant un détecteur rapide et un outil de reconnaissance plus générique.

Après la mise en œuvre de ces trois configurations, une comparaison expérimentale a été menée afin d'évaluer les performances respectives en termes de précision et taux de reconnaissance. Ces évaluations permettent de dégager les avantages et les limites de chaque approche.

Le système a été entièrement développé en Python, en s'appuyant sur les bibliothèques OpenCV, Tkinter, et les plateformes Roboflow et Google Colab pour l'entraînement des modèles. Une interface graphique a également été conçue pour faciliter les tests et l'exploitation du système. Ce chapitre détaille l'implémentation de chaque approche, les choix techniques adoptés, les résultats obtenus, ainsi que l'analyse comparative des performances.

3.2 Environnement de travail

Le choix d'un environnement de développement adapté constitue une condition indispensable à la réussite de tout projet en intelligence artificielle. Celui-ci doit offrir un compromis optimal entre puissance de calcul, souplesse logicielle et efficacité de développement.

Les différentes phases du projet de l'écriture du code aux tests d'interface, en passant par l'entraînement des modèles ont été réparties entre un poste de travail local et des ressources de calcul en nuage.

3.2.1 Poste de travail local

Le développement principal, la création des interfaces graphiques et les tests d'inférence initiaux ont été effectués sur un ordinateur portable dont les caractéristiques sont les suivantes :

- **Processeur** : Intel® Core™ i5-1135G7@ 2.40 GHz.
- **Mémoire vive (RAM)** : 12 Go.
- **Stockage** : Disque SSD de 256 Go, garantissant un chargement rapide du système, des programmes et des ensembles de données.
- **Processeur Graphique (GPU)** : Intel® Iris® Xe Graphics (intégré).

Cette configuration a été suffisante pour le développement, les tests unitaires et la manipulation des fichiers image grâce au SSD rapide. Cependant, la faible puissance du GPU intégré a limité les capacités d'entraînement, rendant cette tâche trop lente en local. D'où le choix d'une solution cloud pour l'entraînement des modèles.

3.2.2 Plateforme d'entraînement Cloud (Google Colaboratory)

L'entraînement des modèles YOLO11(n,s,m,l,x) a été effectué sur Google Colaboratory, qui offre des notebooks avec GPU, notamment des NVIDIA Tesla T4. Cette solution a permis de réduire les temps d'entraînement de plusieurs jours à quelques heures, rendant le cycle d'expérimentation bien plus rapide et efficace.

3.2.3 Logiciels et Langages

a. Python Le projet a été développé en Python (v3.11), langage incontournable en IA grâce à sa syntaxe claire et ses nombreuses bibliothèques (NumPy, OpenCV, PyTorch,tkinter...).

b. Visual Studio Code Le code a été rédigé et débogué avec Visual Studio Code, un éditeur léger et efficace, enrichi d'extensions dédiées à Python.

c. Roboflow Pour la gestion du dataset, nous avons utilisé Roboflow, qui a facilité l'importation, l'annotation, le prétraitement et l'export des images au format YOLO.

3.3 Préparation des données

La qualité des données utilisées pour l'entraînement d'un système de reconnaissance automatique des plaques d'immatriculation (ANPR) est cruciale pour assurer sa robustesse et sa précision. Avant toute phase d'entraînement, il est indispensable de constituer un jeu de données représentatif, de l'annoter de manière précise, puis de le diviser en sous-ensembles destinés à l'entraînement, à la validation et aux tests. Cette section détaille les étapes de préparation des données adoptées dans ce projet.

3.3.1 Constitution des jeux de données

Pour concevoir un système ANPR efficace, il est essentiel de disposer d'un jeu de données riche et varié, contenant des images de véhicules avec leurs plaques d'immatriculation bien visibles dans des conditions diverses. Ces variations concernent notamment les angles de vue, la luminosité, la distance de prise de vue, les types de plaques (avant et arrière), ainsi que les caractéristiques des caractères (formats, police, espacement, etc.).

Dans le cadre de ce projet, un jeu de données personnalisé (custom dataset) a été constitué manuellement à partir de plusieurs sources (voir figure 3.1 , 3.2 et 3.3)

- Captures personnelles réalisées à l'aide d'un smartphone dans des environnements réels tels que des parkings, des routes ou des garages. Ces images ont été prises sous différents angles de vue (face, incliné) et à des distances variables (près et loin), afin de simuler des cas d'observation réalistes rencontrés en conditions réelles.



Figure 3.1 Image avec matricule incliné.



Figure 3.2 Image avec matricule arrière très proche et salle.



Figure 3.3 Image avec matricule en face.

- Les images ont été prises dans des conditions d'éclairage variées (en plein jour, à l'ombre), pour tester la robustesse du système à des conditions non idéales (voir figure 3.4, 3.5).



Figure 3.4 Exemple avec faible luminosité.



Figure 3.5 Exemple avec forte luminosité.

- Divers types de véhicules ont été inclus, présentant des plaques d'immatriculation de formats et de styles différents, avec des chiffres de polices, tailles et espacements variables (voir figure 3.6).



Figures 3.6 Exemples des différentes polices.

Pour des raisons de protection de la vie privée, chaque image collectée a nécessité une modification des plaques originales afin de ne pas exposer de données personnelles. Ce travail supplémentaire a limité le volume global de données : au total, 344 images ont pu être utilisées.

Malgré cette contrainte, le dataset a été enrichi progressivement, et l'effet de cette augmentation a été observé au fil des tests.

À chaque ajout de nouvelles images, une amélioration mesurable des performances du modèle a été constatée, aussi bien en détection des plaques qu'en reconnaissance des caractères, confirmant l'importance de la diversité et de la quantité de données dans l'apprentissage en profondeur.

3.3.2 Annotation des images avec Roboflow

L'annotation des images est une étape essentielle en vision par ordinateur supervisée, permettant de délimiter les zones d'intérêt par des boîtes englobantes associées à des étiquettes. Dans ce projet, les annotations visaient les plaques complètes et les caractères. La plateforme Roboflow a été utilisée pour cette tâche, offrant une interface intuitive pour l'annotation manuelle, ainsi que des outils d'exportation, de prétraitement et d'augmentation de données, facilitant la création d'un dataset adapté aux modèles YOLO.

Deux types d'annotations ont été effectués selon les approches envisagées :

- **Annotation 1 détection des plaques uniquement avec YOLO11**

Dans cette configuration, chaque image a été annotée avec une seule classe, nommée `license_plate`, représentant la plaque entière, qu'elle soit avant ou arrière. Ce jeu d'annotations a servi à entraîner un modèle YOLOv11 dédié uniquement à la détection des plaques. Il a également été utilisé dans une approche combinée avec un système de reconnaissance optique de caractères (OCR) pour effectuer la reconnaissance des chiffres après détection (voir figure 3.7).



Figure 3.7 Exemple annotation pour la détection seulement.

- **Annotation 2 détection et reconnaissance des caractères avec YOLO11**

Dans cette approche, les plaques ont été découpées manuellement, puis chaque caractère (chiffres de 0 à 9) a été annoté individuellement. En plus des chiffres, trois classes supplémentaires ont été introduites, back pour les plaques arrière, front pour les plaques avant, none pour ignorer les zones indésirables. → Cette annotation plus fine, avec 13 classes au total, a permis d'entraîner un modèle YOLO11 non seulement à détecter les plaques, mais également à reconnaître les caractères numériques en les traitant comme des objets distincts (voir figure 3.8).

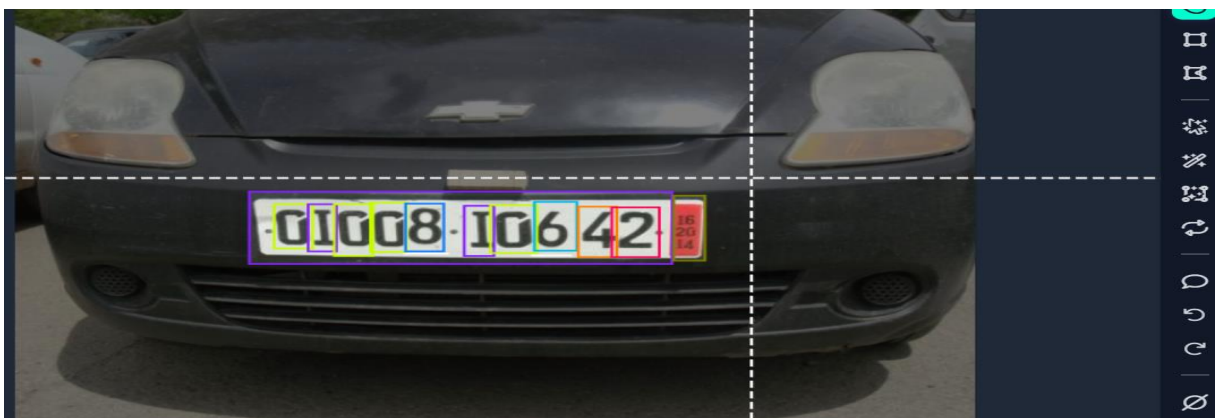


Figure 3.8 Exemple annotation pour la détection et reconnaissance.

3.3.3 Organisation en ensembles d'entraînement, validation et test

Afin d'entraîner, valider et évaluer efficacement notre modèle de détection de plaques d'immatriculation, l'ensemble de données initial, constitué de 344 images, a été réparti selon une stratégie classique en apprentissage supervisé. Ainsi, 278 images (soit 80 %) ont été utilisées pour constituer le jeu d'entraînement, servant à l'apprentissage du modèle. Ensuite, 33 images (10 %) ont été affectées au jeu de validation. Enfin, 33 images (10 %) ont été utilisées comme jeu de test temporaire, destiné à suivre les résultats intermédiaires à chaque étape d'ajout de données. Toutefois, pour l'évaluation finale des performances du modèle, un jeu de test indépendant, composé de 255 images inédites (jamais vues par le modèle durant l'entraînement), a été spécialement préparé. Ce dernier a été utilisé après l'intégration complète du modèle dans l'interface utilisateur, afin de garantir une évaluation réaliste et objective de la capacité de généralisation du système.

3.4 Interface graphique avec tkinter

Une interface graphique interactive a été conçue avec Tkinter, pour tester et visualiser les performances d'un modèle de détection d'objets YOLO. L'interface permet à l'utilisateur de charger des images ou des vidéos, d'appliquer le modèle pour effectuer des prédictions, puis d'afficher les résultats avec des annotations visuelles.

L'utilisateur peut ajuster dynamiquement le seuil de confiance, parcourir les images via des boutons ou des raccourcis clavier, afficher les prédictions sous forme regroupée ou détaillée, et Consulter des statistiques de performance en temps réel (précision, nombre de prédictions correctes/incorrectes, etc.).

L'interface comprend également une fonctionnalité d'analyse comparative par seuils de confiance multiples, représentée sous forme de graphique intégré, facilitant l'évaluation du modèle. Elle supporte les fichiers image standard (.jpg, .png, .bmp) ainsi que les vidéos, et offre une interaction fluide via zoom souris et raccourcis clavier (voir figure 3.26).

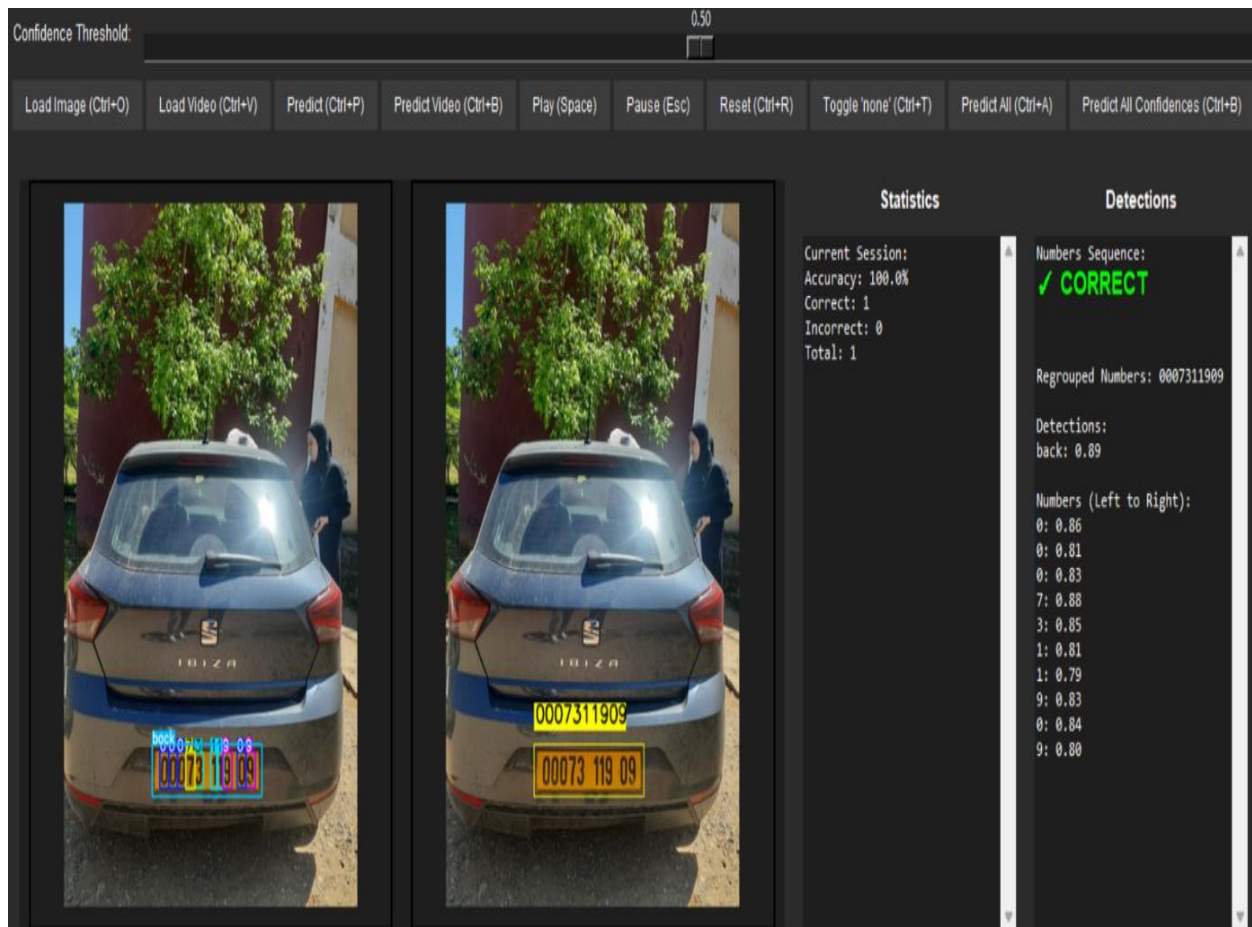


Figure 3.9 Affichage de l'image originale et annotée dans l'interface Tkinter.

3.5 Choix du modèle YOLO11l un compromis optimal entre précision et rapidité

Comme présenté précédemment dans le chapitre 2, plusieurs variantes du modèle YOLO11 ont été développées afin de répondre à différents besoins en termes de performances et de contraintes matérielles.

- **Les versions n , s et m** sont très rapides mais souffrent d'une perte notable en précision, en particulier pour les petits caractères ou les plaques partiellement visibles.
- **La version x** offre une meilleure précision, mais son coût computationnel élevé la rend peu adaptée à une exécution en temps réel sur des machines embarquées ou à ressources limitées.

- **La version YOLO11l** offre le meilleur compromis entre vitesse et précision, avec une bonne capacité de généralisation, une latence faible, et une excellente robustesse sur les cas réels de détection.

Ainsi, YOLO11l a été retenu comme le modèle le plus adapté pour l'implémentation finale de notre système de reconnaissance de plaques d'immatriculation, notamment dans une perspective d'exploitation en temps réel.

Dans le cadre de notre projet, l'ensemble des variantes du modèle YOLO11 a été évalué sur un même jeu de données de test composé de 255 images.

Les résultats expérimentaux ont montré que la version YOLO11l obtenait les meilleures performances, aussi bien en termes de taux de détection que d'identification correcte des plaques.

C'est sur la base de ces performances que YOLO11l a été retenu pour l'implémentation finale de notre système. Elle représente le compromis idéal entre précision, rapidité et robustesse, en parfaite adéquation avec les exigences de notre application embarquée de reconnaissance automatique de plaques d'immatriculation (voir figure 3.10, 3.11, 3.12, 3.13, 3.14).

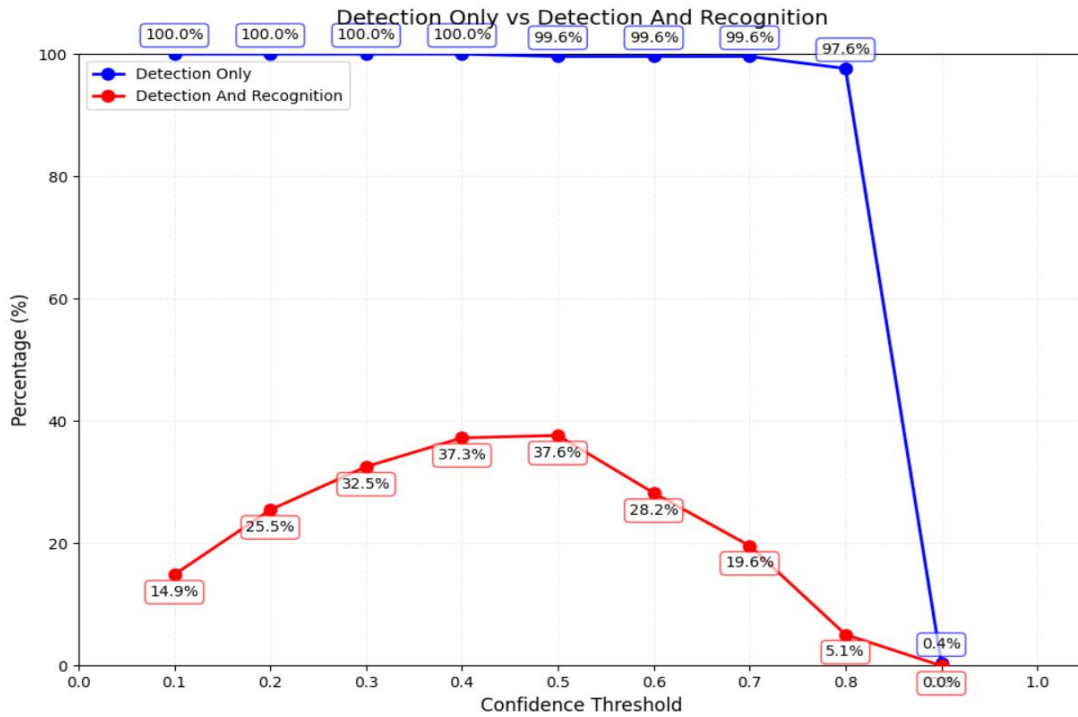


Figure 3.10 Taux de détections et reconnaissances correctes avec YOLO11n (255 images test).

- Le graphique montre que, pour le système de détection et de reconnaissance de plaques utilisant YOLO11n, testé sur un ensemble de 255 images avec des seuils de confiance compris entre 0,1 et 0,9, le taux de reconnaissance correct maximal atteint est de 37,6 %.

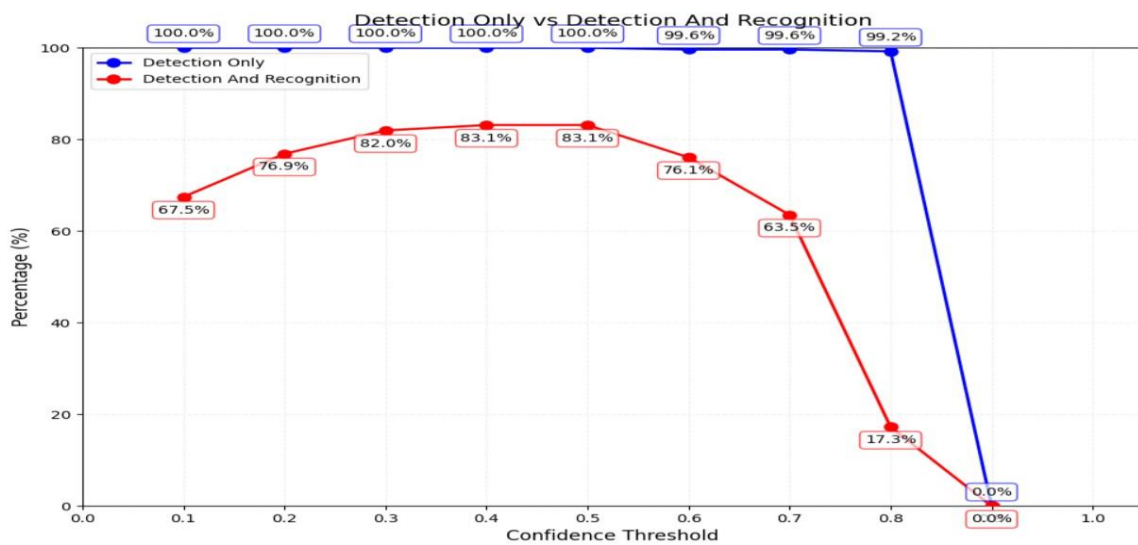


Figure 3.11 Taux de détections et reconnaissances correctes avec YOLO11s (255 images test).

- Le graphique montre que, pour le système de détection et de reconnaissance de plaques utilisant YOLO11s, testé sur un ensemble de 255 images avec des seuils de confiance compris entre 0,1 et 0,9, le taux de reconnaissance correct maximal atteint est de 83.1 %.

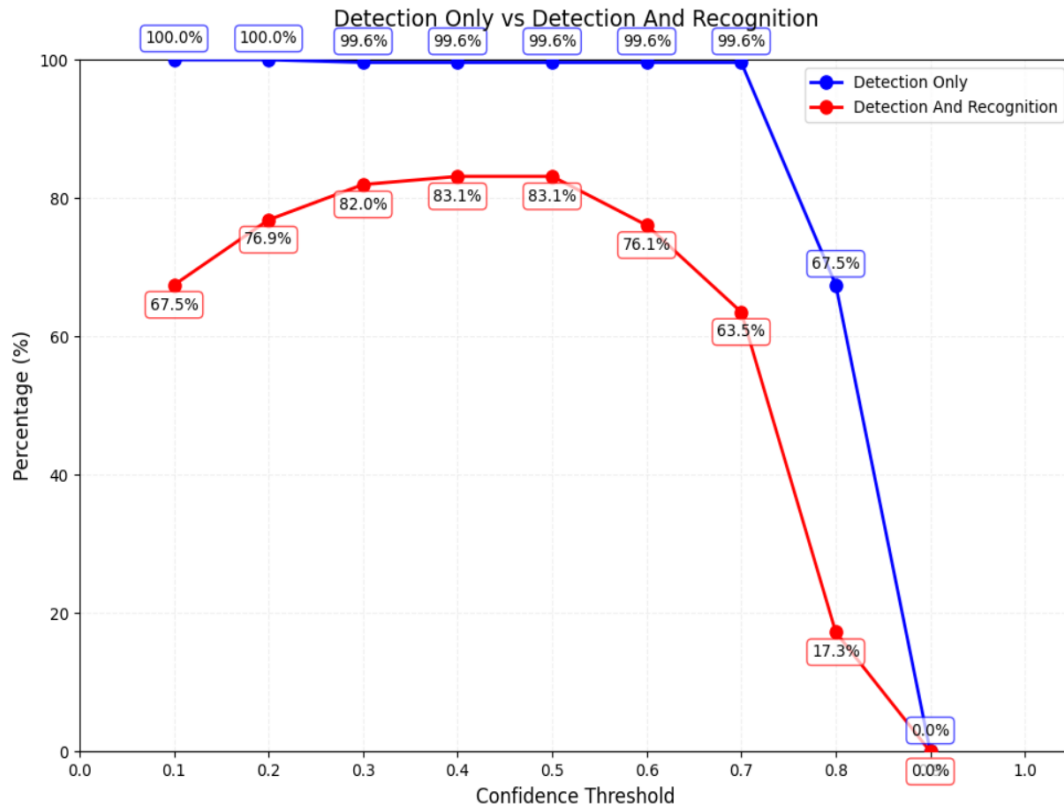


Figure 3.12 Taux de détections et reconnaissances correctes avec YOLO11m (255 images test).

- Le graphique montre que, pour le système de détection et de reconnaissance de plaques utilisant YOLO11m, testé sur un ensemble de 255 images avec des seuils de confiance compris entre 0,1 et 0,9, le taux de reconnaissance correct maximal atteint est de 83.1 %.

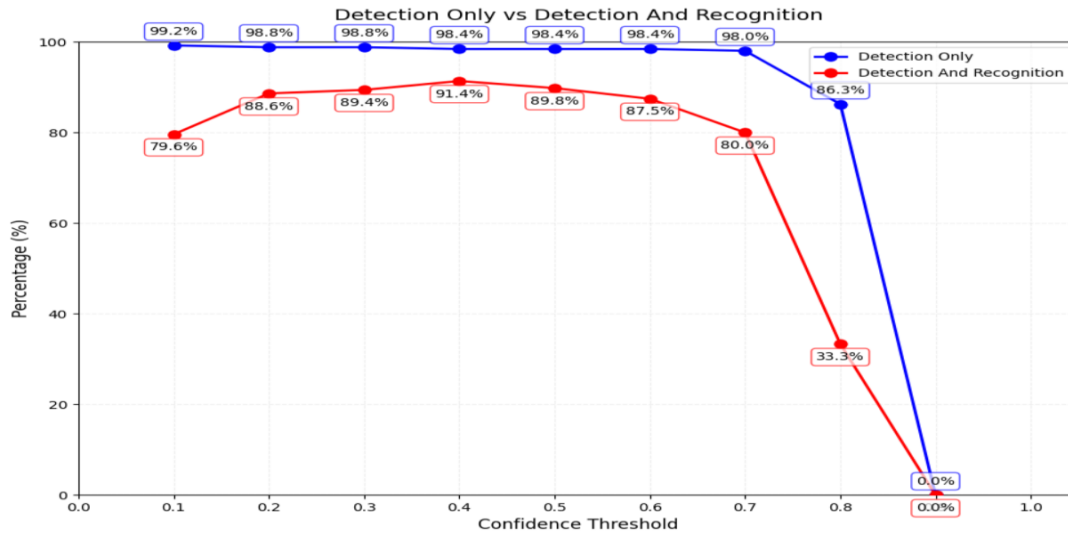


Figure 3.13 Taux de détections et reconnaissances correctes avec YOLO11l (255 images test).

- Le graphique montre que, pour le système de détection et de reconnaissance de plaques utilisant YOLO11l, testé sur un ensemble de 255 images avec des seuils de confiance compris entre 0,1 et 0,9, le taux de reconnaissance correct maximal atteint est de 91.4 %.

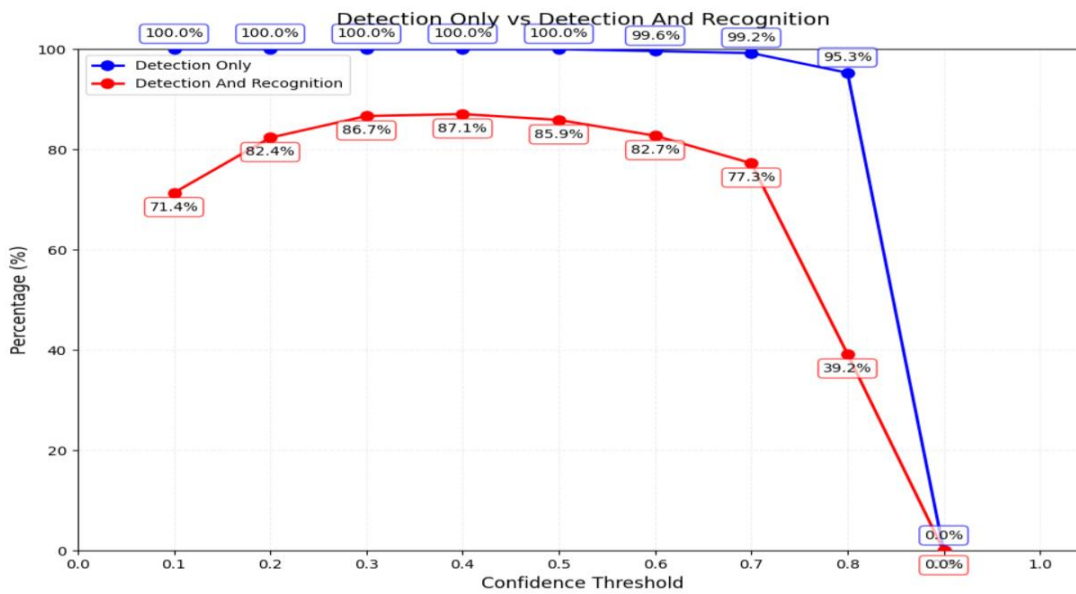


Figure 3.14 Taux de détections et reconnaissances correctes avec YOLO11x (255 images test).

- Le graphique montre que, pour le système de détection et de reconnaissance de plaques utilisant YOLO11x, testé sur un ensemble de 255 images avec des seuils de confiance compris entre 0,1 et 0,9, le taux de reconnaissance correct maximal atteint est de 87.1%.

3.5.1 Détection des plaques avec YOLO11l

Dans cette première approche, l'objectif est de détecter uniquement les plaques d'immatriculation des véhicules à partir d'images capturées dans des environnements réels. Cette étape constitue la base du système ANPR , en localisant précisément les zones contenant les plaques avant toute tentative de lecture ou de reconnaissance (voir figure 3.15).



Figure 3.15 Détection de la plaque avec YOLO11l.

3.5.2 Entraînement

L'entraînement du modèle YOLO11l pour la détection exclusive des plaques d'immatriculation avec une seule classe a été réalisé sur 100 époques à l'aide d'un ensemble de données annotées de tailles 640x640 pixels (voir figure 3.16).

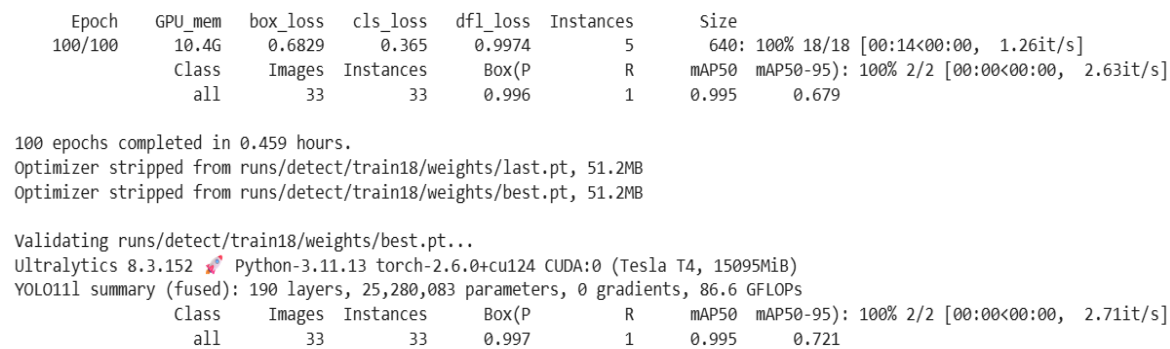


Figure 3.16 Résultats obtenus avec colab.

3.5.3 Résultats obtenus et interprétations

a. Courbes de performances

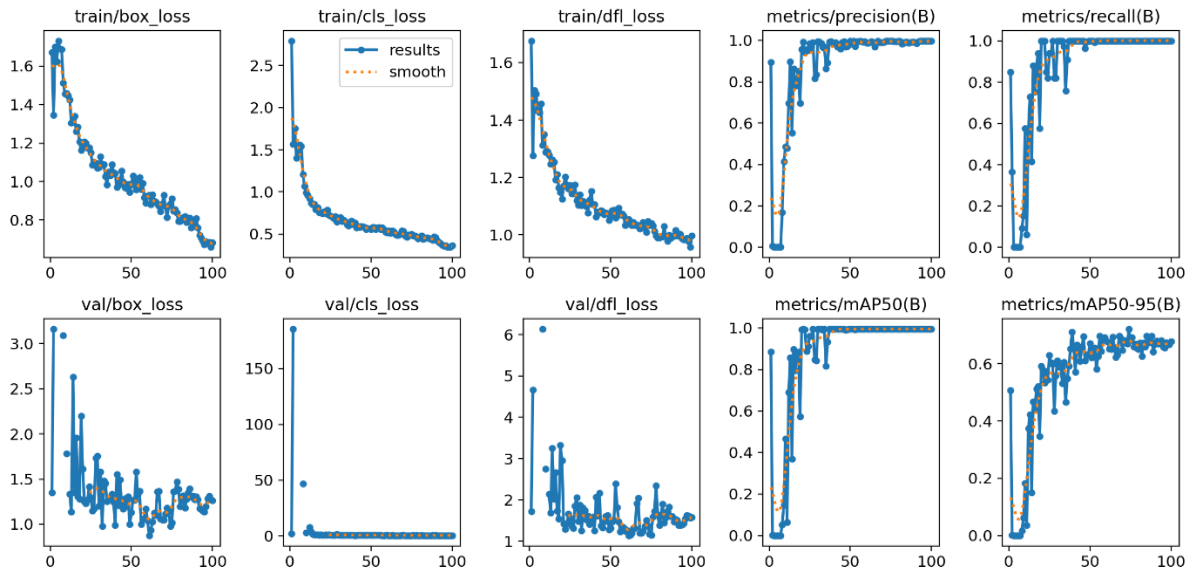


Figure 3.17 Résultats obtenus avec YOLO11l pour la détection (une classe).

L'analyse des courbes de performance montre une convergence progressive des fonctions de perte (box_loss, cls_loss, dfl_loss) aussi bien sur le jeu d'entraînement que sur le jeu de validation, signe d'un apprentissage efficace et sans surapprentissage (voir figure 3.17).

Les indicateurs de performance confirment ces résultats :

- **Précision (Box(P)) = 0.997 (99.7%)**

Presque toutes les détections proposées sont correctes (très peu de faux positifs).

- **Rappel (R) = 1.0 (100%)**

Le modèle a détecté toutes les instances cibles (aucun faux négatif).

- **mAP50 = 0.995 (99.5%)**

Excellente performance pour un seuil de IoU à 50%, indiquant une localisation très précise des objets.

- **mAP50-95 = 0.721 (72.1%)**

Performance légèrement inférieure pour des seuils de IoU plus stricts (50% à 95%), ce qui est normal et montre que la précision diminue lorsque les critères de localisation sont plus rigoureux.

Ces performances justifient pleinement le choix du modèle YOLO11l, qui s'est révélé particulièrement efficace et robuste pour la détection des plaques, tout en maintenant une latence faible adaptée au temps réel.

b. matrice de confusion

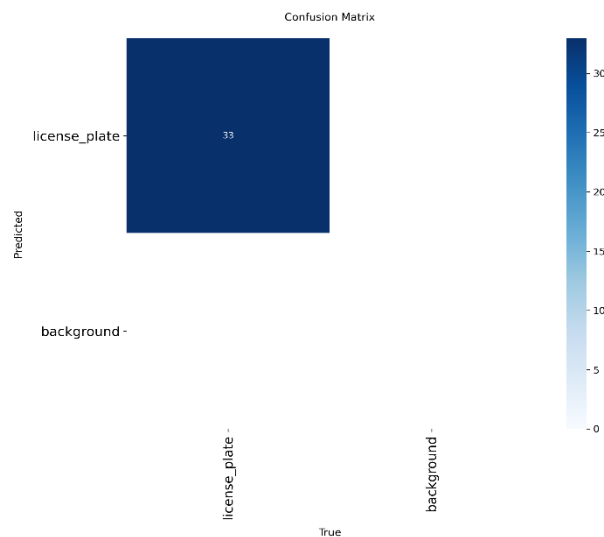


Figure 3.18 Matrice de confusion obtenu avec YOLO11l pour la détection (une classe).

La matrice de confusion montre L'analyse détaillée de la matrice de confusion révèle que le modèle a correctement identifié l'intégralité des 33 instances de plaques d'immatriculation (33 Vrais Positifs), sans aucune omission, comme l'indique la valeur de zéro pour les Faux Négatifs. Cette exhaustivité se traduit par un score de Rappel de 100%. Simultanément, la matrice de confusion confirme qu'aucune erreur de classification n'a été commise, avec zéro Faux Positif enregistré. Cela garantit une Précision de 100%, signifiant que chaque détection est correcte (voir figure 3.18).

c. Taux de détections correctes des plaques

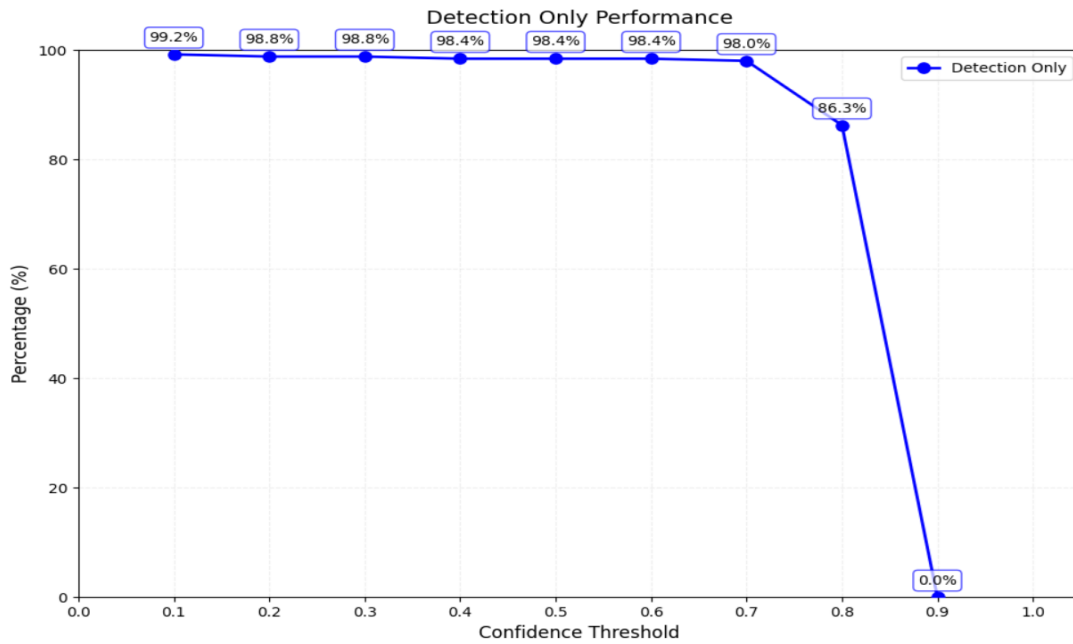


Figure 3.19 Taux de détections correctes avec YOLO11l pour la détection avec 255 images test.

Le graphique montre la performance du système de détection de plaques sur un ensemble de test de 255 images, en fonction du seuil de confiance utilisé (voir figure 3.19).

Analyse des résultats Pour des seuils de confiance compris entre 0.1 et 0.7, le taux de détection correcte des plaques reste très élevé, variant entre 98.0% et 99.2%. Cela signifie que, dans cette plage de seuils, le système détecte correctement presque toutes les plaques présentes dans les images testées.

- À un seuil de confiance de 0.8, la performance chute à 86.3%. Cela indique que le système devient beaucoup plus strict et rejette davantage de détections, ce qui augmente le risque de manquer des plaques valides.
- À un seuil de 0.9, le taux de détection tombe à 0%, ce qui veut dire qu'aucune plaque n'est détectée à ce niveau de confiance très élevé.

3.6 Détection et reconnaissance des plaques avec YOLO11

Une fois les plaques localisées avec précision, l'étape suivante consiste à reconnaître les caractères numériques qu'elles contiennent. Pour cela, le système est étendu afin de réaliser une tâche combinée de détection et reconnaissance des plaques, en utilisant également le modèle YOLO11 mais avec une configuration adaptée à une classification multi-classes plus fine.

Dans cette phase, le modèle est entraîné pour identifier non seulement les plaques avant et arrière, mais également les dix chiffres de 0 à 9. Ainsi, chaque caractère présent sur une plaque devient une instance détectable, et le modèle agit comme un extracteur de caractères individuels. La reconnaissance repose sur le découpage spatial précis des caractères, ce qui nécessite une haute résolution d'entrée ainsi qu'un seuil de confiance ajusté pour réduire les fausses détections (voir figure 3.20).

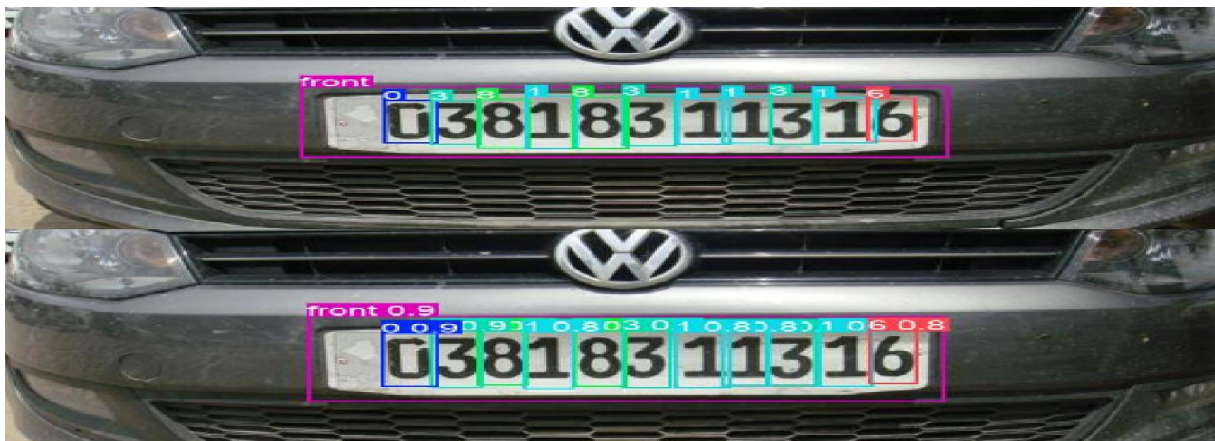


Figure 3.20 détection et reconnaissance de la plaque avec YOLO11.

3.6.1 Entraînement

L'entraînement du modèle YOLOv11-L pour la détection et reconnaissance des plaques d'immatriculation avec 13 classes a été réalisé sur 100 époques à l'aide d'un ensemble de données annotées de tailles 640x640 pixels (voir figure 3.21).

\Class\	Images	Instances	Box(P	R	mAP50	mAP50-95): 100% 2/2 [00:00<00:00, 2.55it/s]
all	33	385	0.964	0.965	0.991	0.654
0	29	53	0.97	1	0.995	0.628
1	33	99	0.984	0.99	0.994	0.614
2	17	26	0.962	1	0.995	0.665
3	15	24	0.906	1	0.985	0.658
4	17	23	0.98	0.957	0.992	0.675
5	12	17	0.944	0.994	0.992	0.668
6	25	31	0.968	0.967	0.993	0.634
7	16	20	0.972	1	0.995	0.701
8	17	20	0.957	0.95	0.965	0.661
9	20	33	1	0.968	0.995	0.631
back	12	12	0.915	1	0.995	0.741
front	21	21	0.971	1	0.995	0.714
none	4	6	1	0.715	0.995	0.513

Figure 3.21 résultats obtenus avec colab.

3.6.2 Résultats obtenus et interprétations

a. courbes de performances

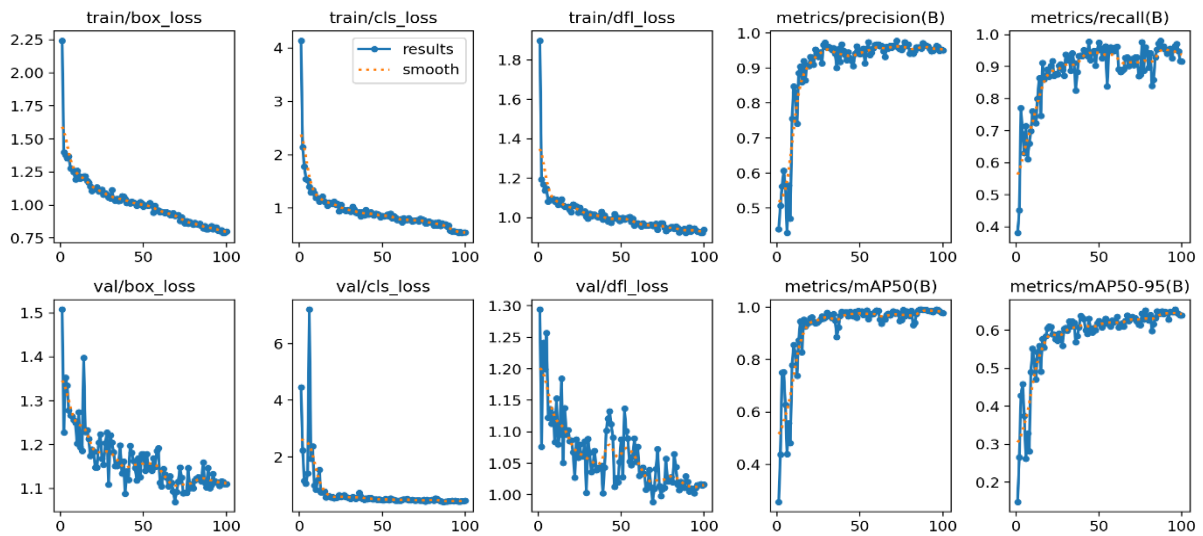


Figure 3.22 Résultats obtenus avec YOLO11l pour la détection et reconnaissance (13 classes).

Le modèle YOLO11l appliqué à la détection et reconnaissance de plaques d'immatriculation montre d'excellents résultats avec une $mAP@50$ de 99,1 % et une $mAP@50-95$ de 65,4 %. Les classes alphanumériques sont bien détectées, avec des précisions et rappels généralement supérieurs à 0,95. Les plaques avant et arrière sont également bien identifiées. Seule la classe "none" présente un rappel plus faible (0,715), suggérant quelques erreurs de détection. Les courbes d'apprentissage montrent une convergence rapide et stable sans surapprentissage, confirmant la robustesse du modèle pour ce type de tâche (voir figure 3.22).

b. matrice de confusion

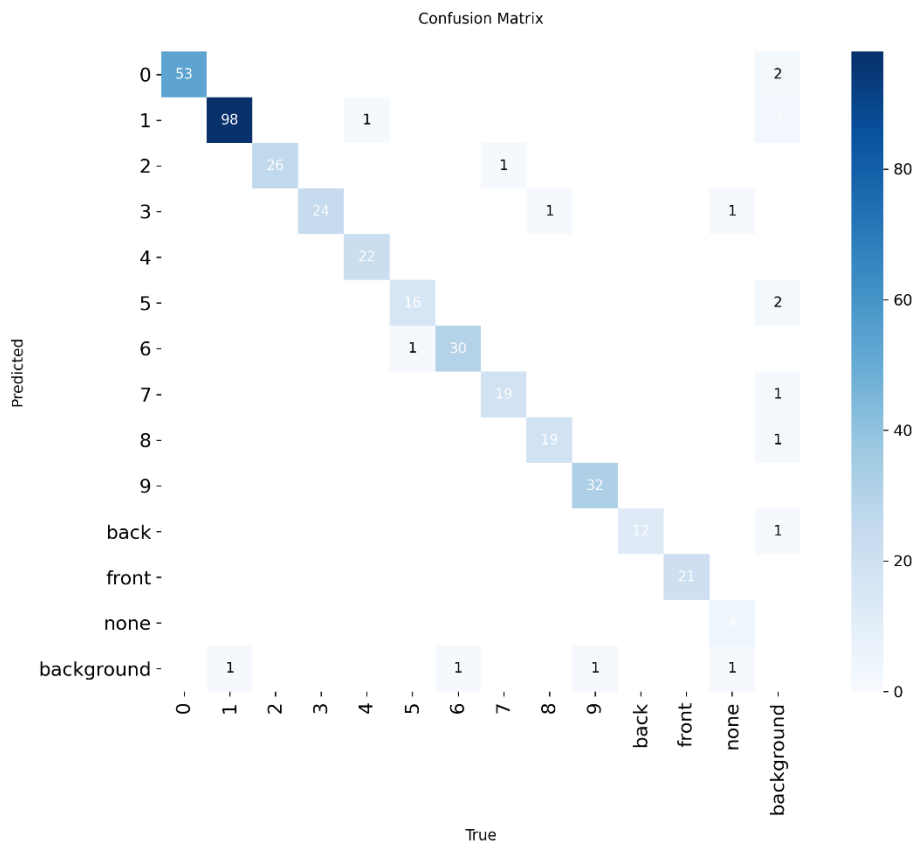


Figure 3.23 Matrice de confusion obtenu avec YOLO11l pour la détection (13 classes).

La matrice de confusion montre : La matrice de confusion montre une très bonne performance globale du modèle YOLO11l dans la reconnaissance des caractères de plaques d'immatriculation. Les classes prédominantes (chiffres de 0 à 9, ainsi que les métaclasse back, front, none, et background) sont majoritairement bien classées, avec une forte diagonale

principale, signe de bonnes prédictions. Les classes 1, 2, 3, etc. présentent des taux de reconnaissance corrects (ex : 1 avec 98 bonnes prédictions). Les confusions sont peu fréquentes et faiblement distribuées, témoignant d'un bon apprentissage inter-classes. Quelques erreurs mineures subsistent (ex. 1 prédit à la place de none, 0 confondu avec background), ce qui peut être dû à la similitude visuelle entre certains caractères ou à une qualité d'image dégradée (voir figure 3.23).

c. Taux de reconnaissances correctes des plaques

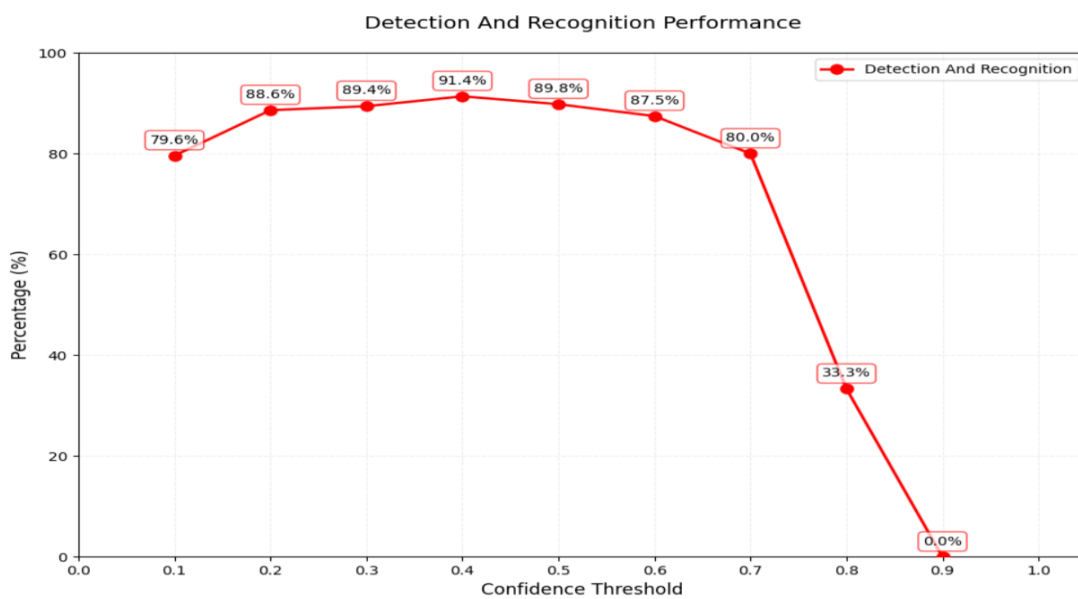


Figure 3.24 Taux de détections et reconnaissances correctes avec YOLO11l avec 255 images test.

Le graphique montre la performance du système de détection et reconnaissance de plaques sur un ensemble de test de 255 images, en fonction du seuil de confiance utilisé (voir figure 3.24).

Analyse des résultats Pour des seuils de confiance compris entre 0.1 et 0.7, le taux de détection correcte des plaques reste très élevé, variant entre 79.6 % et 91.4%. Cela signifie que, dans cette plage de seuils, le système montre une bonne performance globale mais pourrait être optimisé.

- Le meilleur résultat est 91.4%, indiquant que le modèle détecte et reconnaît correctement plus de 9 objets sur 10 dans certaines conditions.

- À un seuil de confiance de 0.8, la performance chute à 33.3%. Cela indique que le système devient beaucoup plus strict et rejette davantage de détections, ce qui augmente le risque de manquer des plaques valides.
- À un seuil de 0.9, le taux de détection tombe à 0%, ce qui veut dire qu'aucune plaque n'est détectée à ce niveau de confiance très élevé .

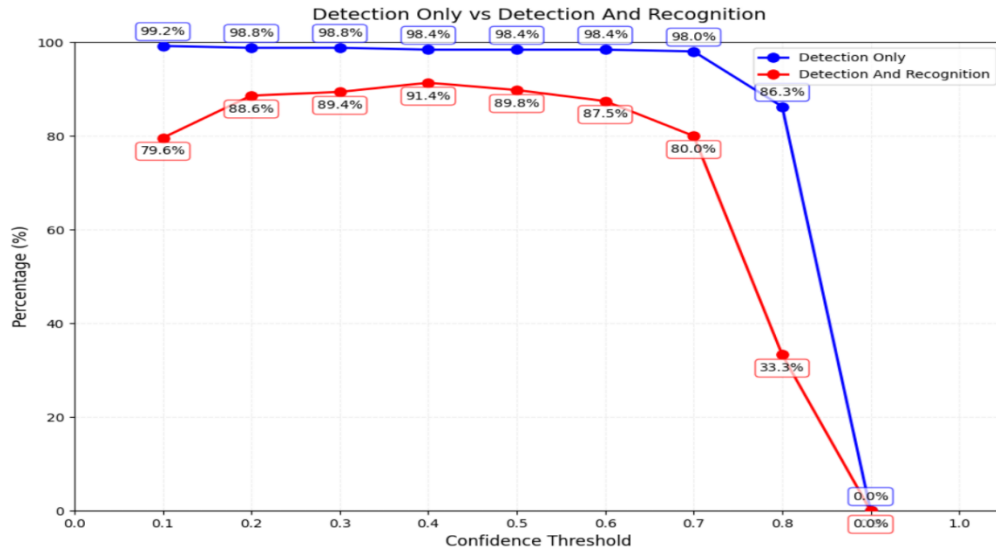


Figure 3.25 Taux de détections et reconnaissances correctes avec YOLO11l par rapport à la détection avec 255 images test.

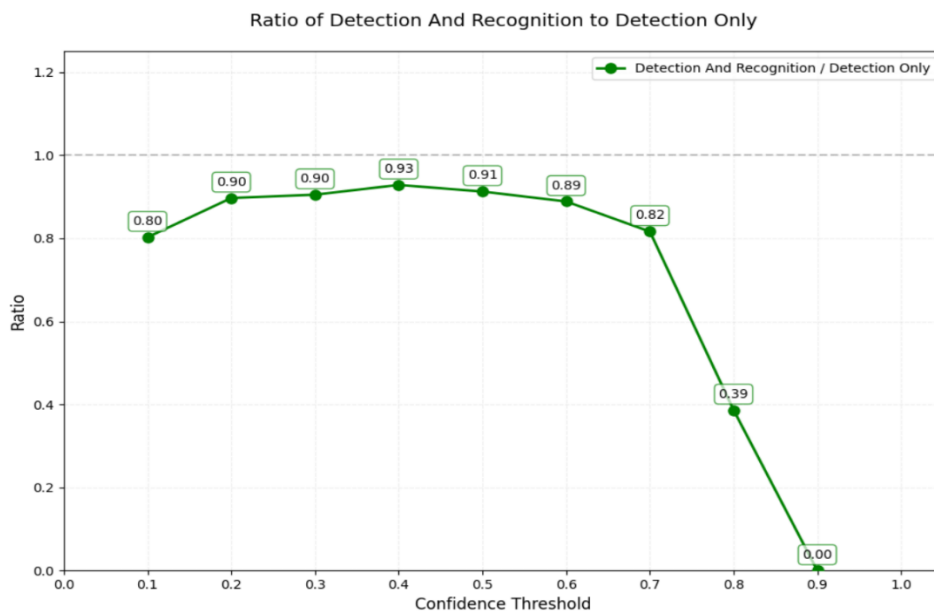


Figure 3.26 Rapport des résultats de détections et reconnaissances correctes avec yolo11-L sur résultats de la détection avec YOLO11l (avec 255 images test).

Tableau 3.1 Résultats avec YOLO11l pour la détection et reconnaissance avec 255 images test.

Seuil de confiance	Nombre d'Identifications des matricules Correctes	Nombre d'Identifications des matricules Incorrectes	accuracy
0.1	203	52	79.6%
0.2	226	29	88.6%
0.3	228	27	89.4%
0.4	233	22	91.4%
0.5	229	26	89.8%
0.6	223	32	87.5%
0.7	204	51	80.0%
0.8	85	170	33.3%

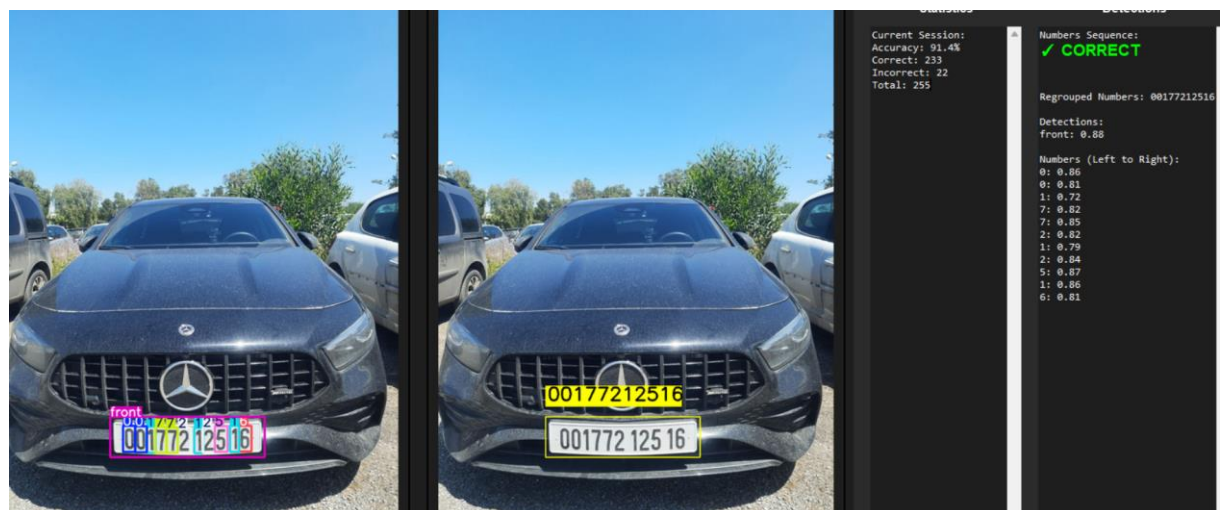


Figure 3.27 Exemple de détection et reconnaissance de la plaque avec YOLO11l.

3.7 Détection des plaques avec YOLO11l et reconnaissance avec OCR

Dans cette approche, nous utilisons YOLO11l pour détecter avec précision les plaques, et deux outils d'OCR pour la reconnaissance : EasyOCR et Tesseract (voir figure 3.25).



Figure 3.27 Détection des plaques avec YOLO11l et reconnaissance avec OCR(Tesseract).

3.7.1 Reconnaissance avec EasyOCR

Dans cette expérimentation, EasyOCR est utilisé pour extraire automatiquement les caractères présents sur les plaques détectées par YOLO11l. EasyOCR repose sur une architecture CRNN (Convolutional Recurrent Neural Network) combinée à une couche CTC (Connectionist Temporal Classification), ce qui lui permet de reconnaître du texte même lorsque les caractères sont légèrement déformés ou partiellement visibles [45].

Afin d'évaluer les performances du système, différents seuils de confiance ont été testés lors de la détection des plaques. À chaque seuil, on a mesuré le nombre de plaques dont les caractères ont été correctement identifiés, ainsi que ceux ayant donné lieu à une reconnaissance erronée. L'accuracy a ensuite été calculée comme le rapport entre les bonnes reconnaissances et le total des reconnaissances tentées (255 images de test).

Tableau 3.2 : Résultats avec YOLO11l pour la détection et EasyOCR pour reconnaissance avec 255 images test.

Seuil de confiance	Nombre d'Identifications des matricules Correctes	Nombre d'Identifications des matricules Incorrectes	accuracy
0.1	159	96	62.4%
0.2	159	96	62.4%
0.3	159	96	62.4%
0.4	159	96	62.4%
0.5	159	96	62.4%
0.6	159	96	62.4%
0.7	158	97	62.0%
0.8	142	113	55.7%

3.7.2 Reconnaissance avec Tesseract

Tesseract est un moteur OCR open-source développé initialement par HP et maintenu par Google. Il repose sur des réseaux de neurones LSTM et fonctionne particulièrement bien sur du texte imprimé clair et bien segmenté. Contrairement à EasyOCR, il est plus sensible au bruit et nécessite souvent un prétraitement de l'image (niveaux de gris, seuillage) [20].

L'évaluation de Tesseract a été effectuée dans les mêmes conditions expérimentales que pour EasyOCR, avec les mêmes images de plaques issues des détections YOLO11l. Le tableau ci-dessous présente les performances mesurées à différents seuils de confiance.

Tableau 3.3 : Résultats avec YOLO11l pour la détection et Tesseract pour reconnaissance avec 255 images test.

Seuil de confiance	Nombre d'Identifications des matricules Correctes	Nombre d'Identifications des matricules Incorrectes	accuracy
0.1	138	117	54.1%
0.2	137	118	53.7%
0.3	137	118	53.7%
0.4	137	118	53.7%
0.5	137	118	53.7%
0.6	137	118	53.7%
0.7	137	118	53.7%
0.8	127	128	49.8%

3.8 Etude comparative

La détection et reconnaissance des plaques avec YOLO11l en une seule étape offre une exécution rapide et une précision élevée, car le modèle apprend simultanément à localiser les plaques et à reconnaître les caractères. En revanche, l'approche combinant YOLO11l pour la détection et un OCR pour la reconnaissance est plus flexible : chaque module peut être optimisé séparément, ce qui la rend plus robuste face aux variations (polices, éclairage), mais elle est souvent plus lente. Le choix dépend donc des priorités entre vitesse et adaptabilité.

Tableau 3.4 : Comparaison des 3 approches.

Approche	Seuil de confiance 0.4			Seuil de confiance 0.8		
	Nombre d'Identifications des matricules Correctes	Nombre d'Identifications des matricules Incorrectes	accuracy	Nombre d'Identifications des matricules Correctes	Nombre d'Identifications des matricules Incorrectes	accuracy
YOLO11l Détection + reconnaissance	233	22	91.4%	85	170	33.3%
YOLO11l Détection+ EasyOCR pour reconnaissance	159	96	62.4%	142	113	55.7%
YOLO11-L Détection+ Tesseract pour reconnaissance	137	118	53.7%	127	128	49.8%

Le modèle YOLO11l, en approche (détection + reconnaissance), montre une précision élevée à seuil bas (91.4% à 0.4), surpassant largement les méthodes utilisant EasyOCR (62.4%) ou Tesseract (53.7%). Toutefois, sa précision chute à 33.3% à seuil 0.8, révélant un manque de confiance dû à un entraînement insuffisamment varié. Cela suggère que bien que YOLO11l soit plus performant que l'OCR, il nécessite une optimisation par l'ajout de données diversifiées pour améliorer sa robustesse à des seuils plus stricts.

3.9 Conclusion

Ce chapitre a présenté le développement d'un système de reconnaissance automatique de plaques d'immatriculation, basé sur YOLO11l et des techniques OCR. Trois approches ont été mises en œuvre : détection seule, détection avec reconnaissance intégrée, et détection suivie de reconnaissance via EasyOCR et Tesseract.

Les résultats expérimentaux démontrent que l'approche tout-en-un utilisant YOLO11l pour la détection et la reconnaissance des caractères offre les meilleures performances globales en termes de précision (jusqu'à 91,4 % à seuil modéré), tout en assurant une exécution rapide, ce qui la rend particulièrement adaptée aux applications en temps réel. L'approche basée sur OCR, bien que plus flexible, présente des performances moindres, notamment en présence de conditions complexes (bruit, inclinaison, faible résolution).

Le choix de la version YOLO11l s'est révélé pertinent, offrant un compromis équilibré entre précision et rapidité, et se montrant robuste face à une grande variété de situations d'acquisition d'images.

L'implémentation d'une interface graphique avec Tkinter a permis de valider le système dans un contexte d'utilisation réel, en facilitant les tests, la visualisation et l'évaluation des performances selon différents seuils.

En définitive, cette phase d'implémentation a permis de confirmer l'efficacité des réseaux de neurones convolutifs pour les tâches de détection et de reconnaissance dans un contexte ANPR, tout en mettant en évidence l'impact crucial de la qualité et de la diversité des données d'entraînement. Ces résultats ouvrent la voie à des améliorations futures, notamment par l'enrichissement du dataset et l'optimisation du système pour une utilisation embarquée à grande échelle.

Conclusion Générale

Conclusion générale

Dans un contexte où la sécurité, la gestion du trafic et l'automatisation des systèmes urbains deviennent des enjeux cruciaux, ce mémoire a proposé une solution intelligente et efficace pour la détection et la reconnaissance automatique des plaques d'immatriculation (ANPR), en s'appuyant sur les avancées récentes en apprentissage profond, notamment le modèle YOLO11.

Tout au long de ce travail, nous avons présenté les fondements théoriques des systèmes de reconnaissance de plaques, depuis les principes du traitement d'image jusqu'aux réseaux de neurones convolutifs et aux modèles de détection en temps réel. L'intégration de YOLO11, pour ses performances en rapidité et en précision, a permis de mettre en œuvre un système capable de détecter et reconnaître des plaques. Trois approches ont été explorées : la détection seule, la détection et reconnaissance par YOLO11, et une méthode hybride combinant YOLO11 et un module OCR (EasyOCR/Tesseract).

Les résultats obtenus, à travers une base de données personnalisée et des tests expérimentaux variés, ont démontré la pertinence de l'approche choisie. Le modèle YOLO11, en particulier, a montré un excellent compromis entre rapidité d'exécution et taux de reconnaissance, ce qui le rend adapté à des applications en temps réel et sur des systèmes embarqués.

Ce projet ouvre la voie à plusieurs perspectives d'amélioration, telles que l'élargissement du dataset, ou encore le déploiement sur des systèmes embarqués à faible consommation.

- [1] Euronews, Rétromachine : apparition de la plaque d'immatriculation, consulté le 15mai2025, disponible sur <https://fr.euronews.com/2013/08/14/retromachine-apparition-de-la-plaque-d-immatriculation> .
- [2] Cockpit, Histoire des plaques d'immatriculation algériennes, consulté le 15mai2025, <https://www.cockpitz.com/post/histoire-des-plaques-d-immatriculation-alg%C3%A9riennes>
- [3] Istockphoto, Collection of European license plates from various countries, consulté le 15mai2025, disponible sur <https://www.istockphoto.com/photo/collection-of-european-license-plates-from-various-countries-gm813166440-131513249?searchscope=image%2Cfilm>
- [4] Cockpit, Normes des plaques d'immatriculation algériennes (MÀJ janvier 2023), consulté le 20 mai2025, disponible sur <https://www.cockpitz.com/post/normes-des-plaques-d-immatriculation-alg%C3%A9riennes>.
- [5] Pr A.NAMANE ,cours de vision artificielle, département d'électronique/université blida1, 2024/2025.
- [6] wikipedia, Plaque d'immatriculation algérienne, consulté le 20 mai2025, disponible sur https://fr.wikipedia.org/wiki/Plaque_d%27immatriculation_alg%C3%A9rienne .
- [7] Researchgate , Sample images of Algerian license plates, consulté le 20 mai2025,disponible sur https://www.researchgate.net/figure/Sample-images-of-Algerian-license-plates_fig2_373951480 .
- [8] Abirami Vina, Utilisation de Ultralytics YOLO11 pour la reconnaissance automatique des plaques minéralogiques, consulté le 20 mai2025, disponible sur <https://www.ultralytics.com/fr/blog/using-ultralytics-yolo11-for-automatic-number-plate-recognition> .
- [9] BENREZGANE, MOHAMED BOUHARKET, and ABDENACER BIAZ. Système d'Identification de Personnes via la Plaque d'Immatriculation de leurs Véhicules. Master. Université Ibn Khaldoun-Tiaret-, 2022.
- [10] AL-BATAT, Reda, ANGELOPOULOU, Anastassia, PREMKUMAR, Smera, et al. Un système de reconnaissance automatisée de plaques d'immatriculation de bout en bout utilisant la détection de véhicules et de plaques d'immatriculation basée sur YOLO avec classification des véhicules. Capteurs , 2022.
- [11] Hina T, Je vais détecter et reconnaître automatiquement les plaques d'immatriculation à l'aide de yolo, CNN, consulté le 20 mai2025, disponible sur https://fr.fiverr.com/hina_tufail/do-automatic-number-plate-recognition-using-yolo-cnn .
- [12] BENACHOUR, Y., & ANEDE, S. Conception et Réalisation d'un Système de Reconnaissance de Plaque d'Immatriculation de Véhicules en Temps Réel .Master, Université SAAD DAHLAB-BLIDA-,2016.
- [13] CHHAYDER, Ameer et MOHAMED, Imene BELHADJ. Système de Reconnaissance Automatique des Plaques Minéralogiques. In : SETIT 2009, 5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. 2009.

- [14] Jatniwalafizza, Matrice de confusion en action : études de cas dans les domaines de la santé, de la finance et du commerce électronique , consulté le 20 mai 2025, disponible sur <https://medium.com/@jatniwalafizza786/confusion-matrix-in-action-case-studies-in-healthcare-finance-and-e-commerce-1d89515ed4c3> .
- [15] Slimane AKLI, Immatriculation des véhicules en Algérie : Merad annonce de nouvelles mesures, consulté le 20 mai 2025, disponible sur <https://dnalgerie.com/immatriculation-des-vehicules-en-algerie-merad-annonce-de-nouvelles-mesures/> .
- [16] KHANAM, Rahima et HUSSAIN, Muhammad. Yolov11: An overview of the key architectural enhancements, 2024.
- [17] Abirami Vina , Construire un système ANPR avec Ultralytics YOLO11 et GPT-4o Mini, consulté le 20 mai 2025, disponible sur <https://www.ultralytics.com/fr/blog/build-an-anpr-system-with-ultralytics-yolo11-and-gpt-4o-mini> .
- [18] Pham, TA. Réseaux neuronaux profonds efficaces pour la détection et la reconnaissance des plaques d'immatriculation. *The Visual Computer* , 2023.
- [19] Redmon, J., Divvala, S., Girshick, R., et Farhadi, A. « On ne regarde qu'une fois : Détection d'objets unifiée et en temps réel ». Dans les actes de la conférence IEEE sur la vision par ordinateur et la reconnaissance de formes , 2016.
- [20] Smith, R. Présentation du moteur OCR Tesseract. *Neuvième conférence internationale sur l'analyse et la reconnaissance de documents* , 2007.
- [21] Canny, J. Une approche computationnelle de la détection des contours. *Transactions IEEE sur l'analyse des motifs et l'intelligence artificielle* , 1986.
- [22] Otsu, N. A threshold selection method from gray-level histograms. *Automatica* , 1975.
- [23] Open Source Computer Vision, Morphological Transformations, consulté le 20 mai 2025, disponible sur https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html .
- [24] KRIZHEVSKY, Alex, SUTSKEVER, Ilya, et HINTON, Geoffrey E. Classification ImageNet avec réseaux neuronaux convolutifs profonds. *Communications de l'ACM* , 2017.
- [25] zilliz, What is a Convolutional Neural Network? An Engineer's Guide, consulté le 01 juin 2025, disponible sur <https://zilliz.com/glossary/convolutional-neural-network> .
- [26] Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review* , 2024.

- [27] Jim James, An intuitive journey to Object Detection through Human Vision, Computer Vision and CNN's, consulté le 01 juin 2025, disponible sur <https://medium.com/codex/an-intuitive-journey-to-object-detection-through-human-vision-computer-vision-and-cnns-58d15ac6578c> .
- [28] Dubey, SR, Singh, SK, et Chaudhuri, BB. Fonctions d'activation en apprentissage profond : une étude exhaustive et une référence. Neurocomputing , 2022.
- [29] Gholamalinezhad, H., et Khosravi, H. Méthodes de regroupement dans les réseaux neuronaux profonds : une revue. 2009.
- [30] Jeczmioneck, E., et Kowalski, PA . Élagage des couches d'aplatissement dans les réseaux de neurones convolutifs. Symétrie, 2021.
- [31] Rany ElHousieny, PhD^{ABD} , Understanding Convolutional Neural Networks (CNNs): The Powerhouse of Image Processing, consulté le 04 juin 2025, disponible sur <https://www.linkedin.com/pulse/understanding-convolutional-neural-networks-cnns-elhousieny-phd%E1%B4%AC%E1%B4%AE%E1%B4%B0-uehc> .
- [32] Yannick Serge Obam , Comprendre les Réseaux de Neurones Convolutifs (CNN), consulté le 04 juin 2025, disponible sur <https://yannicksergeobam.medium.com/comprendre-les-r%C3%A9seaux-de-neurones-convolutifs-cnn-d5f14d963714> .
- [33] Tibor Udvari , Les réseaux de neurones , consulté le 04 juin 2025, disponible sur https://ml4a.github.io/ml4a/fr/neural_networks/.
- [34] Jegham, N., Koh, CY, Abdelatti, M., & Hendawi, A. YOLO Evolution : une analyse comparative et une analyse architecturale complètes de YOLOv12, YOLOv11 et leurs versions précédentes. 2024.
- [35] Khanam, R., & Hussain, M. A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions. 2025.
- [36] Daniel, E. R. Wildfire smoke detection with computer vision. 2023.
- [37] Équipe Blent Data Scientist , YOLO : détection sur les images avec TensorFlow , consulté le 06 juin 2025, disponible sur <https://blent.ai/blog/a/detection-images-yolo-tensorflow>.

- [38] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 2023.
- [39] Kshitiz Verma ,YOLO: Anchor boxes ,consulté le 06 juin 2025, disponible sur <https://www.youtube.com/watch?v=84BL6YNdntI> .
- [40] Tharun Sivamani ,NMS: Non Maximum Suppression,consulté le 06 juin 2025, disponible sur <https://blog.cubed.run/nms-non-maximum-suppression-157be5bc61ca> .
- [41] Glenn Jocher et Jing Qiu, principaux développeurs de YOLO11, Ultralytics YOLO11 ,consulté le 06 juin 2025, disponible sur <https://docs.ultralytics.com/fr/models/yolo11> .
- [42] Zhu, W., Han, X., Zhang, K., Lin, S., et Jin, J. Application du modèle YOLO11 à la convolution par dilatation pyramidale spatiale (SPD-Conv) et à la fusion par compression-excitation effective (EffectiveSE) pour la détection des défauts de voie ferrée. *Capteurs* , 2025.
- [43] Nisha Arya Ahmed ,Qu'est-ce qu'une matrice de confusion dans l'apprentissage automatique ? L'outil d'évaluation des modèles expliqué, consulté le 06 juin 2025, disponible sur <https://www.datacamp.com/fr/tutorial/what-is-a-confusion-matrix-in-machine-learning> .
- [44] XuBing, W., & Yan, C. Empirical evaluation of rectified activations in convolution network. 2015.
- [45] documentprocessing , OCR avancé pour les défis documentaires modernes,consulté le 06 juin 2025, disponible sur <https://products.documentprocessing.com/fr/parser/python/easyocr/> .