

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

**Faculté des Mathématiques, d'Informatique et des Sciences de la
matière**

Département d'Informatique



Filière : Informatique

Option : Sciences et technologie de l'information et de la communication

Thème :

**Conception et implémentation d'un système de contrôle d'accès pour
les péages autoroutiers à l'aide de la reconnaissance des plaques
d'immatriculation**

Présenté par :

Haroun Talhi

N	Nom et prénom	Qualité
1	Chemseddine Chohra	Président
2	Hallaci Samir	Encadreur
3	Hadjeris mourad	examineur

Juin 2024

Remerciement

Merci, Allah, de m'avoir donné la capacité de penser et d'écrire, le courage de tenir bon dans mes convictions et la persévérance pour mener à bien mes rêves.

Je voudrais exprimer ma profonde gratitude à Dr. Samir Hallaci, mon superviseur, pour avoir encadré la réalisation de cette tâche et m'avoir apporté le meilleur de ses connaissances et de son aide.

Je tiens également à exprimer ma gratitude aux membres du jury pour nous avoir donné le privilège de juger ce travail.

Je tiens à exprimer ma gratitude à tous les professeurs du département d'informatique, en particulier : Pr. Farou Brahim, Dr. Chohra Chemseddin, Dr. Kouahla Zinneddine, Dr. Abderrahmane Kefali, Pr. Hamid Seridi, Dr. Mahnaoui, Dr. Zdadra Warda et tous ceux qui ont contribué à la création de ce projet.

Enfin, je tiens à remercier ma mère, mon père, toute ma famille, mes amis, mes collègues de l'Université 08 Mai 1945 Guelma et toute la promotion 2023 d'informatique.

Dédicace

*pour moi
Ma mère
mon père,
mes soeurs
Mes frères*

Surtout Oussama Akram chouaib, wassim Balil et bien sûr Amir et Mimo pour leurs sacrifices, leur patience, leur amour et leur confiance en moi. Ils l'ont fait Tout pour mon bonheur et ma réussite. Aucune dévotion ne peut exprimer ce que je leur dois ; peut être

Nous demandons à Dieu leur santé, leur bien-être et leur longue vie.

*à
mes amis*

Résumé :

Cette étude présente le développement d'un système de contrôle d'accès pour les péages autoroutiers utilisant la reconnaissance automatique de plaques d'immatriculation (ALPR) intégrée à YOLOv8 pour la détection des plaques et une plateforme PHP pour la gestion des conducteurs. L'objectif est d'automatiser l'identification des véhicules et la gestion de l'accès aux postes de péage, améliorant ainsi l'efficacité et la sécurité.

L'architecture du système inclut un modèle YOLOv8 entraîné pour la détection des plaques en temps réel et lire les chiffres du plaques immatriculation une plateforme PHP pour la gestion des données des conducteurs et des véhicules. La plateforme PHP fournit une interface pour l'inscription des conducteurs et gère l'accès des véhicules en fonction des plaques reconnues, contrôlant ainsi les barrières de péage. Les données sur l'entrée et la sortie des véhicules sont enregistrées et traitées pour contrôler efficacement les barrières de péage.

Les composantes clés de l'étude incluent :

- **Préparation du Dataset** : Collecte et annotation d'images diverses de plaques d'immatriculation pour entraîner le modèle YOLOv8.
- **Entraînement du Modèle** : Entraînement et optimisation du modèle YOLOv8 pour obtenir une reconnaissance des plaques précise et rapide.
- **Développement de la Plateforme PHP** : Création d'une application PHP pour la gestion des données des conducteurs et des véhicules, intégrée aux résultats de détection YOLOv8 pour la gestion du contrôle d'accès.
- **Intégration du Système** : Mise en œuvre d'un module pour contrôler les caméras et les barrières de péage en fonction des résultats de reconnaissance, assurant une réaction dynamique du système à la détection des véhicules en temps réel.

Les résultats démontrent que le modèle YOLOv8 fournit une reconnaissance des plaques d'immatriculation précise et efficace, tandis que la plateforme PHP permet une gestion efficace de l'accès des véhicules. Le système automatise les opérations de péage, réduit les temps d'attente et améliore la sécurité, montrant un potentiel de déploiement à grande échelle dans des contextes de péage autoroutier réels.

Mots-clés : Reconnaissance de Plaques d'Immatriculation, YOLOv8, Plateforme PHP, Contrôle d'Accès au Péage, Gestion des Véhicules.

Abstract

This study presents the development of an access control system for highway tolls using Automatic License Plate Recognition (ALPR) integrated with YOLOv8 for plate detection and a PHP platform for driver management. The goal is to automate vehicle identification and access management at toll booths, thereby improving efficiency and security.

The system architecture includes a YOLOv8 model trained for real-time plate detection and a PHP platform for managing driver and vehicle data. The PHP platform provides an interface for driver registration and manages vehicle access based on recognized plates, thereby controlling the toll barriers. Data on vehicle entry and exit are recorded and processed to effectively manage toll barriers.

Key components of the study include:

- **Dataset Preparation:** Collecting and annotating diverse images of license plates to train the YOLOv8 model.
- **Model Training:** Training and optimizing the YOLOv8 model to achieve accurate and fast plate recognition.
- **PHP Platform Development:** Creating a PHP application for managing driver and vehicle data, integrated with YOLOv8 detection results for access control management.
- **System Integration:** Implementing a module to control cameras and toll barriers based on recognition results, ensuring dynamic system response to real-time vehicle detection.

The results demonstrate that the YOLOv8 model provides accurate and efficient license plate recognition, while the PHP platform enables effective vehicle access management. The system automates toll operations, reduces waiting times, and enhances security, showing potential for large-scale deployment in real-world highway toll contexts.

Keywords: License Plate Recognition, YOLOv8, PHP Platform, Toll Access Control, Vehicle Management.

المخلص:

تقدم هذه الدراسة تطوير نظام للتحكم في الوصول إلى بوابات الطرق السريعة باستخدام التعرف التلقائي على لوحات السيارات (ALPR) المتكامل مع YOLOv8 للكشف عن اللوحات ومنصة PHP لإدارة السائقين. الهدف هو أتمتة التعرف على المركبات وإدارة الوصول إلى محطات الرسوم، مما يحسن الكفاءة والأمان.

تشمل بنية النظام نموذج YOLOv8 مدرب للكشف عن اللوحات في الوقت الحقيقي ومنصة PHP لإدارة بيانات السائقين والمركبات. توفر منصة PHP واجهة لتسجيل السائقين وتدير وصول المركبات بناءً على اللوحات المعترف بها، مما يسيطر على حواجز الرسوم. يتم تسجيل ومعالجة بيانات دخول وخروج المركبات للتحكم الفعال في حواجز الرسوم.

تشمل المكونات الرئيسية للدراسة:

- **تحضير مجموعة البيانات:** جمع ووضع تعليقات على صور متنوعة للوحات السيارات لتدريب نموذج YOLOv8.
- **تدريب النموذج:** تدريب وتحسين نموذج YOLOv8 لتحقيق التعرف الدقيق والسريع على اللوحات.
- **تطوير منصة PHP:** إنشاء تطبيق PHP لإدارة بيانات السائقين والمركبات، متكامل مع نتائج كشف YOLOv8 لإدارة التحكم في الوصول.
- **تكامل النظام:** تنفيذ وحدة للتحكم في الكاميرات وحواجز الرسوم بناءً على نتائج التعرف، لضمان استجابة ديناميكية للنظام للكشف عن المركبات في الوقت الحقيقي.

تظهر النتائج أن نموذج YOLOv8 يوفر تعرّفًا دقيقًا وفعالًا على لوحات السيارات، بينما تمكن منصة PHP من إدارة الوصول للمركبات بشكل فعال. يقوم النظام بأتمتة عمليات الرسوم، ويقلل من أوقات الانتظار، ويعزز الأمان، مما يظهر إمكانيات نشر واسعة النطاق في سياقات الرسوم على الطرق السريعة في العالم الحقيقي.

الكلمات المفتاحية: التعرف على لوحات السيارات، YOLOv8، منصة PHP، التحكم في الوصول للرسوم، إدارة المركبات.

Table des contenus

Contents

Listes des tableaux :	9
Listes des figures :	9
Introduction Générale.....	10
Chapitre 01	12
Introduction, problèmes et solutions basés sur l'intelligence artificielle sur l'autoroute	12
1.1 Introduction :	13
1.2 Définition :	13
1.2.1 Définition de l'Autoroute :.....	13
1.2.2 Définition du Péage d'Autoroute :.....	14
1.2.3 Définition du Matricule :.....	14
1.3 Défis rencontrés pour aider les gens à utiliser l'autoroute :	16
1.3.1 Types de problèmes sur l'autoroute :	16
1.4 Solutions classiques pour aider les gens à utiliser l'autoroute :	19
1.4.1 Avantages et limites de chacune de ces solutions :.....	21
1.5 de l'IA au Deep Learning	21
1.5.1 intelligence Artificiel :	21
1.5.2 Machine Learning :	22
1.5.3 Deep Learning :.....	22
1.6 Domaines d'Application du Deep Learning et de l'Intelligence Artificielle	26
1.7 Exemples des Modèles les Plus Utilisés dans la Reconnaissance des Formes	28
1.8 solutions basées sur l'IA pour aider les gens à utiliser l'autoroute :	30
1.9 Travaux connexes	33
1.7 Conclusion :	33
Chapitre 02	35
Etat de l'art	35
2.1 Introduction	36
2.2 contexte de travail	36
2.3 composants Essentiels d'un Système de Péage Autoroutier	37
2.4 Revue de la littérature sur les systèmes de contrôle d'accès pour les péages autoroutiers :	38
2.4.2 Technologies existantes pour la reconnaissance des plaques d'immatriculation :	38
2.4.3 Avantages et limites des technologies IoT dans les péages autoroutiers :	40

2.4.4 Quelques exemples de paillage dans le monde :	40
2.5 Conclusion.....	42
Chapitre 03	44
Conception	44
3.1 Introduction	45
3.2 objectifs du système	45
3.3 Qu'est-ce qui fait de YOLO un meilleur choix pour la détection d'objets ?	45
3.4 Architecture Globale.....	46
3.4.1 Module d'ouverture la camera	48
3.4.2 Module de détection de véhicule	48
3.4.3 Module de détection de matricule et ROI.....	50
3.4.3.2 L'entraînement du modèle YOLO de détection de matricule.....	55
3.4.4 Module de reconnaissance OCR.....	58
3.4.5 Module de La Vérification avec la Base de Données.....	64
Workflow de la Vérification (Suite)	64
3.4.5.2 Flux de Contrôle du Module de Vérification	65
3.5 Métriques	67
3.6 Conclusion.....	68
Chapitre 4	69
Implémentation and Résultat	69
4.1 Introduction.....	70
4.2 Environnement de développement.....	70
4.3 Training and validation.....	73
4.3.1 résultats de l'entraînement :	73
4.3.2 résultats de l'entraînement de modèle OCR :	78
4.4 Test, Résultat et Discussion	81
4.4.1 Test et Résultat	81
4.4.2 Discussion	82
4.5 Conclusion.....	83
Conclusion Générale	84
Bibliographies	85

Listes des tableaux :

TABLEAU 1.1 AVANTAGES ET LIMITES DE CHACUNE DE CES SOLUTIONS.....	21
TABLEAU 1.2 TRAVAUX CONNEXES	33
TABLEAU 2.1 AVANTAGES ET LIMITES DES TECHNOLOGIES IoT DANS LES PÉAGES AUTOROUTIERS.....	40
TABLEAU 3.1 TABLE DU PARAMÈTRE MODELE DE MATRICULE	55
TABLEAU 3.2 TABLE DU PARAMÈTRE MODÈLE DE OCR	62
TABLEAU 4.1 RÉSULTATS 01	78
TABLEAU 4.2 RÉSULTATS 02	81

Listes des figures :

FIGURE 1.1 MATRICULE ALGÉRIEN SELON LA LOI	16
FIGURE 1.2 LA CONSTRUCTION D'UNE ROUTE PROVOQUE DES EMBOUTEILLAGES	17
FIGURE 1.3 ACCIDENT DE VOITURE SUR UNE AUTOROUTE	17
FIGURE 1.4 AUTOROUTE EST-OUEST : LE DERNIER TRONÇON FINALISÉ AVANT LA FIN DE L'ANNÉE 2022	18
FIGURE 1.5 MÉTÉO ET SÉCURITÉ ROUTIÈRE AWSR.....	18
FIGURE 1.6 CONDUCTEUR EN COLÈRE COINCÉ DANS LA CIRCULATION	19
FIGURE 1.7 LES PANNEAUX DE SIGNALISATION ROUTIÈRE ET LEUR SIGNIFICATION	19
FIGURE 1.8 LIMITATION DE VITESSE - CODE DE LA ROUTE	19
FIGURE 1.9 LES AGENTS DE PATROUILLE ROUTIÈRE AURONT.....	20
FIGURE 1.10 CENTRES D'APPELS ROUTIERS	20
FIGURE 1.11 DIFFÉRENTS SOUS-DOMAINES AI, ML ET DL [48]	23
FIGURE 1.12 CNN ARCHITECTURE.....	25
FIGURE 1.13 SCHÉMA D'UN RÉSEAU DE NEURONES RÉCURRENTS [50].....	26
FIGURE 1.14 LES FONCTION D'UN RÉSEAU ANTAGONISTE GÉNÉRATIF [51].....	26
FIGURE 1.15 SYSTÈMES D'AIDE À LA CONDUITE AVANCÉS	31
FIGURE 1.16 TRAFIC INTELLIGENTE	31
FIGURE 1.17 VÉHICULES AUTONOMES.....	31
FIGURE 1.18 COVOITURAGE.....	32
FIGURE 1.19 PAIEMENT SANS CONTACT.....	32
FIGURE 2.1 LES DIFFÉRENTES ARCHITECTURES DE YOLO	40
FIGURE 3.1 NOTRE IMAGINATION DU SYSTÈME DE PÉAGE INTELLIGENTS.....	45
FIGURE 3.2 ARCHITECTURE GLOBALE DU SYSTÈME.....	47
FIGURE 3.3 CAMERA DANS LA ROUTE.....	48
FIGURE 3.4 DÉTECTION DES OBJETS ET LA RÉTENTION DES VÉHICULES	50
FIGURE 3.5 DÉTECTION DES QUATRE CLASSES	50
FIGURE 3.6 DÉTECTION LE MATRICULE	51
FIGURE 3.7 ARCHITECTURE DE YOLOv8[222]	52
FIGURE 3.8 VISUALISATION D'UNE BOÎTE D'ANCRAGE	53
FIGURE 3.9 NEW YOLOv8 C2F MODULE.....	53
FIGURE 3.10 EXEMPLE DE OCR PAR PYTESSERACT.....	63
FIGURE 3.11 DIAGRAMME DE CLASSE.....	66
FIGURE 4.1 PYTHON LOGO	72
FIGURE 4.2 TRAIN BATCHES	73
FIGURE 4.3 VALIDATION BATCHES.....	74
FIGURE 4.4 RÉSULTAT	74
FIGURE 4.5 F1_CURVE (F1).....	75
FIGURE 4.6 P_CURVE (PRECISION).....	76
FIGURE 4.7 PR_CURVE (PRECISION_RECALL)	76
FIGURE 4.8 R_CURVE (RECALL).....	77
FIGURE 4.9 TRAIN BATCHES	79
FIGURE 4.10 VALIDATION BATCHES	79
FIGURE 4.11 RÉSULTAT OCR.....	80
FIGURE 4.12 R_CURVE (RECALL).....	80

Introduction Générale

La gestion des péages autoroutiers est un aspect crucial pour la fluidité du trafic et la sécurité des infrastructures routières. Cependant, les méthodes traditionnelles de collecte de péages souffrent de plusieurs limitations, notamment les erreurs humaines, les fraudes et les temps d'attente prolongés, ce qui peut entraver l'efficacité globale du système et causer des désagréments aux usagers.

Face à ces défis, l'intégration des technologies avancées telles que l'Internet des objets (IoT) et le Deep Learning offre des perspectives prometteuses pour la modernisation et l'automatisation des systèmes de péage. Ce mémoire vise à élaborer un système de contrôle d'accès sophistiqué pour les péages autoroutiers, exploitant la reconnaissance automatique des plaques d'immatriculation à travers des modèles de Deep Learning en l'occurrence YoloV8, la connectivité en temps réel offerte par l'IoT. L'objectif est de concevoir une solution efficace, évolutive et sécurisée qui non seulement améliore la fluidité du trafic mais renforce également la sécurité.

Pour atteindre cet objectif, nous avons choisi d'utiliser le modèle YOLOv8, reconnu pour ses performances élevées en matière de détection et de reconnaissance d'objets en temps réel. Le modèle sera formé sur des ensembles de données diversifiés et benchmark pour les plaques d'immatriculation et les numéros de matricules, garantissant ainsi une reconnaissance précise et fiable. Parallèlement, les technologies IoT permettront une gestion et une surveillance en temps réel des opérations de péage, assurant une communication fluide et continue entre les différents composants du système.

En combinant ces technologies de pointe, ce projet contribuera de manière significative à l'avancement des systèmes de contrôle d'accès intelligents pour les péages autoroutiers. Il ouvrira la voie à des infrastructures plus performantes et sécurisées, répondant ainsi aux besoins croissants en matière de gestion de trafic et de sécurité routière.

Ce mémoire explore en détail les différentes composantes nécessaires à la mise en œuvre de ce système, en mettant en lumière les avantages et les défis liés à l'utilisation de l'intelligence artificielle et de l'IoT dans ce contexte spécifique. Nous discuterons également des considérations éthiques et pratiques pour garantir une application responsable et efficace de ces technologies innovantes dans les infrastructures de péage autoroutier modernes.

Ce mémoire est composé de quatre chapitres :

Chapitre 01: Introduction, problèmes et solutions basés sur l'intelligence artificielle sur l'autoroute

Ce chapitre commence par une introduction aux concepts d'autoroutes et de péages, suivi d'une définition détaillée des termes clés. Il explore les défis courants que les usagers rencontrent sur les autoroutes, présente des solutions traditionnelles avec leurs avantages et limites, et introduit l'évolution de l'intelligence artificielle (IA) vers le deep learning. Les domaines d'application de ces technologies et des modèles de reconnaissance des formes les plus utilisés sont également discutés, concluant par des solutions basées sur l'IA pour améliorer l'utilisation des autoroutes.

Chapitre 02: État de l'art

Ce chapitre propose une revue de la littérature sur les systèmes de contrôle d'accès pour les péages autoroutiers, incluant les composants essentiels de ces systèmes, les technologies existantes pour la reconnaissance des plaques d'immatriculation, et les avantages et limites des technologies IoT dans

ce contexte. Il inclut des exemples internationaux de péages autoroutiers et conclut en résumant l'état de l'art.

Chapitre 03: Conception

Ce chapitre décrit la conception du système de péage intelligent, mettant en avant les objectifs et les raisons pour lesquelles YOLO est choisi pour la détection d'objets. Il présente l'architecture globale du système, détaillant les modules pour l'ouverture de la caméra, la détection des véhicules, la reconnaissance des plaques, et le module OCR pour la reconnaissance optique des caractères. La section se conclut avec les métriques utilisées pour évaluer le système.

Chapitre 04: Implémentation et Résultats

Ce chapitre se concentre sur l'implémentation pratique du système de péage proposé, décrivant l'environnement de développement, les processus de formation et de validation des modèles. Les résultats de l'entraînement du modèle, notamment le modèle OCR, sont analysés. La section se termine par une discussion sur les résultats des tests et les implications de ces résultats pour l'efficacité du système de péage intelligent.

En termine par une conclusion générale.

Chapitre 01

Introduction, problèmes et solutions basés sur l'intelligence artificielle sur l'autoroute

1.1 Introduction :

Dans ce chapitre, nous examinons les défis complexes liés à la gestion des péages autoroutiers et explorons les solutions novatrices offertes par l'intelligence artificielle (IA) et le deep learning. La collecte des péages est essentielle pour assurer la fluidité du trafic routier, mais elle est souvent confrontée à des problèmes tels que les erreurs de perception, les fraudes et les temps d'attente prolongés. Les méthodes traditionnelles ne parviennent souvent pas à résoudre efficacement ces problèmes, affectant ainsi l'expérience des conducteurs et la gestion globale du trafic.

Face à ces défis, les technologies émergentes comme la reconnaissance automatique de plaques d'immatriculation, alimentée par l'IA, proposent des solutions innovantes. En tirant parti des avancées dans le domaine des réseaux de neurones profonds, ces systèmes peuvent rapidement et précisément identifier les véhicules entrant et sortant des péages, améliorant ainsi l'efficacité de la collecte et la fluidité du trafic. Toutefois, l'adoption de ces technologies soulève également des questions concernant la confidentialité des données et la protection de la vie privée, ce qui nécessite une approche réfléchie et éthique.

Dans ce chapitre, nous analysons en profondeur les défis spécifiques de la gestion des péages autoroutiers, ainsi que les avantages et les limites des solutions basées sur l'IA et le deep learning. Nous aborderons également les considérations éthiques et pratiques liées à l'utilisation de ces technologies et proposerons des recommandations pour une mise en œuvre responsable et efficace dans le contexte des autoroutes modernes.

1.2 Définition :

1.2.1 Définition de l'Autoroute :

Une autoroute [1], également appelée voie express, est une route à chaussées séparées, exclusivement réservée à la circulation automobile à grande vitesse. Ce type de route accueille divers types de véhicules motorisés, tels que les voitures, motos, camions et autobus. L'architecture des autoroutes est spécifiquement conçue pour assurer un trafic fluide et rapide, en évitant les croisements et intersections au niveau du sol. Les entrées et sorties sont régulées par des bretelles d'accès, permettant une transition sécurisée et efficace vers et depuis d'autres routes.

Les autoroutes sont souvent accessibles moyennant un péage, un système de tarification permettant de financer leur construction, leur entretien et leur exploitation. Le péage peut être collecté de différentes manières, notamment par des barrières de péage où les conducteurs doivent s'arrêter pour payer, ou par des systèmes de péage électronique qui permettent une collecte automatique sans interruption du trafic.

En plus d'assurer une circulation rapide et fluide, les autoroutes sont dotées de plusieurs dispositifs de sécurité, tels que des glissières de sécurité, des panneaux de signalisation clairs, et parfois des aires

de repos pour les conducteurs. Les conditions de conduite y sont généralement meilleures que sur les routes secondaires, grâce à un entretien régulier et une surveillance continue.

L'utilisation des autoroutes est réglementée par des lois spécifiques qui interdisent l'accès aux piétons, cyclistes et véhicules lents, afin de maintenir des niveaux de sécurité élevés. En somme, les autoroutes jouent un rôle crucial dans le réseau de transport en facilitant les déplacements rapides et sécurisés sur de longues distances, contribuant ainsi de manière significative à l'efficacité économique et à la mobilité des personnes et des biens.

1.2.2 Définition du Péage d'Autoroute :

Le péage d'autoroute [2] est un système de tarification utilisé pour financer la construction, l'entretien et l'exploitation des autoroutes. Les usagers des autoroutes sont tenus de payer un péage pour pouvoir les emprunter. Ce montant peut varier en fonction de plusieurs facteurs :

1. **Distance Parcourue** : Le coût du péage est souvent proportionnel à la distance parcourue sur l'autoroute. Plus la distance est longue, plus le péage est élevé.
2. **Type de Véhicule** : Les tarifs de péage varient en fonction du type de véhicule. Par exemple, les voitures particulières, les motos, les camions et les autobus peuvent avoir des tarifs différents, les véhicules plus lourds et de plus grande taille étant généralement assujettis à des coûts plus élevés.
3. **Heure de la Journée** : Les tarifs de péage peuvent fluctuer selon l'heure de la journée. Dans certaines régions, les prix sont plus élevés pendant les heures de pointe pour réguler la circulation et inciter les usagers à emprunter les routes en dehors de ces périodes.

Le péage peut être collecté par différents moyens :

- **Barrières de Péage** : Les conducteurs s'arrêtent pour payer le péage soit en espèces, soit par carte bancaire. Certaines barrières acceptent également des cartes prépayées ou des abonnements.
- **Péage Électronique** : Ce système permet de collecter les péages automatiquement sans nécessiter l'arrêt des véhicules. Les conducteurs utilisent des dispositifs électroniques, comme des transpondeurs, qui enregistrent leur passage et débitent automatiquement le montant dû. Cela améliore la fluidité du trafic en supprimant les arrêts aux barrières de péage.

Les fonds récoltés par les péages sont essentiels pour maintenir les autoroutes en bon état, financer leur expansion et garantir la sécurité des usagers. En résumé, le système de péage d'autoroute est une composante cruciale de la gestion des infrastructures routières, assurant leur durabilité et leur efficacité pour les années à venir.

1.2.3 Définition du Matricule :

Le matricule [3], également appelé plaque d'immatriculation, est une plaque métallique fixée à l'avant et à l'arrière d'un véhicule automobile. Elle permet d'identifier de manière unique le véhicule. Le

matricule [4] est composé d'une combinaison de chiffres et de lettres attribuée au véhicule par l'administration compétente.

La plaque d'immatriculation joue un rôle crucial dans la régulation et la gestion du trafic, ainsi que dans l'application des lois relatives aux véhicules et à la circulation routière. Voici quelques points supplémentaires pour étendre l'explication :

1. **Identification Unique** : Chaque véhicule dispose d'une plaque d'immatriculation distincte, ce qui simplifie son identification. Cette singularité permet aux autorités de suivre et d'enregistrer les véhicules pour des besoins de sécurité, de fiscalité et de réglementation.
2. **Composants de la Plaque** : La combinaison de chiffres et de lettres sur une plaque d'immatriculation suit souvent un format spécifique qui peut inclure des indicateurs régionaux ou nationaux, des codes spécifiques pour certaines catégories de véhicules, et parfois des informations sur la date de mise en circulation.
3. **Utilisation pour la Sécurité et l'Application de la Loi** : Les plaques d'immatriculation sont essentielles pour l'application de la loi. Elles permettent aux forces de l'ordre de vérifier l'enregistrement d'un véhicule, d'identifier les véhicules volés, et de repérer ceux qui sont impliqués dans des infractions au code de la route.
4. **Technologie et Reconnaissance Automatique** : En raison des progrès technologiques, les systèmes de reconnaissance automatique des plaques d'immatriculation (LPR) sont de plus en plus couramment employés. Les caméras et les logiciels de traitement d'images sont employés par ces systèmes pour lire les plaques en temps réel, ce qui simplifie la gestion du trafic, la collecte de péages et la surveillance de la sécurité.
5. **Réglementation et Normes** : Le respect de normes spécifiques en matière de taille, de police de caractères et de matériaux est essentiel pour garantir la lisibilité et la durabilité des plaques d'immatriculation. Ces standards diffèrent d'un pays à l'autre.
6. **Changement et Transfert** : Lors de la vente ou du transfert de propriété d'un véhicule, la plaque d'immatriculation peut être transférée au nouveau propriétaire ou remplacée par une nouvelle, selon les réglementations locales.

En résumé, le matricule ou plaque d'immatriculation est un élément indispensable pour l'identification et la gestion des véhicules, jouant un rôle clé dans la sécurité routière, la régulation du trafic et l'application des lois.

1.2.3.1 Le matricule algérien :

Un matricule algérien se présente sous la forme "Numéro de Série – Année + Catégorie - Code de la Wilaya". Cette structure permet d'identifier rapidement l'origine, l'année, et la catégorie du véhicule en question. Voici une description détaillée de chaque composant de la plaque d'immatriculation algérienne :

1. **Numéro de Série** : Il s'agit d'un numéro unique attribué séquentiellement à chaque véhicule lors de son enregistrement. Ce numéro permet de distinguer chaque véhicule individuellement.
2. **Année** : Cette partie de la plaque indique l'année d'enregistrement ou de mise en circulation du véhicule. Elle aide à déterminer l'âge du véhicule.
3. **Catégorie** : Représentée par un code spécifique, cette section de la plaque identifie la catégorie du véhicule, par exemple, voiture particulière, camion, bus, etc.
4. **Code de la Wilaya** : Chaque wilaya (province) en Algérie possède un code numérique spécifique. Ce code indique la région administrative où le véhicule a été enregistré.

La combinaison de ces éléments dans la plaque d'immatriculation algérienne permet une identification précise et complète du véhicule, facilitant la gestion administrative et la régulation du trafic. Voici un exemple pour illustrer la structure : "1234 – 20A - 16".

- **1234** : Numéro de série unique
- **20** : Année d'enregistrement (par exemple, 2020)
- **1** : Catégorie du véhicule (par exemple, voiture particulière)
- **16** : Code de la wilaya (par exemple, Alger)

Cette structure offre une méthode systématique pour enregistrer et suivre les véhicules en Algérie, en fournissant des informations essentielles sur chaque véhicule de manière concise et standardisée.



Figure 1.1 Matricule algérien selon la loi

1.3 Défis rencontrés pour aider les gens à utiliser l'autoroute :

Les usagers de l'autoroute sont confrontés à divers défis qui peuvent affecter leur sécurité, leur confort et leur expérience globale. Ces défis peuvent être classés en plusieurs catégories.

1.3.1 Types de problèmes sur l'autoroute :

Les problèmes rencontrés sur les autoroutes peuvent être classés en plusieurs catégories :

1 Embouteillage routière : L'embouteillage routière est un problème majeur sur les autoroutes, en particulier aux heures de pointe et dans les grandes villes. Cela peut entraîner des retards importants, de la frustration et même des accidents.



Figure 1.2 La construction d'une route provoque des embouteillages

2 Accidents :

Les accidents sont un autre problème majeur sur les autoroutes. Ils peuvent être causés par un certain nombre de facteurs, notamment la vitesse excessive, la conduite distraite, la conduite sous l'influence de l'alcool ou de drogues et les conditions météorologiques défavorables.



Figure 1.3 Accident de voiture sur une autoroute

3 Infrastructure défailante :

L'infrastructure des autoroutes peut se détériorer avec le temps, ce qui peut entraîner des nids-de-poule, des fissures et d'autres dangers. Cela peut augmenter le risque d'accidents et rendre la conduite plus difficile.



Figure 1.4 Autoroute Est-Ouest : le dernier tronçon finalisé avant la fin de l'année 2022

4 Conditions météorologiques défavorables :

La conduite sur autoroute peut être dangereuse en cas de conditions météorologiques défavorables, comme la pluie, la neige et le brouillard. Il est possible de diminuer la visibilité, ce qui rend la réaction aux dangers plus complexe. Il est également possible que les chaussées glissantes accroissent le risque de perte de contrôle du véhicule.



Figure 1.5 Météo et sécurité routière | AWSR

5 Manque d'information :

Les conducteurs peuvent ne pas disposer des informations dont ils ont besoin pour naviguer en toute sécurité et efficacement sur les autoroutes. Cela peut inclure des informations sur les conditions de circulation, les accidents, les fermetures de routes et les travaux routiers.

6 Stress : La conduite sur autoroute peut être stressante, surtout dans des conditions de circulation dense ou lorsque les conditions météorologiques sont défavorables. Cela peut entraîner de la fatigue, de l'agressivité et d'autres problèmes de santé.



Figure 1.6 Conducteur en colère coincé dans la circulation

7 Manque de commodités : Les autoroutes peuvent manquer de commodités, telles que des aires de repos, des toilettes et des restaurants. Cela peut être particulièrement difficile pour les personnes qui voyagent sur de longues distances.

1.4 Solutions classiques pour aider les gens à utiliser l'autoroute :

Avant l'avènement des solutions basées sur l'IA, il existait un certain nombre de solutions classiques pour aider les gens à utiliser l'autoroute, notamment :

1 Signalisation routière : On utilise des panneaux et des marquages au sol afin de tenir les conducteurs au courant des règles de circulation, des dangers et des informations essentielles.



Figure 1.7 Les Panneaux de Signalisation Routière et leur Signification

2 Limites de vitesse : On instaure des limites de vitesse afin d'assurer la sécurité des conducteurs et des piétons.



Figure 1.8 Limitation de vitesse - code de la route

3 Patrouilles routières : Les patrouilles routières appliquent les lois de la circulation et aident les conducteurs en cas de besoin.



Figure 1.9 Les agents de patrouille routière auront

4 Centres d'appels routiers : Les centres d'appels routiers fournissent des informations et une assistance aux conducteurs en cas d'urgence ou de panne.



Figure 1.10 Centres d'appels routiers

Il y a d'autre :

1. **Barrières de Péage Traditionnelles :** Les barrières de péage manuelles, où les conducteurs s'arrêtaient pour payer en espèces ou par carte bancaire, étaient courantes. Cela permettait de réguler l'accès aux autoroutes tout en collectant des fonds pour leur entretien.
2. **Cartes Routières :** Avant l'arrivée des systèmes de navigation GPS, il était essentiel d'utiliser des cartes routières papier pour organiser les itinéraires sur les autoroutes. Elles donnaient des renseignements précis sur les routes, les sorties et les lieux d'attraction.
3. **Aires de Repos et de Service :** Les stations-service, les restaurants et les toilettes étaient des espaces de repos qui permettaient aux conducteurs de se reposer et de se ravitailler.
4. **Radioguidage :** Les stations de radio spécialisées dans les renseignements routiers diffusaient des informations en temps réel sur les conditions de circulation, les travaux routiers et les accidents, ce qui permettait aux conducteurs d'éviter les embouteillages et de prendre des itinéraires alternatifs.
5. **Bornes d'Appel d'Urgence :** Disposées le long des autoroutes, ces bornes permettaient aux conducteurs en panne ou en difficulté de contacter rapidement les services d'assistance et de secours.
6. **Publications et Guides :** Des guides routiers et des publications sur les itinéraires autoroutiers fournissaient des conseils et des informations utiles aux voyageurs, incluant des recommandations sur les meilleures routes, les attractions touristiques et les services disponibles.
7. **Programmes de Réduction de Tarif :** Certaines autoroutes offraient des programmes de réduction ou des abonnements pour les usagers fréquents, permettant de réduire les coûts pour ceux qui utilisaient régulièrement ces routes.

Ces solutions classiques, bien que moins sophistiquées que les technologies modernes basées sur l'IA, ont joué un rôle crucial dans la facilitation de l'utilisation des autoroutes, contribuant à la sécurité et à l'efficacité du réseau routier.

1.4.1 Avantages et limites de chacune de ces solutions :

Solution	Avantages	Limites
Signalisation routière	<ul style="list-style-type: none"> -Relativement peu coûteux à mettre en place et à entretenir. -Facile à comprendre pour la plupart des conducteurs. -Peut être utilisé pour diffuser une large gamme d'informations. 	<ul style="list-style-type: none"> -Peut être submergé par la circulation dense ou des conditions météorologiques défavorables. -Ne peut pas fournir d'informations en temps réel. -Peut ne pas être adapté à toutes les situations. -Limites de vitesse.
Limites de vitesse	<ul style="list-style-type: none"> -Peut aider à réduire le nombre d'accidents et de blessures. -Peut améliorer la fluidité de la circulation. 	<ul style="list-style-type: none"> -Peut être difficile à faire respecter. -Peut ne pas être adapté à toutes les conditions routières. -Peut être perçu comme restrictif par certains conducteurs. -Patrouilles routières
Patrouilles routières	<ul style="list-style-type: none"> -Peut dissuader les comportements de conduite dangereux. -Peut fournir une assistance aux conducteurs en cas de besoin. -Peut appliquer les lois de la circulation. 	<ul style="list-style-type: none"> -Peut être coûteux à maintenir. -Ne peut pas être partout à la fois. -Peut être perçu comme intrusif par certains conducteurs. -Centres d'appels routiers.
Centres d'appels routiers	<ul style="list-style-type: none"> -Peut fournir des informations et une assistance aux conducteurs en cas d'urgence ou de panne. -Peut être utilisé pour signaler des incidents routiers. -Peut offrir un service client. 	<ul style="list-style-type: none"> -Peut être submergé par les appels en cas de conditions météorologiques défavorables ou d'événements majeurs. -Peut ne pas être en mesure de fournir une assistance en temps réel dans tous les cas. -Peut dépendre de la précision des informations fournies par les appelants.

Tableau 1.1 Avantages et limites de chacune de ces solutions

1.5 de l'IA au Deep Learning

1.5.1 intelligence Artificiel :

L'intelligence artificielle (IA) [5] est un domaine de l'informatique qui se concentre sur la création de systèmes capables d'accomplir des tâches nécessitant habituellement un niveau d'intelligence humaine. Ce domaine implique la recherche et le développement d'algorithmes et de modèles permettant aux machines d'apprendre, de prendre des décisions basées sur des données, de raisonner, de planifier et de communiquer en langage naturel. L'IA comprend divers sous-domaines tels que

l'apprentissage automatique, le traitement du langage naturel, la vision par ordinateur, la robotique et les systèmes experts.

En somme, l'IA englobe une vaste gamme de technologies et de méthodes visant à rendre les machines capables de reproduire des fonctions cognitives humaines, facilitant ainsi la résolution de problèmes complexes et l'amélioration de divers aspects de la vie quotidienne et professionnelle.

1.5.2 Machine Learning :

L'apprentissage automatique est une branche de l'intelligence artificielle qui repose sur l'utilisation d'algorithmes statistiques permettant aux systèmes informatiques d'améliorer leurs performances sur des tâches spécifiques au fil du temps, grâce à l'analyse des données, sans nécessiter une programmation explicite. Ce sous-domaine de l'intelligence artificielle se concentre sur le développement d'algorithmes et de modèles statistiques permettant aux systèmes d'apprendre à partir de données et de prendre des décisions ou d'effectuer des prédictions basées sur ces informations.

L'apprentissage automatique peut être catégorisé en plusieurs types :

1. **Apprentissage Supervisé** : Dans cette situation, les algorithmes sont développés en utilisant un ensemble de données étiquetées, c'est-à-dire des données dont les résultats souhaités sont déjà établis. Il s'agit de donner au modèle la capacité de prédire les résultats pour de nouvelles données non étiquetées. La classification d'images, la reconnaissance vocale et la prédiction de valeurs numériques sont des exemples d'applications.
2. **Apprentissage Non Supervisé** : Dans cette situation, les algorithmes sont chargés de repérer des motifs et des liens dans des ensembles de données non séparées. La réduction de dimensionnalité est une méthode d'apprentissage non supervisé qui comprend le clustering (regroupement). La segmentation de clients, la détection d'anomalies et l'exploration de données sont des applications fréquentes.
3. **Apprentissage Semi-Supervisé** : Ce type d'apprentissage utilise à la fois des données étiquetées et non étiquetées pour entraîner les modèles. Il est particulièrement utile lorsque l'étiquetage des données est coûteux ou chronophage, permettant ainsi d'améliorer les performances des modèles avec moins de données étiquetées.

La reconnaissance d'images et vocales, le traitement du langage naturel, la détection de fraude, les systèmes de recommandation et les véhicules autonomes sont autant d'applications de l'apprentissage automatique. [6][7][8]

1.5.3 Deep Learning :

La branche avancée de l'apprentissage automatique appelée apprentissage profond [9] utilise des réseaux de neurones à plusieurs couches afin d'apprendre et de extraire des représentations

hiérarchiques à partir de données. Ces réseaux, couramment désignés sous le nom de réseaux de neurones profonds, sont entraînés sur de grandes quantités de données et ont la capacité de découvrir de manière autonome des caractéristiques et des modèles complexes sans recevoir de directives chiffrées.

L'apprentissage profond [10] a été largement appliqué avec succès dans divers domaines, notamment : **Reconnaissance d'Images, Reconnaissance Vocale, Traitement du Langage Naturel (NLP), Vision par Ordinateur, ...etc.**

Les réseaux de neurones profonds se distinguent par leur capacité à apprendre des représentations de données à différents niveaux d'abstraction. Par exemple, dans la reconnaissance d'images, les premières couches peuvent détecter des bords et des textures, tandis que les couches plus profondes peuvent reconnaître des formes complexes et des objets entiers.

L'entraînement des réseaux de neurones profonds nécessite souvent une puissance de calcul significative et de grandes quantités de données, ce qui a été rendu possible grâce aux avancées en matière de matériel informatique, comme les GPU (unités de traitement graphique) et les TPU (unités de traitement tensoriel), ainsi qu'à l'accès à des ensembles de données massifs.

En résumé, l'apprentissage profond est une technologie puissante et polyvalente de l'intelligence artificielle, permettant aux machines de surpasser les performances humaines dans certaines tâches complexes et ouvrant de nouvelles perspectives dans divers domaines technologiques et scientifiques.

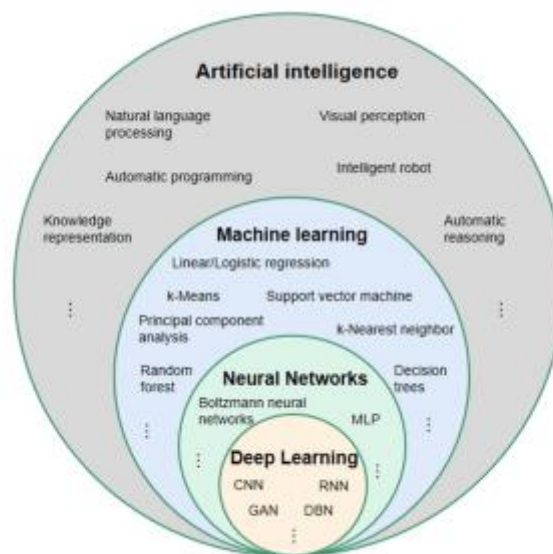


Figure 1.11 Différents sous-domaines AI, ML et DL [48]

1.5.3.1 Terminologie du Deep Learning :

La terminologie du Deep Learning englobe un ensemble d'éléments et de concepts fondamentaux qui sont utilisés pour comprendre et développer des réseaux de neurones artificiels. Voici un aperçu des principaux termes :

1. **Neurone Artificiel** : Un neurone artificiel est une composante essentielle d'un réseau de neurones. Il reçoit une entrée, met en œuvre une fonction d'activation (généralement une fonction non linéaire) et génère une réponse. Tous les neurones sont reliés entre eux par des connexions qui ont des poids particuliers. Ces neurones constituent les éléments essentiels des réseaux de neurones virtuels.
2. **Couches** La structure des réseaux de neurones artificiels est structurée en plusieurs couches. Chaque couche est constituée de neurones qui sont reliés de manière particulière entre eux. Les trois catégories principales de couches sont :
 - **Couche d'entrée** : Cette couche reçoit les données d'entrée du réseau. Par exemple, en traitement d'images, chaque neurone de la couche d'entrée peut recevoir la valeur des pixels de l'image.
 - **Couches cachées** : Ces couches sont responsables des transformations complexes de l'information et ne sont pas directement observables. Elles jouent un rôle crucial dans la modélisation et l'apprentissage des données.
 - **Couche de sortie** : Cette couche fournit la réponse ou les prédictions du réseau, comme les résultats de classification pour des tâches de reconnaissance.
3. **Couche d'Entrée** : Chaque réseau neuronal a une couche d'entrée, qui reçoit les données externes. Par exemple, pour une image, chaque neurone de la couche d'entrée pourrait recevoir l'intensité des pixels dans une certaine région de l'image. Les couches d'entrée sont structurées pour recevoir des données spécifiques au domaine d'apprentissage.
4. **Couches Cachées** : Le nombre et la structure des couches cachées dans un réseau neuronal peuvent varier. Elles jouent un rôle essentiel dans la transformation des données d'entrée en une représentation appropriée pour les tâches de sortie. La complexité des couches cachées permet aux réseaux de capturer des motifs et des relations dans les données de manière non linéaire.
5. **Couche de Sortie** : Cette couche est la dernière étape du réseau, où les données sont transformées en une sortie spécifique. Dans des tâches comme la reconnaissance d'images, la couche de sortie produit une distribution sur plusieurs classes, chacune représentant une classe d'images spécifiques (par exemple, les chiffres de 0 à 9 en reconnaissance de chiffres manuscrits).
6. **Poids** : Les poids sont les éléments qui établissent les liens entre les neurones dans les équations. La force et le sens de la connexion entre les neurones sont modifiés par eux. Au cours de l'entraînement, les poids sont adaptés afin de réduire au minimum l'erreur de sortie du réseau.
7. **Fonctions d'Activation** : Les fonctions d'activation sont des fonctions mathématiques appliquées aux neurones pour introduire de la non-linéarité dans le processus de calcul. Elles

permettent aux réseaux de capturer et de modéliser des relations complexes dans les données. Certaines fonctions couramment utilisées incluent la fonction sigmoïde, la fonction hyperbolique tangentielle (tanh), et la fonction rectifiée (ReLU).

Ces concepts sont essentiels pour comprendre comment les réseaux de neurones fonctionnent et sont utilisés dans divers domaines comme la reconnaissance d'images, le traitement du langage naturel, et plus encore.

1.5.3.2 Sous-Domains du Deep Learning :

Les sous-domaines du Deep Learning couvrent une gamme variée de techniques spécialisées, chacune adaptée à des types spécifiques de données et de problèmes. Voici un résumé des principaux sous-domaines :

1. **Réseaux de Neurones Convolutifs (CNN)** : Les Réseaux de Neurones Convolutifs [11] sont spécifiquement conçus pour le traitement des données structurées en grille, comme les images. Ils utilisent des couches convolutives pour extraire des caractéristiques locales des données, des couches de pooling pour réduire la dimensionnalité, et des couches entièrement connectées pour la classification ou la régression. Les CNN exploitent le partage des poids et l'invariance locale, ce qui les rend efficaces pour des tâches telles que la reconnaissance d'images, la détection d'objets et la segmentation d'images.

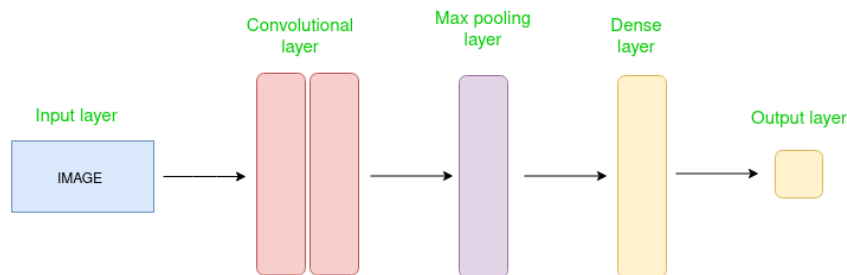


Figure 1.12 CNN architecture

2. **Réseaux de Neurones Récurrents (RNN)** : Les Réseaux de Neurones Récurrents (RNN) [12] sont conçus pour traiter des données séquentielles en intégrant des boucles de rétroaction dans leur architecture. Cela permet aux RNN de maintenir un état ou une mémoire des entrées précédentes, ce qui est essentiel pour modéliser des séquences de données de longueur variable. Les RNN sont largement utilisés dans des applications telles que la modélisation du langage naturel, la reconnaissance vocale, la prédiction de séries temporelles et la traduction automatique.

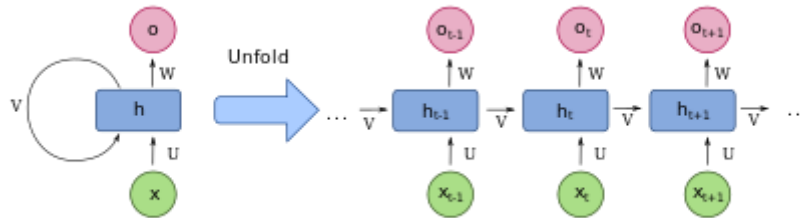


Figure 1.13 Schéma d'un réseau de neurones récurrents [50]

3. **Réseaux Antagonistes Génératifs (GAN)** : Les GAN [13] sont constitués de deux réseaux de neurones qui rivalisent : un générateur et un discriminateur. Le générateur acquiert la capacité de générer des échantillons de données artificielles qui ressemblent à des données réelles, tandis que le discriminateur acquiert la capacité de différencier les données générées des données réelles. En collaboration, ces réseaux sont soumis à des entraînements antagonistes afin d'améliorer la qualité des données synthétiques générées. Les GAN servent à créer des images réalistes, à synthétiser des données et à d'autres applications créatives.

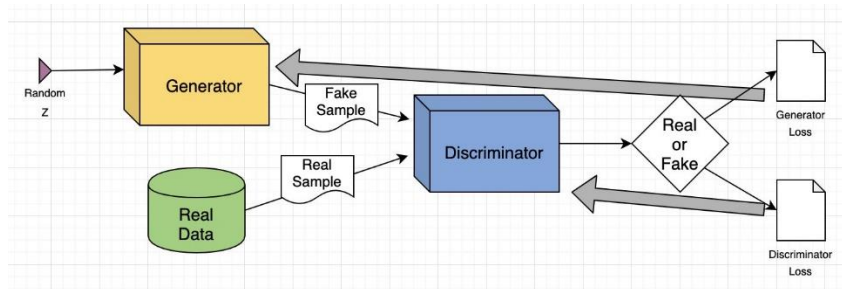


Figure 1.14 Les fonction d'un réseau antagoniste génératif [51]

4. **L'Apprentissage par Renforcement** : L'Apprentissage par Renforcement [52] est une méthode où un agent apprend à prendre des décisions séquentielles pour maximiser une récompense cumulative en interagissant avec un environnement dynamique. L'agent reçoit des récompenses ou des pénalités en fonction de ses actions, et ajuste sa politique d'action pour optimiser ses performances sur le long terme. Cette approche est utilisée dans des domaines comme les jeux informatiques, la robotique et la gestion de portefeuille.

Chaque sous-domaine du Deep Learning a ses propres applications spécifiques et techniques avancées, contribuant à l'évolution rapide des capacités des systèmes d'intelligence artificielle dans divers domaines industriels et scientifiques.

1.6 Domaines d'Application du Deep Learning et de l'Intelligence Artificielle

Les applications du Deep Learning et de l'Intelligence Artificielle (IA) couvrent divers domaines, chacun exploitant les capacités avancées de ces technologies pour traiter efficacement des données complexes. Voici un résumé des principaux domaines d'application :

1. **Reconnaissance Optique de Caractères (OCR)** : En utilisant la reconnaissance optique de caractères [80], il est possible de transformer des images contenant du texte en texte numérique. Pour cette tâche, on utilise fréquemment des modèles de Deep Learning tels que les réseaux neuronaux convolutifs (CNN) et les réseaux de neurones récurrents (RNN). La reconnaissance et l'interprétation efficaces du texte manuscrit ou imprimé sont possibles grâce à ces modèles, ce qui facilite la numérisation et l'indexation de vastes volumes de documents . [55]
2. **Détection d'Objets** : La détection d'objets implique l'identification et la localisation précises d'objets spécifiques dans des images ou des vidéos. Des architectures comme YOLO (You Only Look Once) [53] et Faster R-CNN [54] sont populaires pour cette tâche. Ces modèles permettent de détecter divers objets dans des scènes complexes, utilisant des techniques avancées telles que les réseaux neuronaux convolutifs en combinaison avec des méthodes de détection d'objets spécifiques.
3. **Suivi d'Objets** : Le suivi d'objets consiste à suivre la trajectoire et à identifier des objets spécifiques à travers une séquence d'images ou de vidéos. Des algorithmes comme SORT (Simple Online and Realtime Tracking) [56] et des techniques basées sur les réseaux neuronaux sont utilisés pour cette tâche. Cela trouve des applications dans la surveillance vidéo, la navigation autonome et d'autres domaines où le suivi précis des objets en mouvement est essentiel [57].
4. **Reconnaissance des Formes** : La reconnaissance des formes [81] vise à détecter des configurations ou des motifs spécifiques dans des images. Les réseaux neuronaux convolutifs (CNN) [58] sont couramment utilisés pour cette finalité en raison de leur capacité à extraire des caractéristiques pertinentes à partir d'images complexes. Cette technologie est utilisée dans la reconnaissance faciale, la détection de motifs médicaux, et d'autres applications où la reconnaissance de motifs est cruciale.
5. **Traitement du Langage Naturel (NLP)** : Le traitement du langage naturel (NLP) [60] habilite les machines à comprendre, interpréter et produire du langage humain. Des architectures telles que BERT (Bidirectional Encoder Representations from Transformers) [59] et GPT (Generative Pre-trained Transformer) [59] sont largement employées dans ce domaine.
6. **Chatbots** : Les chatbots [61][62] utilisent l'IA pour simuler des conversations humaines dans un langage naturel. Ils sont utilisés dans divers contextes tels que le service client automatisé, les assistants personnels virtuels et les applications d'interaction utilisateur. Les avancées dans le Deep Learning permettent aux chatbots d'améliorer leur capacité à répondre de manière contextuelle et précise aux requêtes des utilisateurs.

Ces applications démontrent la polyvalence et l'impact significatif du Deep Learning et de l'IA dans des secteurs allant de la reconnaissance visuelle à la compréhension du langage humain, ouvrant de nouvelles perspectives pour l'automatisation intelligente et la prise de décision assistée par ordinateur.

1.7 Exemples des Modèles les Plus Utilisés dans la Reconnaissance des Formes

Voici quelques-uns des modèles les plus utilisés et influents dans le domaine de la reconnaissance des formes, basés sur les architectures de réseaux neuronaux convolutifs (CNN) :

1. **VGG16 et VGG19** : Les architectures VGG [63] se distinguent par leur simplicité et leur profondeur. Elles utilisent des couches convolutives en cascade avec des filtres de petite taille (3x3). VGG16 et VGG19 se différencient principalement par le nombre de ces couches convolutives et de couches entièrement connectées, offrant ainsi une approche robuste pour la reconnaissance d'images.
2. **InceptionV3** : InceptionV3 [64] utilise des modules "Inception" qui combinent des convolutions de différentes tailles pour capturer des caractéristiques à différentes échelles spatiales. Cette architecture optimise l'efficacité des ressources computationnelles et améliore les performances globales du modèle.
3. **ResNet** : ResNet (Residual Network) [65] introduit des "connexions résiduelles" qui facilitent l'entraînement de réseaux très profonds en contrant le problème de dégradation des performances. Cette technique aide à préserver et à transmettre plus efficacement les informations à travers les couches du réseau.
4. **Xception** : Xception [66] améliore le concept d'Inception en utilisant des convolutions séparables en profondeur. Cette approche remplace les convolutions traditionnelles par des opérations plus légères, ce qui augmente l'efficacité et les performances du modèle tout en réduisant le nombre de paramètres.
5. **DenseNet** : DenseNet [67] utilise des connexions denses entre toutes les couches adjacentes, ce qui permet une transmission directe et efficace des caractéristiques à travers le réseau. Cette architecture favorise la réutilisation des caractéristiques et renforce le flux d'informations, améliorant ainsi la performance du modèle.
6. **GoogleNet (InceptionV1)** : GoogleNet [68], également connu sous le nom d'InceptionV1, est une architecture modulaire qui fusionne différents types de convolutions et de pooling au sein de ses modules "Inception". Cette approche modulaire aide à capturer efficacement les caractéristiques complexes dans les images.
7. **MobileNet** : MobileNet [69] est spécifiquement conçu pour être efficace sur des appareils mobiles en utilisant des convolutions séparables en profondeur. Cette architecture réduit la complexité computationnelle tout en maintenant des performances robustes, adaptant ainsi le Deep Learning aux contraintes de ressources des appareils mobiles.

8. **EfficientNet** : EfficientNet [70] utilise une approche d'échelonnage uniforme sur les dimensions de profondeur, de largeur et de résolution du réseau. Cette méthode permet d'optimiser à la fois les performances et l'efficacité computationnelle du modèle, en sélectionnant automatiquement la meilleure combinaison de paramètres.

Ces modèles représentent une avancée significative dans la reconnaissance des formes grâce aux architectures sophistiquées et aux techniques innovantes intégrées dans le domaine du Deep Learning. Ils sont largement utilisés dans des applications telles que la vision par ordinateur, la détection d'objets, et d'autres tâches nécessitant une analyse complexe des données visuelles.

1.8 Datasets :

Quelques-uns des principaux ensembles de données utilisés en vision par ordinateur, en particulier pour la détection d'objets et la classification d'images, sont décrits ci-dessous :

1. **MS COCO dataset** : Le dataset MS COCO (Microsoft Common Objects in Context) [14] est une base de données d'images à grande échelle spécialement conçue pour la reconnaissance d'objets, la segmentation sémantique et la légende d'images. Son contenu comprend plus de 330 000 images et 2,5 millions d'objets annotés répartis dans 80 catégories distinctes. Les notes comprennent des boîtes englobantes pour les objets, des masques de segmentation et des légendes. L'utilisation fréquente de MS COCO pour l'entraînement et l'évaluation des modèles de deep learning est due à la qualité de ses annotations et à la variété de ses situations.
PASCAL VOC dataset : PASCAL VOC (Visual Object Classes) [15] est une autre base de données renommée en vision par ordinateur. Elle comprend environ 17 000 images annotées avec des boîtes englobantes pour 20 catégories d'objets différentes comme des animaux, des véhicules et des articles ménagers. PASCAL VOC a été utilisé pour des tâches de détection d'objets, de segmentation sémantique et de classification d'images. Les ensembles de données d'entraînement et de validation contiennent respectivement environ 11 000 et 5 000 images, avec des annotations détaillées pour chaque instance d'objet.
2. **ImageNet dataset** : ImageNet est une base de données massive contenant plus de 14 millions d'images labellisées, couvrant plus de 21 000 catégories d'objets. Cette base de données a joué un rôle crucial dans le développement des modèles de classification d'images. Chaque image est annotée avec une étiquette décrivant l'objet ou le concept représenté. ImageNet a permis l'entraînement de nombreux modèles de deep learning et a servi de référence pour évaluer les performances des algorithmes sur des tâches de classification d'images.
3. **Open Images dataset** : Open Images [16] est une vaste base de données d'images qui regroupe plus de 9 millions d'images annotées et plus de 15 millions de boîtes englobantes qui couvrent 600 catégories d'objets. Il s'agit d'un des plus exhaustifs ensembles de données pour la détection d'objets et il est subdivisé en sous-ensembles pour l'entraînement, la

validation et les tests. Les annotations comprennent aussi des liens visuels et des concepts visuels précis, ce qui en fait un outil puissant pour la recherche avancée en vision par ordinateur.

4. **KITTI dataset** : La base de données KITTI est largement utilisée pour la recherche en vision par ordinateur appliquée à la conduite autonome. Il inclut différentes scènes urbaines prises par un véhicule muni de capteurs tels que des caméras, des lidars et des radars. Les données sont accompagnées d'informations comme la détection d'objets (véhicules, piétons), la segmentation de la route et la localisation précise des véhicules dans le temps. L'entraînement et la validation des modèles de détection d'objets, de suivi d'objets et de perception de l'environnement sont fréquemment effectués avec KITTI.
5. **OpenALPR dataset** : OpenALPR est une base de données d'images de plaques d'immatriculation qui regroupe une vaste sélection de photographies de plaques d'immatriculation prises dans divers environnements et conditions d'éclairage. Les indications fournies comprennent les localisations précises des plaques d'immatriculation dans les images, ce qui permet de créer des modèles pour détecter et reconnaître les plaques d'immatriculation et les identifier. Les applications de sécurité, de surveillance et de gestion du trafic bénéficient particulièrement de cette base de données.
6. **SVHN dataset (Street View House Numbers)** : Bien que principalement utilisé pour la reconnaissance de chiffres dans les adresses sur les bâtiments, le jeu de données SVHN comprend également des plaques d'immatriculation visibles sur les véhicules capturés par Google Street View. Il contient plus de 600 000 images annotées de chiffres, y compris ceux trouvés sur les plaques d'immatriculation. Bien que ce ne soit pas spécifiquement conçu pour les plaques d'immatriculation, il peut être utilisé pour des tâches de reconnaissance de numéros dans des environnements urbains.

Chaque ensemble de données mentionné ci-dessus a contribué de manière significative au développement et à l'évaluation des modèles de deep learning pour diverses tâches en vision par ordinateur. Leur disponibilité publique a facilité la recherche et l'innovation dans le domaine de l'intelligence artificielle, en permettant aux chercheurs et aux développeurs d'entraîner des modèles sur des données réalistes et diversifiées.

1.8 solutions basées sur l'IA pour aider les gens à utiliser l'autoroute :

L'intelligence artificielle (IA) a le potentiel de révolutionner le transport routier, en rendant la conduite sur autoroute plus sûre, plus efficace et plus agréable pour tous. Voici cinq exemples de solutions basées sur l'IA qui pourraient y contribuer :

1 Systèmes d'aide à la conduite avancés: Les ADAS (Advanced Driver Assistance Systems) utilisent des capteurs et des algorithmes pour aider les conducteurs à rester dans leur voie, à freiner

automatiquement en cas de danger et à éviter les collisions. Ces systèmes peuvent déjà être trouvés sur de nombreux véhicules neufs et devraient devenir encore plus répandus et sophistiqués à l'avenir.



Figure 1.15 Systèmes d'aide à la conduite avancés

2 Gestion du trafic intelligente : Les systèmes de gestion du trafic intelligents utilisent des données en temps réel sur la circulation pour optimiser la signalisation et les feux de circulation, ce qui peut aider à réduire les embouteillages et à améliorer le temps de trajet. Ces systèmes peuvent également être utilisés pour informer les conducteurs des incidents et des travaux routiers, afin qu'ils puissent planifier leur itinéraire en conséquence.

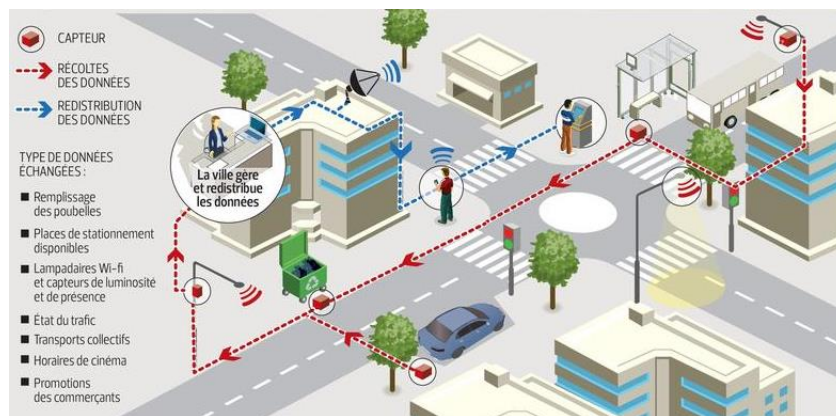


Figure 1.16 Trafic intelligente

3 Véhicules autonomes : Les véhicules autonomes sont capables de naviguer et de se déplacer sans intervention humaine. Ils ont le potentiel de rendre la conduite sur autoroute beaucoup plus sûre et plus efficace, car ils peuvent réagir plus rapidement aux dangers et éviter les erreurs humaines.



Figure 1.17 Véhicules autonomes

4 Covoiturage : Les plateformes de covoiturage basées sur l'IA peuvent mettre en contact des conducteurs qui se rendent au même endroit, ce qui permet de réduire le nombre de véhicules sur la route et de diminuer la congestion. Cela peut également contribuer à réduire la pollution et les émissions de gaz à effet de serre.



Figure 1.18 Covoiturage

5 Paiement de péage sans contact : Les systèmes de péage sans contact utilisent l'IA pour identifier automatiquement les véhicules et facturer leurs conducteurs sans qu'ils aient à s'arrêter. Cela peut rendre le passage des péages plus fluide et plus efficace, en réduisant les embouteillages et les temps d'attente.



Figure 1.19 Paiement sans contact

Ce ne sont que quelques exemples des nombreuses façons dont l'IA peut être utilisée pour améliorer la conduite sur autoroute. À mesure que la technologie continue de se développer, nous pouvons nous attendre à voir apparaître encore plus de solutions innovantes qui rendront la conduite plus sûre, plus efficace et plus agréable pour tous.

1.9 Travaux connexes

Année	Titre	Auteurs	Approche	Capteurs supplémentaires	Dataset	Précision	Références
2023	<i>Système de reconnaissance automatique des plaques d'immatriculation pour le contrôle d'accès aux véhicules</i>	<i>Wei, G., & Zhang, S.</i>	<i>Réseau neuronal convolutif profond</i>	<i>Caméras thermiques</i>	<i>Plaque d'immatriculation LPR</i>	99.7 %	[17]
2022	<i>Reconnaissance des plaques d'immatriculation basée sur l'IA pour le système de contrôle d'accès aux parkings</i>	<i>Mittal, A., & Singh, M. K.</i>	<i>Réseau neuronal convolutif</i>	--	<i>Plaque d'immatriculation LPR</i>		[18]
2021	<i>Système de reconnaissance automatique des plaques d'immatriculation pour le contrôle d'accès aux parkings intelligents</i>	<i>Khan, M. A., et al.</i>	<i>Réseau neuronal convolutif</i>	--	<i>Plaque d'immatriculation LPR</i>		[19]
2020	<i>Reconnaissance des plaques d'immatriculation basée sur la CNN pour le système de contrôle d'accès aux parkings</i>	<i>Li, H., et al.</i>	<i>Réseau neuronal convolutif</i>	--	<i>Plaque d'immatriculation LPR</i>	95.4 %	[20]
2019	<i>Système de reconnaissance des plaques d'immatriculation en temps réel pour le contrôle d'accès aux parkings</i>	<i>Jiang, W., et al.</i>	<i>Réseau neuronal convolutif</i>	--	<i>Plaque d'immatriculation LPR</i>	94.8 %	[21]
2018	<i>Système de reconnaissance automatique des plaques d'immatriculation pour le contrôle d'accès aux parkings</i>	<i>Han, C., et al.</i>	<i>Réseau neuronal convolutif</i>	--	<i>Plaque d'immatriculation LPR</i>	92.7 %	[22]
2017	<i>Reconnaissance des plaques d'immatriculation basée sur SVM pour le système de contrôle d'accès aux parkings</i>	<i>Yu, X., & Wang, Y.</i>	<i>Machine à vecteurs de support</i>	--	<i>Plaque d'immatriculation LPR</i>	96.4 %	

Tableau 1.2 Travaux connexes

1.7 Conclusion :

L'intelligence artificielle (IA) et le Deep Learning ont le potentiel de transformer radicalement le transport routier, en offrant des solutions innovantes qui améliorent la sécurité, l'efficacité et l'expérience globale de conduite sur autoroute. Des avancées telles que les systèmes d'aide à la conduite avancés, les véhicules autonomes, la gestion intelligente du trafic et les plateformes de covoiturage montrent comment l'IA peut répondre à divers défis rencontrés sur les routes.

Les systèmes d'aide à la conduite avancés, par exemple, utilisent des capteurs et des algorithmes pour assister les conducteurs, réduisant ainsi les risques d'accidents grâce à des fonctionnalités telles que le maintien dans la voie et le freinage automatique en cas de danger. Les véhicules autonomes

représentent une avancée majeure en permettant une conduite automatisée, plus sûre et plus efficace, réduisant la dépendance aux erreurs humaines.

La gestion intelligente du trafic utilise les données en temps réel pour optimiser les signaux lumineux et réduire les embouteillages, améliorant ainsi la fluidité du trafic et réduisant les temps de trajet. Les plateformes de covoiturage facilitent la mise en relation des conducteurs partageant un même itinéraire, réduisant ainsi le nombre de véhicules sur les routes et contribuant à la diminution de la pollution.

Bien que ces technologies offrent des avantages significatifs, elles doivent faire face à des défis tels que les coûts, les préoccupations en matière de sécurité et l'acceptation du public. Cependant, les bénéfices potentiels sont considérables : routes plus sûres, circulation plus fluide, réduction de l'empreinte carbone et expérience de conduite améliorée pour tous les usagers.

Alors que nous nous tournons vers un avenir où l'IA façonne le transport routier, il est essentiel de développer et de déployer ces technologies de manière responsable et éthique. Une collaboration étroite entre les secteurs public et privé est cruciale pour exploiter pleinement le potentiel de l'IA et créer un système de transport routier plus intelligent et plus durable, bénéficiant à l'ensemble de la société.

Chapitre 02

Etat de l'art

2.1 Introduction

La détection d'objets est devenue un domaine crucial en vision par ordinateur, offrant une multitude d'applications dans divers secteurs tels que l'automobile autonome, la surveillance et la médecine. Grâce aux progrès de l'apprentissage profond, cette discipline connaît une évolution significative, permettant une détection plus précise et efficace des objets par rapport aux méthodes traditionnelles.

Les modèles de détection d'objets les plus avancés incluent R-CNN, SSD, RetinaNet et YOLO. Chacun de ces modèles utilise des approches différentes pour localiser et classifier les objets dans une image en temps réel. Par exemple, R-CNN introduit la notion de régions d'intérêt et applique un réseau convolutif pour extraire des caractéristiques spécifiques à chaque région. SSD, quant à lui, prédit simultanément les classes d'objets et les boîtes englobantes à partir de plusieurs échelles de convolution. RetinaNet améliore la détection des petits objets en introduisant une perte focalisée, tandis que YOLO adopte une approche "You Only Look Once" pour prédire les boîtes englobantes et les scores de confiance en une seule passe.

La reconnaissance des plaques d'immatriculation est une application spécifique de la détection d'objets qui joue un rôle crucial dans la sécurité routière et d'autres domaines. Cette technologie utilise également des modèles avancés de détection d'objets pour identifier et lire les plaques d'immatriculation dans différentes conditions d'éclairage et d'angle de vue.

Ces avancées ne se limitent pas seulement à la sécurité routière, mais visent également à améliorer l'accessibilité et la qualité de vie des personnes ayant des déficiences visuelles. En permettant aux systèmes autonomes de reconnaître et de comprendre leur environnement, ces technologies ouvrent la voie à une plus grande indépendance et sécurité pour les utilisateurs.

En résumé, la détection d'objets et la reconnaissance des plaques d'immatriculation sont des technologies clés qui bénéficient grandement des avancées en apprentissage profond. Leur développement continue de jouer un rôle essentiel dans l'évolution des solutions d'intelligence artificielle visant à améliorer divers aspects de notre quotidien, de la sécurité routière à l'accessibilité pour tous.

2.2 contexte de travail

Pour répondre à la nécessité d'automatisation des péages autoroutiers en Algérie, notre thèse se concentre sur la conception et l'implémentation d'un système de contrôle d'accès basé sur la reconnaissance des plaques d'immatriculation. Notre architecture globale est structurée en trois étapes principales, chacune avec ses sous-modules spécifiques, assurant une intégration cohérente et complète du système.

Le besoin essentiel de ce projet réside dans l'automatisation et l'amélioration de l'efficacité des péages autoroutiers en Algérie. Actuellement, ces péages peuvent être sujets à des goulots d'étranglement et à des retards dus à la collecte manuelle des frais de péage et à la vérification des documents des conducteurs. En automatisant le processus à l'aide de la reconnaissance des plaques d'immatriculation, on peut réduire les temps d'attente, augmenter la fluidité du trafic et renforcer la sécurité.

Un exemple concret en Algérie pourrait être l'autoroute Est-Ouest, l'une des principales autoroutes du pays reliant les villes d'Est en Ouest sur une longue distance. Cette autoroute est souvent sujette à un trafic dense, surtout aux points d'accès et de sortie où les péages peuvent être particulièrement congestionnés. En introduisant un système

automatisé basé sur la reconnaissance des plaques d'immatriculation, on pourrait faciliter le passage des véhicules tout en assurant un contrôle efficace et sécurisé de l'accès aux péages.

Ainsi, ce projet vise à moderniser l'infrastructure autoroutière en Algérie en intégrant des technologies innovantes pour optimiser la gestion du trafic et améliorer l'expérience des usagers de la route.

2.3 composants Essentiels d'un Système de Péage Autoroutier

Les systèmes de péage autoroutier intègrent une multitude de composants essentiels qui travaillent de concert pour assurer un fonctionnement fluide et sécurisé. Voici une vue d'ensemble des principaux composants :

1. **Gestion des Droits (Toll Management)** : Définit et met à jour les tarifs de péage, applique des réductions ou des majorations, ajuste les prix en temps réel.
2. **Gestion des Comptes (Account Management)** : Administre les comptes des utilisateurs, gère la facturation, les paiements, l'historique des transactions et les réclamations.
3. **Gestion des Utilisateurs (User Management)** : Enregistre les informations personnelles des utilisateurs, gère les identifiants et les authentications, communique via notifications et alertes.
4. **Système d'Intelligence Artificielle (AI System)** : Utilise des algorithmes pour la reconnaissance des plaques d'immatriculation, prédire les flux de trafic et détecter les comportements anormaux.
5. **Capteurs (Sensors)** : Détectent les véhicules avec des capteurs de présence, mesurent le poids avec des capteurs de poids, et capturent la vitesse avec des capteurs de vitesse.
6. **Caméras** : Enregistrent des photos et des vidéos afin d'identifier les véhicules et de surveiller les zones de péage, y compris les caméras ANPR pour les plaques d'immatriculation.
7. **Barrières** : Contrôlent physiquement l'accès aux voies de péage en fonction des autorisations de paiement, synchronisées avec les systèmes de paiement.
8. **Systèmes de Paiement (Payment Systems)** : assurent la gestion des paiements manuels (sur les stations de péage), électroniques (télépéage, cartes bancaires) et automatisés (reconnaissance des plaques d'immatriculation).
9. **Systèmes de Communication (Communication Systems)** : Facilitent l'échange d'informations entre les composants du système et avec les utilisateurs via SMS, e-mail, ou applications mobiles.
10. **Interface Utilisateur (User Interface)** : Fournit des interfaces telles que des applications mobiles, sites web et kiosques de paiement pour interagir avec le système de péage.
11. **Infrastructure Réseau (Network Infrastructure)** : Assure la connectivité sécurisée entre les dispositifs sur le terrain et les centres de gestion, garantissant la transmission sécurisée des données.

12. **Bases de Données (Databases)** : Stockent les informations relatives aux utilisateurs, transactions et configurations du système, gérant les données de facturation et d'historique de trafic.
13. **Systèmes de Sécurité (Security System)** : assurent la protection du système contre les intrusions et les problèmes, assurent la surveillance des accès et des transactions, et garantissent la sécurité des données et des communications.

Ces composants interagissent harmonieusement pour offrir un système de péage autoroutier efficace, améliorant la gestion des péages tout en garantissant une expérience utilisateur optimale et sécurisée.

2.4 Revue de la littérature sur les systèmes de contrôle d'accès pour les péages autoroutiers :

Les systèmes de contrôle d'accès pour les péages autoroutiers ont connu une transformation significative au fil du temps, en réponse aux exigences croissantes en matière de fluidité du trafic et d'efficacité opérationnelle. Initialement, les péages manuels avec des barrières et des collectes en espèces par des opérateurs humains étaient courants, entraînant des retards importants et des coûts élevés en main-d'œuvre [23].

La révolution a débuté avec l'introduction des systèmes de péage électronique (ETC), qui utilisent la technologie RFID pour permettre aux véhicules de passer sans s'arrêter, réduisant ainsi les embouteillages et augmentant le débit de circulation. Parallèlement, la reconnaissance automatique des plaques d'immatriculation (ALPR) a été intégrée pour améliorer la précision et renforcer la sécurité lors de la collecte des péages.

Les systèmes de péage multi-voies libres (MLFF) ont marqué une avancée supplémentaire, permettant aux véhicules de passer sous des portiques à grande vitesse sans ralentir, souvent en combinant RFID et ALPR pour une efficacité maximale. Ces avancées technologiques ont apporté des avantages significatifs en termes de fluidité du trafic et de satisfaction des utilisateurs, comme en témoigne le succès de systèmes tels que l'E-ZPass aux États-Unis [24].

Cependant, l'adoption de ces technologies avancées n'est pas sans défis. Les coûts initiaux élevés pour l'installation et la maintenance de ces systèmes sophistiqués sont une préoccupation majeure. De plus, la protection des données personnelles devient de plus en plus préoccupante, en particulier avec l'utilisation répandue de l'ALPR.

L'évolution des systèmes de péage a permis de répondre efficacement aux besoins croissants des infrastructures routières modernes, en améliorant la fluidité du trafic tout en nécessitant une gestion prudente des coûts et de la confidentialité des données.

2.4.2 Technologies existantes pour la reconnaissance des plaques d'immatriculation :

Dans le domaine de la reconnaissance de plaques d'immatriculation, plusieurs applications du Deep Learning ont été réalisées en utilisant des architectures telles que SSD (Single Shot MultiBox Detector), RetinaNet et YOLO (You Only Look Once), combinées avec la technologie OCR (Reconnaissance Optique de Caractères) et RFID (Identification par Radiofréquence) :

SSD (Single Shot MultiBox Detector) : SSD est un modèle de détection d'objets qui se distingue par sa capacité à détecter rapidement et avec précision les plaques d'immatriculation dans des images complexes. En segmentant l'image en plusieurs régions et en prédisant simultanément les boîtes de délimitation ainsi que les classes d'objets, SSD offre une solution efficace pour la reconnaissance des plaques d'immatriculation [29].

RetinaNet : RetinaNet a été largement adopté pour résoudre le problème du déséquilibre de classe dans la détection de plaques d'immatriculation. En utilisant une combinaison de réseaux neuronaux convolutifs (CNN) et de mécanismes de focalisation, RetinaNet parvient à détecter les plaques d'immatriculation avec une grande précision, même dans des conditions difficiles où les plaques peuvent être partiellement obscurcies ou déformées [30].

R-CNN (Regions with Convolutional Neural Networks) : R-CNN génère des régions candidates via des algorithmes de recherche sélective, puis applique des réseaux de neurones convolutionnels pour extraire les caractéristiques et classifier les objets [72].

Fast R-CNN : Fast R-CNN optimise R-CNN en fusionnant les étapes d'extraction des caractéristiques et de classification en un seul réseau, tout en adoptant une méthode d'apprentissage de bout en bout [73].

Faster R-CNN : Faster R-CNN intègre des réseaux de propositions de régions (RPN) pour générer les propositions de régions directement dans le réseau. Cela permet de réduire le temps de calcul et d'améliorer la précision [74].

Mask R-CNN : Mask R-CNN améliore Faster R-CNN en incorporant une branche supplémentaire dédiée à la segmentation des objets. Cette branche génère des masques binaires pour chaque objet détecté, en plus des boîtes englobantes et des classifications [75].

EfficientDet : EfficientDet utilise l'architecture EfficientNet comme backbone et intègre une nouvelle méthode de fusion de caractéristiques appelée "BiFPN" (Réseau de Pyramide de Caractéristiques Bidirectionnelle). Il applique également un échelonnement uniforme aux différentes composantes du réseau, permettant ainsi une détection efficace des objets avec une meilleure utilisation des ressources computationnelles [76].

OCR (Reconnaissance Optique de Caractères) : La technologie OCR est essentielle pour convertir les caractères d'une plaque d'immatriculation capturée visuellement en texte numérique. Une fois la plaque détectée par un système de vision par ordinateur, l'OCR analyse l'image pour identifier les caractères alphanumériques, facilitant ainsi leur lecture automatique dans les systèmes de péage autoroutier [32].

RFID (Identification par Radiofréquence) : La technologie RFID permet d'identifier les véhicules à l'aide de tags RFID placés sur le pare-brise ou les plaques d'immatriculation. Ces tags émettent un signal radiofréquence contenant des informations d'identification uniques pour chaque véhicule. Les lecteurs RFID installés aux péages autoroutiers peuvent lire ces informations et valider l'identité du véhicule, permettant un péage automatique sans interaction directe avec le conducteur [33].

YOLO (You Only Look Once) : YOLO est célèbre pour sa rapidité et son efficacité dans la détection d'objets en temps réel, y compris les plaques d'immatriculation. En analysant l'image globalement en une seule passe, YOLO permet une détection rapide et précise des plaques, ce qui le rend particulièrement adapté pour les applications nécessitant une réponse en temps réel comme le contrôle du trafic et les systèmes de péage autoroutier [31].

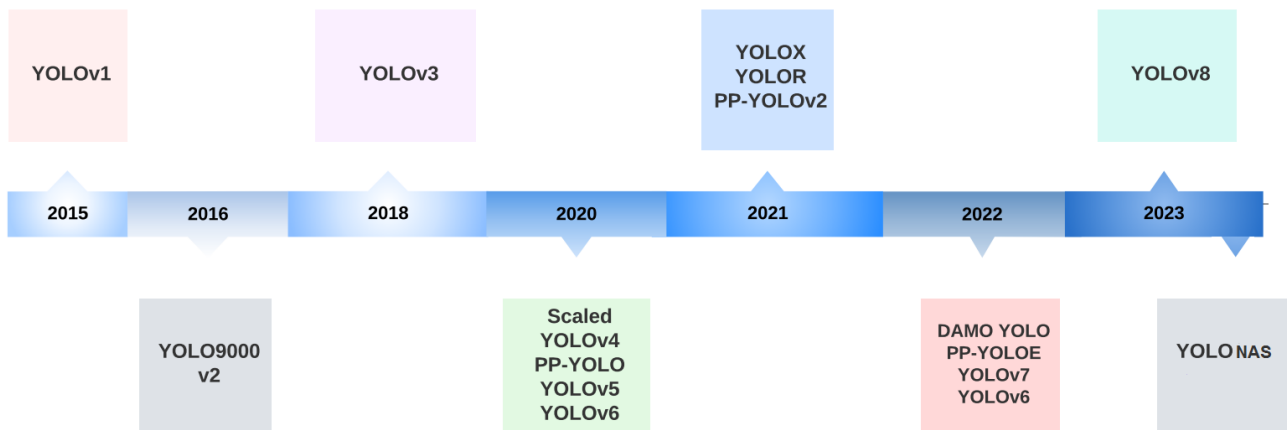


Figure 2.1 les différentes architectures de YOLO

Ces modèles représentent divers compromis en termes de complexité, précision, et performance en temps réel, et sont largement utilisés dans des applications allant de la surveillance et de la conduite autonome à l'analyse de vidéos en temps réel et aux systèmes de sécurité.

2.4.3 Avantages et limites des technologies IoT dans les péages autoroutiers :

Bien que les technologies IoT offrent de nombreux avantages pour les péages autoroutiers en termes d'automatisation, d'efficacité opérationnelle et d'amélioration de l'expérience utilisateur, elles présentent également des défis significatifs en termes de coûts, de sécurité, d'intégration et de gestion des données. Une mise en œuvre réussie nécessite une planification minutieuse et une gestion proactive des risques associés.

Voici un tableau résumant les avantages et les limites des technologies IoT dans les péages autoroutiers, Ce tableau récapitule comment les technologies IoT peuvent améliorer les péages autoroutiers tout en mettant en lumière les défis potentiels associés à leur mise en œuvre et à leur exploitation. :

Avantages	Limites
Automatisation et efficacité accrue	Investissement initial élevé
Collecte de données en temps réel	Dépendance à la connectivité
Optimisation des voies de péage	Préoccupations en matière de sécurité des données
Maintenance proactive	Intégration complexe
Amélioration de la sécurité	Besoin de compétences techniques
Réduction des coûts opérationnels	Évolution rapide de la technologie
Expérience utilisateur améliorée	Gestion des données volumineuses

Tableau 2.1 Avantages et limites des technologies IoT dans les péages autoroutiers

2.4.4 Quelques exemples de paillage dans le monde :

Ces exemples montrent comment les technologies IoT et d'autres innovations sont utilisées pour moderniser les systèmes de péage à travers le monde, améliorant ainsi l'efficacité opérationnelle et l'expérience des utilisateurs sur les routes.

Péage électronique (e-Tolls) en Australie

Le système de péage électronique en Australie est largement adopté pour automatiser la collecte des péages en temps réel. Ce système utilise principalement la technologie RFID (Radio-Frequency Identification) et des caméras de surveillance. Les véhicules équipés de tags RFID peuvent passer sous des portiques spécifiques sans s'arrêter, permettant une gestion efficace du trafic sur les

autoroutes. Les caméras de surveillance capturent également les plaques d'immatriculation pour des véhicules non équipés de RFID, assurant ainsi une couverture complète. Ce système réduit significativement les temps d'attente aux guichets traditionnels de péage, améliorant ainsi la fluidité du trafic et l'expérience des conducteurs.

Péage urbain à Singapour

Singapour a mis en œuvre un système de péage urbain innovant basé sur les technologies IoT. Ce système utilise des caméras de surveillance haute technologie et des capteurs de trafic pour surveiller et réguler la circulation dans la ville. Les caméras capturent les plaques d'immatriculation des véhicules entrant dans certaines zones urbaines, tandis que les capteurs mesurent le flux de circulation en temps réel. En combinant ces données, Singapour peut appliquer des péages variables en fonction des heures de pointe, des zones congestionnées ou des objectifs de gestion du trafic urbain. Cela contribue à réduire les embouteillages, à améliorer la qualité de l'air urbain et à optimiser l'utilisation des infrastructures routières.

Systèmes de péage intelligents aux États-Unis

Aux États-Unis, de nombreux États ont adopté des systèmes de péage intelligents pour simplifier la collecte des péages sur les autoroutes et les ponts à péage. Ces systèmes utilisent des technologies avancées telles que des capteurs de péage et des dispositifs de communication sans fil pour identifier les véhicules et prélever les péages de manière automatisée. Les conducteurs peuvent choisir entre plusieurs méthodes de paiement, y compris le télépéage (ou EZ Pass), les cartes de crédit liées à des comptes prépayés, ou même des applications mobiles spécifiques. Cette automatisation réduit les temps d'attente et les congestions aux guichets de péage traditionnels, améliorant ainsi la fluidité du trafic et l'efficacité des déplacements.

Péage interurbain en Europe

En Europe, de nombreux pays ont mis en place des systèmes de péage interurbains utilisant des technologies IoT avancées. Ces systèmes sont généralement déployés sur les autoroutes et les routes à péage, où les véhicules sont identifiés à l'aide de capteurs de péage et de technologies de communication sans fil. Les capteurs détectent les véhicules et les péages sont prélevés automatiquement en fonction de la distance parcourue ou d'autres critères spécifiques. L'objectif principal de ces systèmes est de gérer efficacement le trafic, de répartir équitablement les coûts d'entretien des infrastructures routières et d'améliorer la sécurité sur les routes principales à travers l'Europe.

Ces exemples illustrent comment les technologies de pointe telles que l'IoT, les RFID, les caméras de surveillance et les capteurs sont utilisées pour moderniser les systèmes de péage à travers le monde, contribuant ainsi à une gestion plus efficace du trafic, à une réduction des temps d'attente et à une meilleure expérience utilisateur sur les routes.

Autoroute Est-Ouest en Algérie : Points de Péage en Construction

L'autoroute Est-Ouest est l'une des plus grandes infrastructures routières en Algérie, traversant le pays sur environ 1 216 km d'est en ouest, reliant les frontières tunisienne et marocaine. Elle passe par plusieurs grandes villes, dont Alger, Oran, Constantine, et Annaba. Ce projet vise à désengorger les routes nationales, améliorer le transport de marchandises, et réduire le temps de trajet entre les principales villes du pays. Voici un aperçu des points de péage en construction le long de cette autoroute :

Points de Péage sur l'Autoroute Est-Ouest en Algérie :

1. Point de Péage de Lakhdaria [90]

- **Localisation :** Wilaya de Bouira

- **Fonction** : Gère le trafic en provenance d'Alger vers l'Est.
- **Objectif** : Organiser le flux de véhicules et assurer la collecte des péages pour financer la maintenance de l'autoroute.

2. Point de Péage d'El Harrach [91]

- **Localisation** : Près d'Alger, capitale de l'Algérie
- **Fonction** : Contrôle le trafic sortant d'Alger vers l'Est et l'Ouest.
- **Objectif** : Faciliter la gestion du trafic aux abords de la capitale et collecter les péages pour financer les infrastructures.

3. Point de Péage de Chlef [92]

- **Localisation** : Wilaya de Chlef
- **Fonction** : Gère le flux de véhicules entre l'Ouest et le centre du pays.
- **Objectif** : Faciliter les déplacements sur l'autoroute en collectant des péages pour l'entretien et le développement de l'infrastructure.

4. Point de Péage de Constantine [93]

- **Localisation** : Près de la ville de Constantine
- **Fonction** : Gère le trafic vers l'Est, particulièrement entre Constantine et Annaba.
- **Objectif** : Assurer un transit efficace et collecter les péages pour soutenir les projets d'infrastructure dans la région.

5. Point de Péage d'Oran [94]

- **Localisation** : À l'entrée de la ville d'Oran
- **Fonction** : Facilite la gestion du trafic de l'Ouest vers le centre du pays.
- **Objectif** : Améliorer la fluidité du trafic et collecter les péages pour financer les projets autoroutiers dans la région.

Ces points de péage sont stratégiquement situés le long de l'autoroute Est-Ouest pour réguler le flux de véhicules, assurer la sécurité routière et financer la maintenance continue de cette infrastructure vitale pour l'Algérie.

2.5 Conclusion

Ce chapitre souligne l'importance croissante de l'automatisation des péages autoroutiers, catalysée par les avancées technologiques dans la reconnaissance des plaques d'immatriculation. Ces technologies représentent une évolution significative par rapport aux méthodes traditionnelles, offrant la promesse d'améliorer non seulement l'efficacité opérationnelle mais aussi la sécurité sur les autoroutes. En explorant divers modèles de détection d'objets comme SSD, RetinaNet, YOLO, ainsi que les technologies OCR et RFID, nous avons identifié leurs capacités à automatiser la collecte des péages, réduire les temps d'attente et optimiser la gestion du trafic.

Les avantages de ces systèmes incluent une collecte automatisée des péages en temps réel, une réduction des embouteillages grâce à des voies de péage optimisées, une meilleure sécurité routière

via la détection des plaques d'immatriculation et des alertes en cas d'incidents, et une amélioration globale de l'expérience utilisateur. Cependant, ces progrès ne sont pas sans défis : les coûts initiaux élevés pour la mise en place de ces technologies avancées, les préoccupations croissantes concernant la confidentialité des données personnelles collectées, ainsi que les exigences en termes de maintenance et d'interopérabilité des systèmes existants constituent des défis importants à surmonter.

Dans les chapitres suivants, nous approfondirons notre exploration en proposant une réflexion sur la manière de relever ces défis tout en maximisant les avantages des technologies modernes pour les péages autoroutiers. Notre objectif est de développer une solution innovante qui combine efficacité opérationnelle, sécurité accrue et respect des normes de confidentialité, afin de répondre aux besoins croissants des infrastructures routières modernes.

Chapitre 03

Conception

3.1 Introduction

Dans ce projet, nous proposons une approche avancée pour la détection et la reconnaissance des plaques d'immatriculation en utilisant des techniques de Deep Learning. Notre choix s'est porté sur l'algorithme YOLO (You Only Look Once) pour la détection des plaques d'immatriculation dans les images, complété par la fonction pytesseract pour la reconnaissance des caractères. Ce chapitre initial se concentre sur les objectifs fondamentaux de notre système, son architecture globale, les différentes étapes de conception et les algorithmes spécifiques que nous avons adoptés.

L'objectif principal de ce projet est de parvenir à une détection robuste et efficace des plaques d'immatriculation à partir d'images variées, et à une reconnaissance précise des caractères qui composent ces plaques. En exploitant les capacités du Deep Learning, nous visons à surpasser les approches traditionnelles en matière de précision et de vitesse de traitement. Cette combinaison de technologies permettra non seulement d'améliorer la gestion des infrastructures routières, mais aussi d'ouvrir la voie à de nouvelles applications dans le domaine de la sécurité routière et de la gestion du trafic.

Au cours des prochaines sections, nous détaillerons les phases de développement, les choix architecturaux spécifiques, ainsi que les résultats obtenus à travers l'implémentation de notre système. Nous explorerons également les défis rencontrés et les solutions envisagées pour optimiser les performances et assurer la fiabilité du système dans des conditions réelles.

3.2 objectifs du système

Dans le cadre de ce projet, notre objectif principal est de concevoir et mettre en œuvre un système avancé de contrôle d'accès pour les péages autoroutiers en exploitant la reconnaissance des plaques d'immatriculation. Ce système sera capable de détecter de manière efficace les plaques d'immatriculation dans les images capturées, puis d'extraire précisément le contenu de ces plaques, notamment les symboles et les caractères alphanumériques.

Le but ultime est de permettre l'ouverture automatisée des barrières de péage en fonction des informations extraites des plaques d'immatriculation. Cela implique non seulement la détection fiable des véhicules et de leurs plaques, mais aussi la reconnaissance exacte des informations nécessaires pour autoriser le passage.



Figure 3.1 Notre imagination du Système de péage intelligents

3.3 Qu'est-ce qui fait de YOLO un meilleur choix pour la détection d'objets ?

YOLO (You Only Look Once) est un modèle de détection d'objets largement adopté en raison de plusieurs caractéristiques clés qui le distinguent des autres approches :

1. **Détection en une seule étape** : YOLO est un modèle de détection en une seule étape, ce qui signifie qu'il prédit directement les boîtes englobantes et les classes des objets dans une seule passe du réseau neuronal. Contrairement aux modèles en deux étapes, YOLO ne nécessite pas de propositions de régions préalables, ce qui le rend beaucoup plus rapide.
2. **Vitesse élevée** : Conçu pour le temps réel, YOLO est optimisé pour des performances rapides sur des images et des vidéos. Il divise l'image en une grille de cellules et prédit les boîtes englobantes et les probabilités de classe pour chaque cellule. Cette approche par grille permet à YOLO de fonctionner à des cadences élevées, ce qui est idéal pour des applications telles que la surveillance vidéo et la conduite autonome.
3. **Simplicité d'architecture** : YOLO a une architecture relativement simple par rapport à d'autres modèles de détection d'objets. Il se compose principalement de couches convolutionnelles suivies de couches entièrement connectées pour prédire les boîtes englobantes et les classes. Sa simplicité facilite non seulement l'entraînement mais aussi le déploiement sur différents types de matériel.
4. **Prédictions multi-échelles** : YOLO effectue des prédictions à différentes échelles spatiales en utilisant des cartes de caractéristiques de différentes profondeurs dans le réseau. Cela lui permet de détecter efficacement des objets de différentes tailles dans une image, améliorant ainsi sa capacité à gérer des scènes avec des objets variés.
5. **Haute précision** : Bien que YOLO ne soit **pas** toujours le modèle le plus précis par rapport aux dernières architectures comme EfficientDet ou Cascade R-CNN, il maintient un équilibre solide entre vitesse et précision. Les différentes **versions** de YOLO continuent d'introduire des améliorations qui augmentent sa performance sur divers jeux de données.
6. **Score d'objectivité** : YOLO introduit le concept de "score d'objectivité", qui évalue la probabilité qu'une boîte englobante contienne un objet réel. Ce score aide à filtrer les détections de faible confiance, améliorant ainsi la qualité globale des détections.
7. **Optimisation conjointe de la localisation et de la classification** : YOLO optimise une fonction de perte conjointe qui prend en compte à la fois la précision de localisation des boîtes englobantes et la précision de la classification des objets. Cette approche intégrée permet à YOLO de traiter la détection d'objets comme un problème unifié, même si elle peut ne pas être aussi spécialisée que les méthodes en deux étapes.

En conclusion, YOLO se distingue par sa rapidité, sa simplicité d'architecture, sa capacité à gérer des prédictions multi-échelles et son équilibre entre vitesse et précision. Ces caractéristiques en font un choix idéal pour de nombreuses applications nécessitant une détection d'objets en temps réel efficace et fiable.

3.4 Architecture Globale

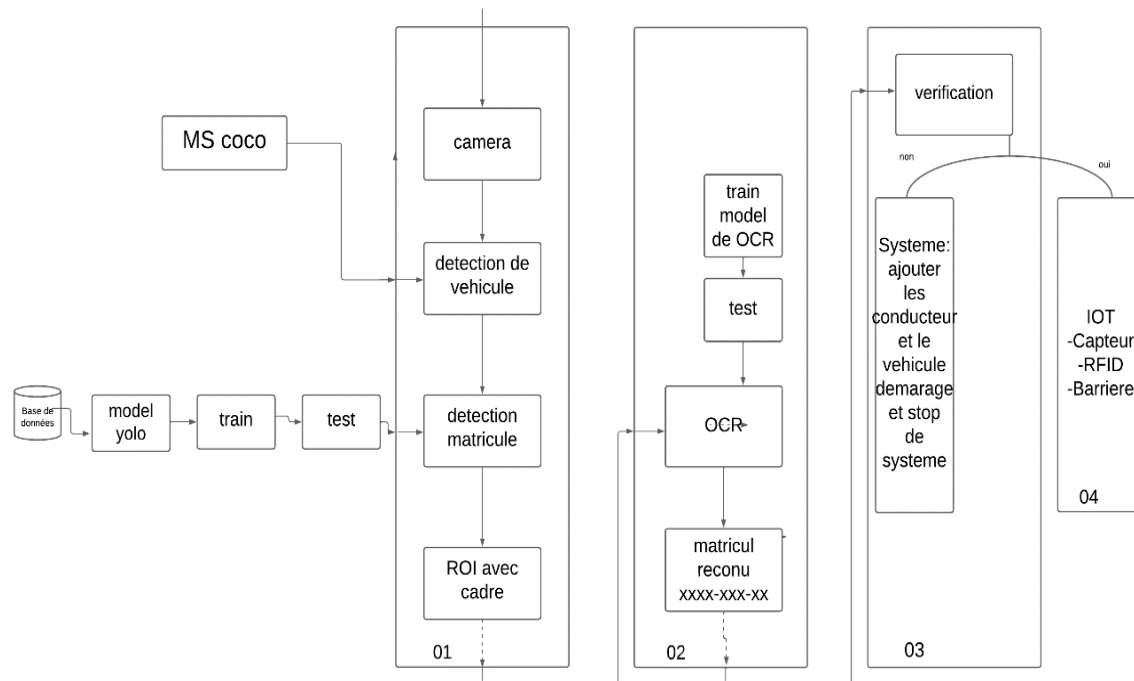


Figure 3.2 architecture globale du système

Notre structure globale est subdivisée en trois sections de base pour chaque étape, chacune ayant ses propres sous-modules qui complètent l'autre. En résumé, il s'agit de l'identification de l'image et de sa subdivision en véhicules et plaques numérotées. La deuxième étape consiste à lire la plaque d'immatriculation entièrement. La troisième étape consiste en une combinaison d'ouverture automatique de la barrière et de préparation de l'étage pour les individus non enregistrés, bien sûr après avoir comparé la plaque d'immatriculation avec les plaques enregistrées dans la base de données.

Nous avons organisé notre architecture globale en trois sections principales avec leurs sous-modules respectifs, voici comment chaque étape et ses composants sont organisés dans notre projet de reconnaissance de plaques d'immatriculation pour les péages autoroutiers :

1. Première étape : Identification et détection des plaques d'immatriculation

Modules sous-jacents :

- **Détection d'objets** : Utilisation de YOLO pour localiser les véhicules et les plaques d'immatriculation dans les images.
- **Segmentation d'image** : Extraction de la région contenant la plaque d'immatriculation à partir de l'image complète du véhicule.
- **Prétraitement d'image** : Normalisation et amélioration de la qualité de l'image pour une meilleure reconnaissance.

2. Deuxième étape : Reconnaissance optique de caractères (OCR)

Modules sous-jacents :

- **Extraction de caractères** : Identification des caractères de la plaque d'immatriculation détectée en utilisant des bibliothèques telles que Pytesseract.
- **Post-traitement** : Correction et validation des caractères reconnus pour assurer une lecture précise.

3. Troisième étape : Contrôle d'accès et gestion des barrières

Modules sous-jacents :

- **Comparaison avec la base de données** : Contrôle de la plaque d'immatriculation reconnue par rapport aux informations enregistrées dans la base de données des utilisateurs autorisés.
- **Décision d'accès** : Automatisation de l'ouverture de la barrière si la plaque est autorisée ou envoi d'une alerte pour les plaques non autorisées.

- **Intégration système :** Liaison avec le système de contrôle des barrières et de gestion du trafic pour une ouverture et une fermeture synchronisée.

La partie suivante est une explication détaillée de tous les modules de notre système :

3.4.1 Module d'ouverture la camera

Dans notre système de reconnaissance de plaques d'immatriculation pour les péages autoroutiers, la caméra joue un rôle crucial en tant que lien direct entre le système informatique et l'environnement extérieur. Voici comment nous avons conçu et intégré ce module dans notre architecture :

Importance de la caméra

La caméra est essentielle car elle capture les images des véhicules entrant dans la zone de péage. Sa performance et son efficacité impactent directement le fonctionnement global du système de reconnaissance des plaques d'immatriculation. Nous avons sélectionné des caméras de haute qualité parmi une base de données pour garantir des résultats optimaux.

Caractéristiques clés du module :

1. **Choix de la caméra :** Nous avons soigneusement sélectionné des caméras performantes adaptées à notre environnement spécifique. Ces caméras sont capables de capturer des images de haute résolution avec une grande clarté, ce qui est essentiel pour une détection précise des plaques d'immatriculation.
2. **Flexibilité multi-caméras :** Notre système est conçu pour fonctionner avec plusieurs caméras simultanément. Cela permet de couvrir efficacement toutes les voies de péage et de gérer un volume élevé de trafic sans compromettre la vitesse ou la précision de la détection.
3. **Intégration transparente :** Les caméras sont intégrées de manière transparente dans l'infrastructure du système de péage autoroutier. Elles sont positionnées stratégiquement pour capturer les images des véhicules entrants de manière optimale.
4. **Performance optimisée :** Pour assurer une performance optimale, les caméras sont équipées de fonctionnalités avancées telles que la mise au point automatique, la compensation de lumière et la résistance aux conditions météorologiques adverses.
5. **Connectivité réseau :** Toutes les caméras sont reliées au réseau informatique du système, garantissant ainsi une transmission rapide et sécurisée des images capturées vers les modules de traitement des données et de reconnaissance des plaques en circulation.

En intégrant soigneusement ce module de caméra dans notre architecture globale, nous optimisons la collecte initiale de données visuelles pour le processus de reconnaissance des plaques d'immatriculation, facilitant ainsi une gestion efficace du contrôle d'accès aux péages autoroutiers.



Figure 3.3 Camera dans la route

3.4.2 Module de détection de véhicule

Le module de détection de véhicule joue un rôle essentiel dans notre système de reconnaissance des plaques d'immatriculation pour les péages autoroutiers, car il permet d'identifier et de classer les différents types de véhicules qui pénètrent dans la zone de péage. Voici la méthode utilisée pour mettre en place ce module en utilisant l'algorithme.

YOLO (You Only Look Once) :

Importance de la détection de véhicule

La capacité à reconnaître précisément les types de véhicules est essentielle pour le bon fonctionnement du système de péage autoroutier. Notre objectif principal est d'identifier quatre catégories spécifiques de véhicules : vélo, moto, voiture et camion. Chaque catégorie est associée à un numéro unique de classe d'objet dans le modèle YOLO, facilitant ainsi la reconnaissance et la classification automatiques.

Fonctionnalités clés du module :

1. **Utilisation de YOLO** : Nous avons choisi YOLO en raison de sa capacité prouvée à détecter et classer efficacement les objets en temps réel, y compris les véhicules. YOLO est pré-entraîné pour reconnaître spécifiquement les quatre catégories de véhicules avec une grande précision.
2. **Catégories de véhicules** :
 - **Classe 2 : Vélo**
 - **Classe 3 : Moto**
 - **Classe 5 : Voiture**
 - **Classe 7 : Camion**

YOLO attribue un score de confiance à chaque détection, indiquant la probabilité que l'objet détecté appartienne à l'une de ces catégories. Cette information est essentielle pour la suite du processus de reconnaissance des plaques d'immatriculation.

3. **Mesure précise des dimensions** : Outre la classification, YOLO est capable de mesurer avec précision les dimensions des véhicules détectés. Cette fonctionnalité est importante pour évaluer l'espace occupé par chaque véhicule dans la voie de péage, aidant ainsi à optimiser le placement et l'exploitation des barrières de péage.
4. **Performance en temps réel** : YOLO est optimisé pour fonctionner en temps réel, permettant une détection rapide des véhicules même à des vitesses élevées. Cette caractéristique est cruciale pour les environnements de péage autoroutier où la fluidité du trafic est primordiale.

Avantages du module :

- **Précision élevée** : YOLO offre une précision élevée dans la détection et la classification des véhicules grâce à son architecture robuste et à son entraînement sur des jeux de données diversifiés.
- **Vitesse de traitement** : En étant un modèle en une seule étape, YOLO garantit des performances rapides sans compromettre la précision, ce qui est essentiel pour la gestion efficace des péages autoroutiers.
- **Adaptabilité** : Le module est conçu pour être adaptable à différents types de véhicules, ce qui permet une application universelle dans divers scénarios de péage autoroutier sans nécessiter de modifications majeures.

En intégrant efficacement le module de détection de véhicule basé sur YOLO dans notre architecture globale, nous optimisons la capacité du système à identifier rapidement et avec précision les véhicules entrant dans la zone de péage, posant ainsi les bases pour une reconnaissance efficace des plaques d'immatriculation dans les étapes suivantes du processus.

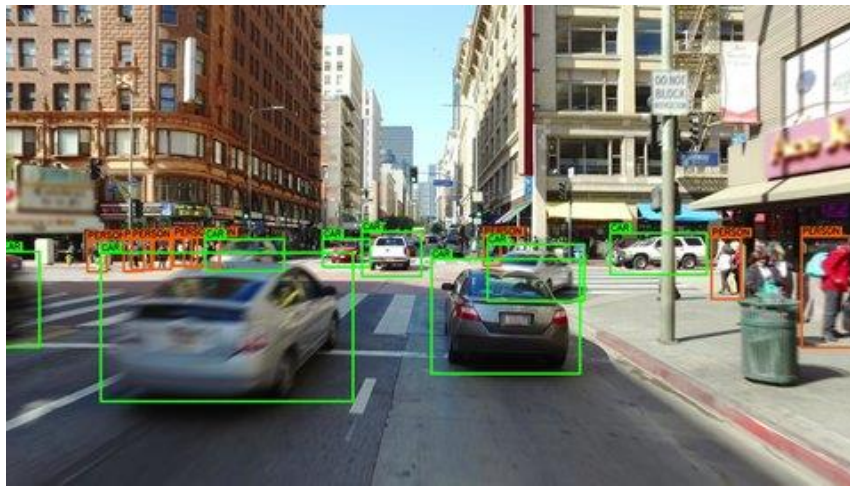


Figure 3.4 Détection des objets et la rétention des véhicules

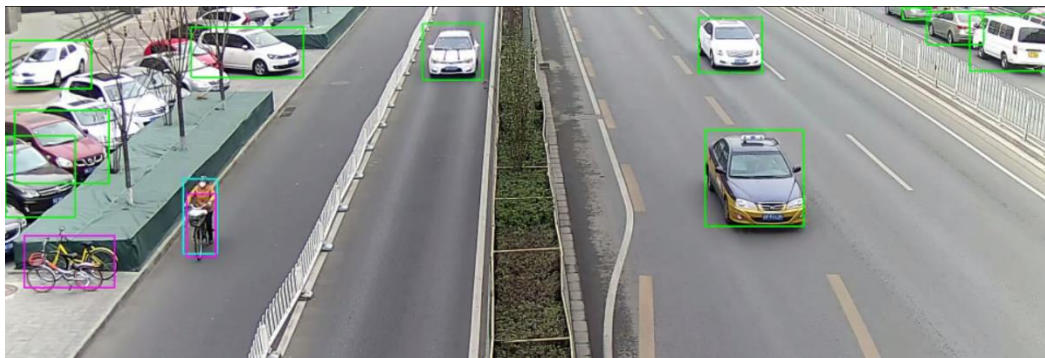


Figure 3.5 Détection des quatre classes

3.4.3 Module de détection de matricule et ROI

Le module de détection de plaque d'immatriculation est crucial pour localiser précisément les plaques sur les véhicules détectés. Voici comment nous avons conçu ce module et intégré YOLO (You Only Look Once) pour cette tâche spécifique :

Importance du module de détection de plaque d'immatriculation

La détection précise des plaques d'immatriculation est essentielle pour identifier chaque véhicule de manière unique dans le système de péage autoroutier. Ce module permet non seulement de localiser la plaque d'immatriculation mais aussi de délimiter une région d'intérêt (ROI) autour de celle-ci, facilitant ainsi la reconnaissance optique des caractères (OCR) dans les étapes suivantes.

Fonctionnalités clés du module :

1. **Utilisation de YOLO** : Nous avons adapté YOLO pour qu'il soit capable de détecter spécifiquement les plaques d'immatriculation des voitures. YOLO est entraîné sur un ensemble de données diversifié pour reconnaître les caractéristiques distinctives des plaques d'immatriculation, telles que leur forme et leur structure.
2. **Détection avec un cadre de délimitation** : Lorsque YOLO identifie une plaque d'immatriculation sur un véhicule, il encadre cette zone avec un cadre délimitant (bounding box) précis. Ce cadre est ajusté pour entourer la plaque avec les dimensions appropriées, permettant ainsi une localisation précise de la plaque sur l'image.
3. **Identification de la région d'intérêt (ROI)** : On définit la zone d'intérêt (ROI) comme la partie de l'image où la plaque d'immatriculation est repérée. Ce retour sur investissement (ROI) est extrait de l'image globale du véhicule et servira de point d'entrée pour la prochaine étape du processus, qui consiste à réaliser la reconnaissance optique des caractères (OCR).
4. **Capacité à gérer diverses conditions d'éclairage et d'angle** : YOLO est robuste et capable de détecter les plaques d'immatriculation dans différentes conditions d'éclairage et d'angle de

vue, ce qui est crucial pour assurer la fiabilité du système de péage autoroutier quelles que soient les conditions météorologiques ou la luminosité ambiante.



Figure 3.6 Détection de la plaque

3.4.3.1 Architecture de YOLOv8

L'architecture de YOLOv8 représente une évolution significative par rapport aux versions précédentes de YOLO, visant à améliorer à la fois la vitesse et la précision de la détection d'objets. Voici un aperçu détaillé des principaux éléments de YOLOv8 :

1. Backbone

Le backbone de YOLOv8 utilise le réseau Darknet-53, qui a été optimisé pour être à la fois rapide et précis. Le choix du Darknet-53 comme backbone est crucial pour l'extraction efficace des caractéristiques visuelles à partir des images. Comparé aux versions antérieures de YOLO, ce backbone améliore la capacité du modèle à capturer des informations discriminantes dans les images tout en conservant une vitesse de traitement élevée.

2. Neck

L'utilisation des Réseaux de Pyramide de Caractéristiques (FPN) dans le neck (cou) de YOLOv8 permet de détecter des objets de diverses tailles. Il s'agit d'une composante essentielle dans l'architecture, car elle intègre les caractéristiques extraites par le backbone à diverses échelles spatiales. En se basant sur les FPN, YOLOv8 a la capacité de produire des cartes de caractéristiques riches qui captent des objets à la fois de petite taille et de grande taille dans une scène.

3. Head

La tête (head) de YOLOv8 est conçue pour prédire les boîtes englobantes (bounding boxes) via une approche sans ancres (anchor-free). Cette technique améliore l'efficacité du modèle en éliminant la nécessité de prédéfinir des ancres spécifiques pour différentes tailles d'objets. La carte de caractéristiques plus large et le réseau convolutionnel amélioré de la tête permettent au modèle de produire des prédictions plus précises et rapides pour la localisation des objets dans une image.

Avantages supplémentaires

Outre les éléments clés mentionnés précédemment, YOLOv8 est également équipé d'une API conviviale qui facilite sa mise en œuvre dans différentes applications. L'architecture rapide et précise de cette version de YOLO est particulièrement adaptée pour détecter des objets en temps réel. Il est possible d'utiliser cette technologie pour des applications de vision par ordinateur avancées comme la segmentation d'instances, ce qui permet de détecter et de localiser de manière précise de nombreux objets dans des images ou des vidéos complexes.

En résumé, YOLOv8 représente une avancée significative dans la famille YOLO en matière de détection d'objets. Grâce à son architecture optimisée et à ses composants améliorés, il offre une performance supérieure pour la détection en temps réel tout en conservant une précision élevée, ce qui en fait un choix idéal pour les applications nécessitant une réponse rapide et fiable aux objets dans des scénarios variés.

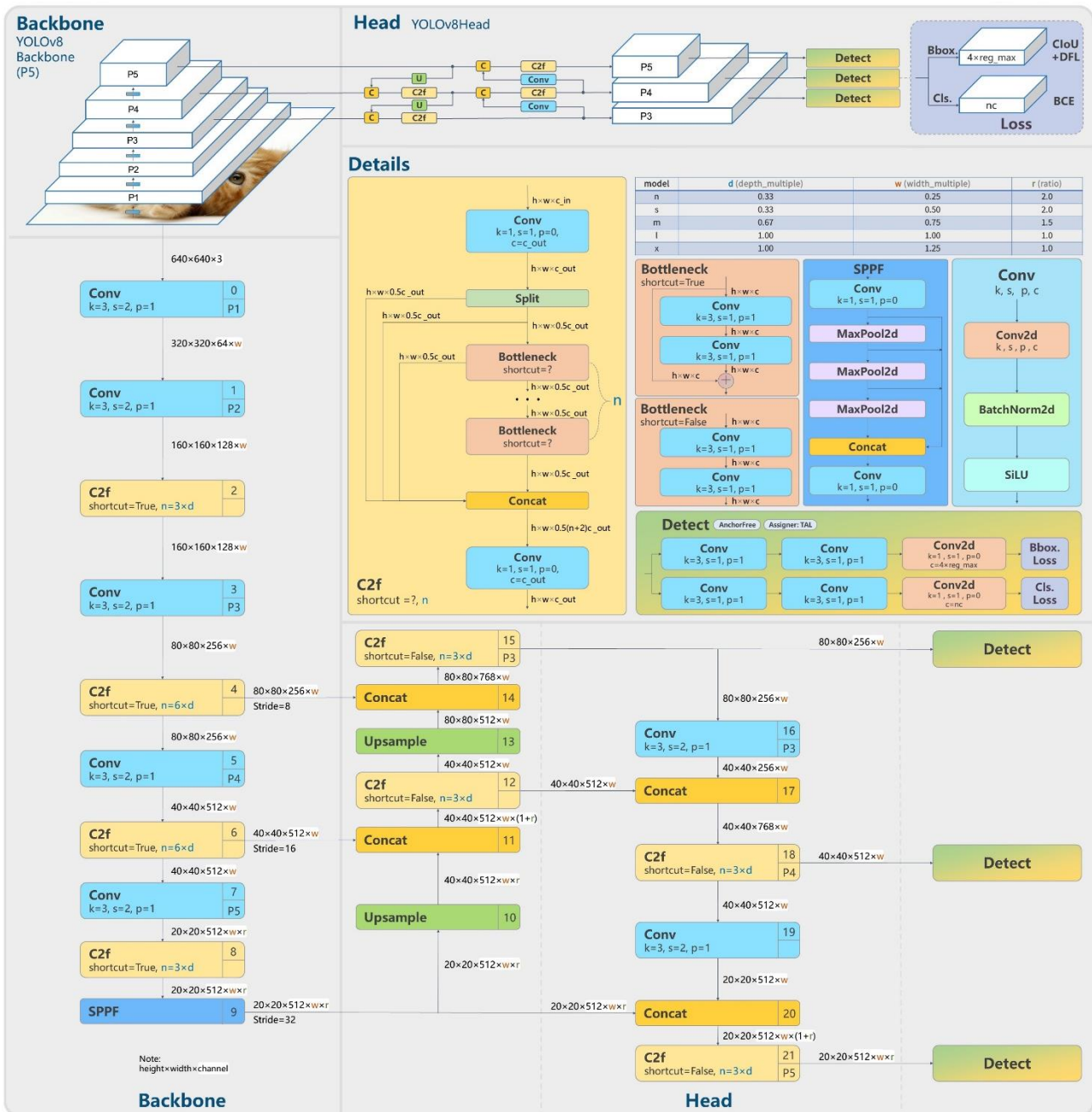


Figure 3.7 Architecture de YOLOv8[222]

3.4.3.1.1 Quelles sont les principales caractéristiques de YOLOv8 ?

YOLOv8 apporte des améliorations à la fois dans son architecture et dans l'expérience de développement.

1. Détection sans ancre

YOLOv8 utilise une approche de détection sans ancre. Contrairement aux modèles traditionnels qui nécessitent des boîtes d'ancrage pré-définies pour prédire les décalages par rapport à ces ancres, YOLOv8 prédit directement le centre d'un objet sans recourir à ces ancres. Cela simplifie le processus de prédiction des objets et augmente généralement la précision de la détection. Les boîtes d'ancrage sont des boîtes rectangulaires pré-définies de différentes tailles et formes utilisées par les modèles traditionnels pour améliorer la détection des objets à différentes échelles.

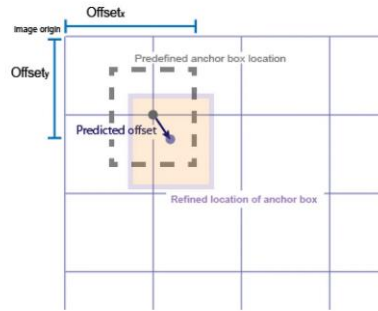


Figure 3.8 Visualisation d'une boîte d'ancrage

2. Nouvelles convolutions

Module initial (stem) :

Dans YOLOv8, le module initial a été modifié par rapport aux versions précédentes. La convolution initiale de taille 6×6 a été remplacée par une convolution de 3×3 . Cette modification vise à réduire la complexité et à améliorer l'efficacité du modèle dès le début du traitement des données.

Bloc de construction principal (C2f) :

Le bloc C2f remplace le bloc C3 des versions antérieures. Dans ce bloc, "C" fait référence à une couche de convolution, "2" indique la profondeur du bloc, et "f" représente le nombre de caractéristiques générées par les couches du bloc. Il comprend une couche de convolution, une normalisation par lot, et une fonction d'activation SiLU (Swish).

- **Structure Bottleneck** : La structure Bottleneck de YOLOv8 reste similaire à celle de YOLOv5, mais avec un changement notable dans le noyau de convolution initiale qui passe de 1×1 à 3×3 . Ce changement rapproche la structure de celle des blocs ResNet, augmentant ainsi la capacité du modèle à capturer les caractéristiques spatiales essentielles dans les images.

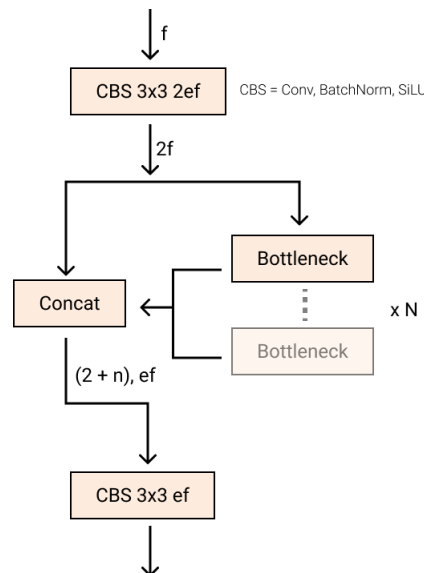


Figure 3.9 New YOLOv8 C2f module

3 Concaténation directe des caractéristiques

Dans la section "neck" de YOLOv8, une caractéristique notable est l'utilisation de la concaténation directe des caractéristiques sans maintenir les mêmes dimensions de canaux. Cette approche est une technique efficace pour réduire le nombre de paramètres et la taille globale des tenseurs du modèle, ce qui contribue à le rendre plus efficace et plus léger.

1. **Concaténation directe des caractéristiques** :

- Dans le contexte de réseaux neuronaux convolutionnels (CNN), la concaténation des caractéristiques se réfère à l'opération où plusieurs cartes de caractéristiques sont combinées le long de la dimension de canal.
- Contrairement aux approches traditionnelles qui visent à maintenir les mêmes dimensions de canal lors de la concaténation (ce qui peut augmenter la taille des tenseurs et le nombre de paramètres), YOLOv8 utilise une méthode où les dimensions de canal peuvent varier entre les caractéristiques concaténées.
- Cette flexibilité permet de combiner efficacement les informations extraites à différentes échelles spatiales et de niveaux de détail, sans ajouter de complexité excessive au modèle.

2. Avantages de la concaténation sans contrainte de dimensions :

- **Réduction de la complexité** : En permettant la concaténation directe des caractéristiques sans imposer de contraintes strictes sur les dimensions de canal, YOLOv8 réduit la complexité du modèle tout en maintenant sa capacité à capturer des informations détaillées à différentes résolutions.
- **Économie de mémoire et de calcul** : Moins de contraintes sur les dimensions de canal signifient moins de paramètres à apprendre et une taille de modèle réduite. Cela contribue à une meilleure efficacité en termes de mémoire et de calcul, ce qui est crucial pour les applications nécessitant une exécution rapide et légère.
- **Amélioration de la performance** : La capacité de fusionner efficacement des caractéristiques extraites à partir de différentes couches du modèle permet à YOLOv8 de maintenir ou d'améliorer sa précision tout en conservant une vitesse d'exécution élevée, ce qui est particulièrement important pour des applications telles que la détection d'objets en temps réel.

Les paramètres de YOLOv8 peuvent être divisés en plusieurs catégories :

Catégories	Paramètre	Définition	Valeur
Hyperparamètres du modèle	Model	Le type de modèle à utiliser (e.g., yolov8n, yolov8s, yolov8m, yolov8l, yolov8x, où n, s, m, l, x représentent les variantes nano, small, medium, large, et extra-large).	Yolov8n
	input_size	La taille de l'image d'entrée (par exemple 640, 1280).	640
	num_classes	Le nombre de classes à détecter.	1
Hyperparamètres d'entraînement	Epochs	Le nombre de cycles d'entraînement.	15
	batch_size	Le nombre d'exemples par lot.	16
	learning_rate	Le taux d'apprentissage.	0.01
	Momentum	Le momentum pour l'optimiseur.	0.937
	weight_decay	Le coefficient de décroissance de poids.	0.0005
	warmup_epochs	Le nombre d'époques de warmup (pour initialiser l'entraînement).	3.0

	cos_lr	Utilisation de la politique de taux d'apprentissage cosinus.	False
Hyperparamètres de régularisation	Dropout	Taux de dropout pour la régularisation.	0.0
	label_smoothing	Lissage des étiquettes pour réduire la suradaptation.	0.0
Paramètres de données	Data	Le chemin vers le fichier de configuration des données (dataset), qui définit les chemins des ensembles d'entraînement, de validation et de test, ainsi que les classes.	/content/dataset/data.yaml
	Augmentation	Techniques d'augmentation des données (e.g., flip, rotation, zoom, etc.).	Non spécifié
Paramètres de l'architecture	depth_multiple	Facteur de multiplicateur de profondeur pour ajuster le nombre de couches.	Non spécifié
	width_multiple	Facteur de multiplicateur de largeur pour ajuster la largeur des couches.	Non spécifié
Paramètres de post-traitement	conf_threshold	Seuil de confiance pour les prédictions.	Non spécifié
	iou_threshold	Seuil d'Intersection over Union (IoU) pour la suppression des doublons (Non-Maximum Suppression).	0.7
Options de logging et de sauvegarde	save_period	Intervalle d'époques pour sauvegarder les checkpoints.	-1
	Project	Dossier pour enregistrer les résultats.	Non spécifié
	Name	Nom de l'expérience.	train2

Tableau 3.1 Table du paramètre modèle de matricule

3.4.3.2 L'entraînement du modèle YOLO de détection de matricule

Pour entraîner notre modèle YOLO à détecter les plaques d'immatriculation, nous avons suivi une série d'étapes méthodiques, depuis la préparation de l'environnement jusqu'à l'évaluation du modèle. Voici un aperçu détaillé de ce processus :

Étape 1: Préparer l'Environnement

Nous avons commencé par préparer notre environnement de travail en installant les bibliothèques nécessaires et en configurant les répertoires appropriés.

1. **Bibliothèques** : La bibliothèque `ultralytics` est utilisée pour YOLOv8.
2. **Dataset** : Les données sont placées dans le répertoire `/content/dataset`.

```
ROOT_DIR = '/content/dataset'
import os
from ultralytics import YOLO
```

Étape 2: Initialiser le Modèle YOLOv8

Nous avons initialisé un nouveau modèle YOLOv8 en utilisant le fichier de configuration `yolov8n.yaml`.

```
# Initialisation du modèle YOLOv8
model = YOLO('yolov8n.yaml')
```

Étape 3: Préparer la Dataset

- 1. Présentation du Dataset Utilisé** Nous avons utilisé le **License Plate Recognition Object Detection Dataset (v4, resized640_aug3x-ACCURATE)** disponible sur [Roboflow Universe](#). Ce dataset est crucial pour entraîner et évaluer notre modèle de détection de plaques d'immatriculation.
 - **Taille du Dataset** : 24,242 images annotées.
 - **Conditions variées** : Les images incluent divers types de plaques sous différentes conditions de luminosité, angles de vue, et contextes environnementaux.
- 2. Répartition du Dataset**
 - **Ensemble d'entraînement (TRAIN SET)** : 87% (21,174 images)
 - **Ensemble de validation (VALID SET)** : 8% (2,048 images)
 - **Ensemble de test (TEST SET)** : 4% (1,020 images)
- 3. Prétraitement des Données** Afin d'assurer la qualité et la cohérence des informations, nous avons mis en place les étapes de prétraitement suivantes :
 - **Orientation Automatique (Auto-Orient)** : Correction de l'orientation des images.
 - **Redimensionnement (Resize)** : Toutes les images ont été redimensionnées à 640x640 pixels. Ce redimensionnement uniformise la taille des images, facilitant leur traitement par les modèles de réseaux neuronaux convolutifs.
- 4. Annotations** Chaque image est annotée avec des bounding boxes définissant la position des plaques d'immatriculation et des étiquettes indiquant les caractères alphanumériques. Ces annotations fournissent les informations nécessaires pour entraîner un modèle capable non seulement de localiser les plaques, mais aussi de reconnaître le texte qu'elles contiennent.

Ce dataset a été choisi en raison de sa richesse et de sa diversité, offrant une large gamme de scénarios de plaques d'immatriculation capturées dans des conditions variées. Il a servi à entraîner un modèle de détection et de reconnaissance des plaques, permettant une évaluation précise et une généralisation robuste du modèle développé.

L'ensemble de données que nous avons téléchargé est joint sous la forme suivante :


```
/content/dataset
|
|─ images/
|   |─ train/
|   |─ val/
|   └─ test/
|─ labels/
|   |─ train/
|   |─ val/
|   └─ test/
data.yaml
```

Le fichier data.yaml contient la configuration pour le dataset, incluant les chemins vers les images et les annotations pour l'entraînement, la validation, et les tests. Voici une structure exemple de data.yaml :

```
train: /content/dataset/images/train
val: /content/dataset/images/val
test: /content/dataset/images/test

nc: 1 # Nombre de classes (par exemple, 1 pour les plaques d'immatriculation)
names: ['license_plate']
```

Étape 4: Entraîner le Modèle

Nous avons démarré le processus d'entraînement en spécifiant la configuration du dataset, le nombre d'époques, et d'autres paramètres.

```
model.train(data='/content/dataset/data.yaml', epochs=15)
```

1. **Chargement des Données** : Le modèle charge les images et les annotations selon les chemins spécifiés dans data.yaml.
2. **Entraînement du Modèle** : Le modèle s'entraîne sur le nombre d'époques spécifié, en mettant à jour les poids pour minimiser la perte.
3. **Validation** : À chaque époque, le modèle évalue ses performances sur l'ensemble de validation.

```

Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/train2
Starting training for 15 epochs...

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
1/15 2.44G 2.102 2.212 2.472 20 640: 100% | 1323/1324 [08:51<00:00, 1.71it/s] Plan failed with a cudnnExcepti
1/15 2.49G 2.102 2.211 2.471 3 640: 100% | 1324/1324 [08:53<00:00, 2.48it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:25<00:00, 2.46it/s]
all 2046 2132 0.89 0.745 0.813 0.424

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
2/15 2.32G 1.485 1.106 1.721 8 640: 100% | 1324/1324 [08:24<00:00, 2.62it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:24<00:00, 2.59it/s]
all 2046 2132 0.936 0.839 0.906 0.545

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
3/15 2.31G 1.405 0.9681 1.654 6 640: 100% | 1324/1324 [08:20<00:00, 2.64it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:26<00:00, 2.44it/s]
all 2046 2132 0.937 0.867 0.925 0.568

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
4/15 2.31G 1.344 0.8982 1.6 8 640: 100% | 1324/1324 [08:26<00:00, 2.61it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:24<00:00, 2.66it/s]
all 2046 2132 0.953 0.888 0.941 0.572

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
12/15 2.3G 1.086 0.5189 1.416 5 640: 100% | 1324/1324 [07:59<00:00, 2.76it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:24<00:00, 2.64it/s]
all 2046 2132 0.979 0.931 0.97 0.645

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
13/15 2.3G 1.07 0.5 1.402 5 640: 100% | 1324/1324 [07:52<00:00, 2.80it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:25<00:00, 2.49it/s]
all 2046 2132 0.982 0.93 0.969 0.645

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
14/15 2.3G 1.057 0.4844 1.392 5 640: 100% | 1324/1324 [08:01<00:00, 2.75it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:23<00:00, 2.77it/s]
all 2046 2132 0.983 0.937 0.97 0.653

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
15/15 2.3G 1.038 0.4707 1.373 6 640: 100% | 1324/1324 [08:10<00:00, 2.70it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 64/64 [00:27<00:00, 2.34it/s]
all 2046 2132 0.985 0.938 0.972 0.661

```

Étape 5: Sauvegarder le Modèle Entraîné

Une fois l'entraînement terminé, nous avons enregistré les poids du modèle pour une utilisation ultérieure.

```
model.save('yolov8_trained_model.pt')
```

Étape 6: Afficher les Métriques

Nous avons présenté les métriques d'entraînement pour évaluer les performances du modèle.

```
# Affichage des métriques d'entraînement
results = model.val()
print(results)
```

Ce processus structuré nous a permis de développer un modèle YOLOv8 capable de détecter efficacement les plaques d'immatriculation dans diverses conditions, en assurant une haute précision et une robustesse dans les prédictions.

3.4.4 Module de reconnaissance OCR

La reconnaissance des plaques d'immatriculation est une étape cruciale pour le bon fonctionnement de notre système de contrôle d'accès aux péages autoroutiers. Bien que notre modèle initial ait

rencontré des difficultés à lire les numéros de plaque avec précision, nous avons trouvé une solution efficace en utilisant Pytesseract, une bibliothèque spécialisée en reconnaissance optique de caractères (OCR). Voici les détails de cette approche et l'entraînement du modèle de reconnaissance.

3.4.4.1 L'entraînement du modèle de reconnaissance avec YOLOv8

Pour améliorer la précision de la reconnaissance des plaques d'immatriculation, nous avons entraîné un modèle YOLOv8 spécifiquement pour détecter les plaques d'immatriculation avant de les passer à Pytesseract pour la reconnaissance des caractères. Voici les étapes détaillées :

Étape 1: Préparer l'Environnement

Nous avons préparé notre environnement de travail en installant les bibliothèques nécessaires et en configurant les répertoires appropriés.

1. **Bibliothèques** : Utilisation des bibliothèques `ultralytics` pour YOLOv8 et `pytesseract` pour l'OCR.
2. **Dataset** : Les données sont placées dans le répertoire `/content/dataset`.

Étape 2: Initialiser le Modèle YOLOv8

Nous avons initialisé un modèle YOLOv8 pour reconnaître les numéros dans les plaques d'immatriculation.

```
from ultralytics import YOLO

# Initialisation du modèle YOLOv8
model = YOLO('yolov8n.yaml')
```

Étape 3: Préparer Votre Dataset

Cette base de données contient des images alphanumériques et annotées, pour être ensuite entraîné avec YOLOv8.

1. **Présentation du Dataset Utilisé**
 - **LicenseplateOCR Dataset.**
 - **Taille du Dataset** : 24,242 images annotées.
 - **Répartition du Dataset** :
 - **Ensemble d'entraînement (TRAIN SET)** : 5247 images
 - **Ensemble de validation (VALID SET)** : 499 images
 - **Ensemble de test (TEST SET)** : 252 images
2. **Prétraitement des Données**
 - **Redimensionnement (Resize)** : Toutes les images ont été redimensionnées à 640x640 pixels.

Le fichier `data.yaml` contient la configuration du dataset, incluant les chemins vers les images et les annotations.

Étape 4 : Entraîner le Modèle

Nous avons démarré le processus d'entraînement en spécifiant la configuration du dataset, le nombre d'époques, et d'autres paramètres.

```
model.train(data='/content/dataset/data.yaml', epochs=10)
```

1. **Chargement des Données** : Le modèle charge les images et les annotations selon les chemins spécifiés dans `data.yaml`.
2. **Entraînement du Modèle** : Le modèle s'entraîne sur le nombre d'époques spécifié, en mettant à jour les poids pour minimiser la perte.
3. **Validation** : À chaque époque, le modèle évalue ses performances sur l'ensemble de validation.

Étape 5: Sauvegarder le Modèle Entraîné

Une fois l'entraînement terminé, nous avons enregistré les poids du modèle pour une utilisation ultérieure.

```
model.save('yolov8_trained_model.pt')
```

Catégories	Paramètre	Définition	Valeur
Hyperparamètres du modèle	Model	Le type de modèle à utiliser (e.g., yolov8n, yolov8s, yolov8m, yolov8l, yolov8x, où n, s, m, l, x représentent les variantes nano, small, medium, large, et extra-large).	Yolov8n
	input_size	La taille de l'image d'entrée (par exemple 640, 1280).	640
	num_classes	Le nombre de classes à détecter.	1
Hyperparamètres d'entraînement	Epochs	Le nombre de cycles d'entraînement.	10
	batch_size	Le nombre d'exemples par lot.	8
	Workers	/	8
	learning_rate	Le taux d'apprentissage.	0.01
	Momentum	Le momentum pour l'optimiseur.	0.937
	weight_decay	Le coefficient de décroissance de poids.	0.0005
	warmup_epochs	Le nombre d'époques de warmup (pour initialiser l'entraînement).	3
	cos_lr	Utilisation de la politique de	False

		taux d'apprentissage cosinus.	
Hyperparamètres de régularisation	Dropout	Taux de dropout pour la régularisation.	0.0
	label_smoothing	Lissage des étiquettes pour réduire la suradaptation.	0.0
Paramètres de données	Data	Le chemin vers le fichier de configuration des données (dataset), qui définit les chemins des ensembles d'entraînement, de validation et de test, ainsi que les classes.	/content/dataset/data.yaml
	Augmentation	Techniques d'augmentation des données (e.g., flip, rotation, zoom, etc.).	Non spécifié
Paramètres de l'architecture	depth_multiple	Facteur de multiplicateur de profondeur pour ajuster le nombre de couches.	Non spécifié
	width_multiple	Facteur de multiplicateur de largeur pour ajuster la largeur des couches.	Non spécifié
Paramètres de post-traitement	conf_threshold	Seuil de confiance pour les prédictions.	Non spécifié
	iou_threshold	Seuil d'Intersection over Union (IoU) pour la suppression des doublons (Non-Maximum Suppression).	0.7
Options de logging et de sauvegarde	save_period	Intervalle d'époques pour sauvegarder les checkpoints.	-1
	Project	Dossier pour enregistrer les résultats.	Non spécifié

	Name	Nom de l'expérience.	Train3
--	-------------	----------------------	--------

Tableau 3.2 Table du paramètre modèle de OCR

Étape 6: Reconnaissance des Caractères avec Pytesseract

Après avoir détecté les plaques d'immatriculation, nous avons utilisé Pytesseract pour la reconnaissance des caractères.

1. **Prétraitement de l'Image** Nous avons appliqué des filtres pour améliorer la qualité de l'image avant de la passer à Pytesseract.

```
import cv2
import pytesseract

# Lecture de l'image et application de filtres
image = cv2.imread('path_to_detected_plate_image.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
filtered = cv2.bilateralFilter(gray, 11, 17, 17)
edged = cv2.Canny(filtered, 30, 200)

# Utilisation de Pytesseract pour l'OCR
text = pytesseract.image_to_string(edged, config='--psm 11')
print("Detected License Plate Number is:", text)
```

2. **Reconnaissance des Caractères** Pytesseract a été utilisé pour lire les caractères des plaques d'immatriculation détectées.

3.4.4.2 Explication Théorique de PyTesseract

1. Prétraitement des Images

Le prétraitement des images est une étape cruciale pour améliorer la précision de la reconnaissance optique de caractères (OCR) avec PyTesseract. Ce processus consiste en une série de transformations appliquées à l'image avant l'application de l'OCR pour améliorer la qualité du texte extrait. Voici les étapes typiques du prétraitement :

- **Conversion en niveaux de gris (Grayscale)** : La conversion de l'image en niveaux de gris réduit l'image à des nuances de gris, éliminant les informations de couleur tout en conservant les informations essentielles sur l'intensité des pixels. Cela facilite la détection des caractères en augmentant le contraste entre le texte et le fond.
- **Binarisation** : La binarisation convertit l'image en noir et blanc, en définissant les pixels au-dessus d'un seuil spécifique comme blancs et ceux en dessous comme noirs. Cette opération renforce le contraste entre les caractères et le fond, facilitant ainsi leur détection.
- **Suppression du bruit** : Les techniques de filtrage sont utilisées pour réduire le bruit dans l'image, comme les petites taches ou les distorsions qui pourraient affecter la précision de l'OCR. Des filtres comme le filtre bilatéral ou le filtre gaussien sont couramment utilisés.
- **Redimensionnement** : Le redimensionnement de l'image à une taille plus appropriée pour l'OCR peut améliorer les résultats, en particulier pour les textes petits ou de faible résolution. Cela permet d'ajuster la taille des caractères pour mieux correspondre aux capacités de reconnaissance du modèle.
- **Correction de l'orientation** : Les images avec des textes inclinés ou déformés peuvent être corrigées pour aligner le texte horizontalement, ce qui facilite l'OCR. Des algorithmes de détection des contours et des transformations géométriques sont utilisés pour rectifier l'orientation.

2. OCR (Reconnaissance Optique de Caractères)

L'étape principale de l'OCR avec PyTesseract consiste à utiliser Tesseract pour extraire le texte de l'image traitée. Voici comment cela fonctionne théoriquement :

- **Analyse des pixels** : Tesseract analyse les pixels de l'image pour détecter les contours et les formes des caractères. Cette analyse pixel par pixel permet de différencier le texte du fond.
- **Segmentation** : L'image est segmentée en blocs de texte, lignes et caractères. Cette étape identifie où se trouvent les caractères et comment ils sont regroupés en mots et lignes. La segmentation précise est cruciale pour une reconnaissance exacte.
- **Correspondance de motifs** : Les formes détectées sont comparées à des modèles de caractères connus à l'aide de techniques de correspondance de motifs. Tesseract utilise des réseaux neuronaux et des algorithmes de classification pour identifier chaque caractère. Ce processus permet d'associer les pixels segmentés à des lettres spécifiques.
- **Reconnaissance des caractères** : Tesseract associe chaque forme de caractère à une lettre, un chiffre ou un symbole spécifique, produisant ainsi une représentation textuelle de l'image. Les résultats sont alors assemblés pour former des mots et des phrases cohérents.

3. Post-Traitement

Le post-traitement implique le nettoyage et la correction du texte extrait pour améliorer sa lisibilité et sa précision. Les étapes courantes comprennent :

- **Correction des erreurs de reconnaissance** : Le texte OCR brut peut contenir des erreurs dues à des caractères mal reconnus ou mal segmentés. Des techniques comme les dictionnaires de mots ou les correcteurs orthographiques peuvent être utilisées pour corriger ces erreurs. Des algorithmes de correction automatique peuvent également être employés pour ajuster les caractères mal interprétés.
- **Formatage** : La mise en forme du texte, comme la suppression des espaces indésirables ou l'ajustement des marges, peut être nécessaire pour obtenir un texte propre et bien structuré. Cela inclut également la normalisation des espaces entre les mots et les lignes.
- **Validation** : Assurez-vous que le contenu extrait correspond aux formats ou aux motifs attendus, tels que les adresses e-mail, les numéros de téléphone, etc. Il est possible que cela nécessite l'emploi de regex (expressions régulières) afin de garantir que le texte reconnu respecte les formats spécifiques et attendus.

En conclusion, le prétraitement, l'application de l'OCR et le post-traitement sont des étapes cruciales pour améliorer la précision de la reconnaissance de texte avec PyTesseract. En appliquant ces étapes méthodiquement, on peut significativement améliorer les performances de la reconnaissance des plaques d'immatriculation dans notre système de contrôle d'accès aux péages autoroutiers.

Le prétraitement des images est crucial pour améliorer la précision de la reconnaissance optique de caractères (OCR) avec PyTesseract. Ce processus consiste en une série de transformations appliquées à l'image avant l'application de l'OCR pour améliorer la qualité du texte extrait.

4 Exemple :

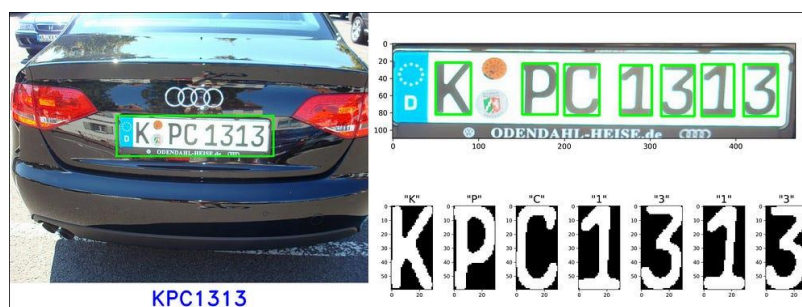


Figure 3.10 Exemple de OCR par Pytesseract

En utilisant YOLOv8 pour détecter les plaques d'immatriculation et Pytesseract pour la reconnaissance des caractères, nous avons considérablement amélioré la précision et l'efficacité de notre système. Cette approche a permis de surmonter les défis initiaux et de fournir des résultats fiables pour le contrôle d'accès aux péages autoroutiers.

3.4.5 Module de La Vérification avec la Base de Données

Dans cette partie, nous comparons le numéro délivré par la lecture de la plaque d'immatriculation avec les numéros de la base de données de manière très simplifiée. Cela aboutit soit à une correspondance, soit à une non-concordance, c'est-à-dire que la voiture est immatriculée ou non.

3.4.5.1 Système de Gestion de Contrôle d'Accès

Le système de contrôle d'accès est une plateforme qui permet de superviser les gardes des points de contrôle qui sont responsables de l'ajout de conducteurs et de l'ouverture de la barrière. La plateforme propose des options pour ajouter des conducteurs et leurs véhicules, permettant ainsi à une personne d'être enregistrée avec plusieurs véhicules, chacun étant identifié par des informations précises, telles que le numéro de plaque d'immatriculation. Si aucun conducteur n'a été enregistré auparavant ou si la plaque d'immatriculation ne correspond pas, le conducteur peut être ajouté manuellement ou s'ajouter lui-même.

Grâce à la plateforme, il est également possible d'ouvrir la caméra appropriée ou plusieurs caméras simultanément.

a. La Base de Données

Les tables de la base de données sont constituées de huit, chacune avec des attributs et des clés primaires et étrangères.

Voici la structure détaillée des tables de la base de données :

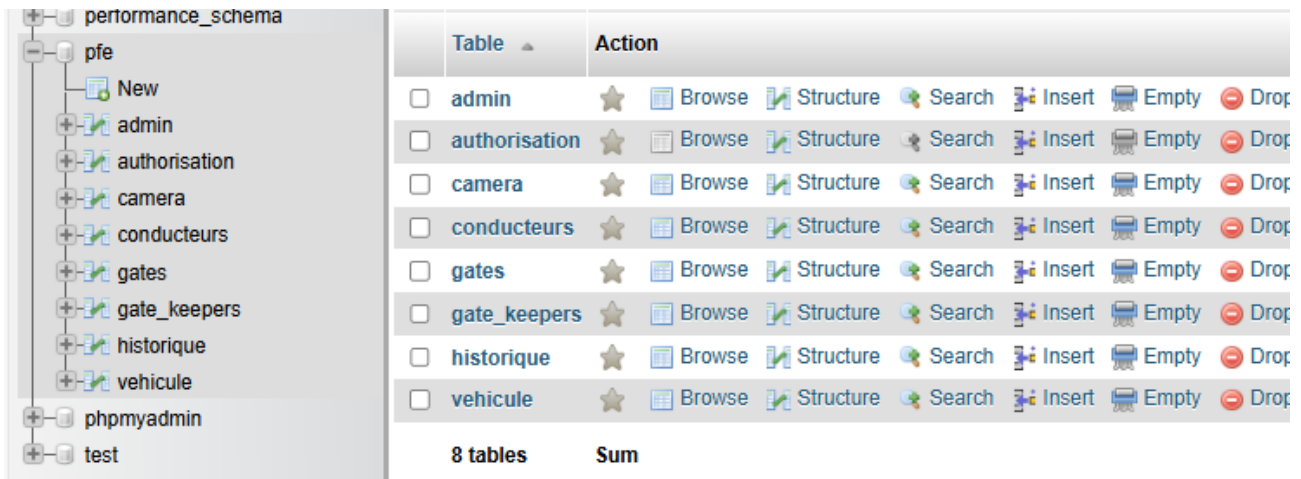


Table	Action
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> authorisation	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> camera	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> conducteurs	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> gates	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> gate_keepers	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> historique	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> vehicule	★ Browse Structure Search Insert Empty Drop
8 tables	Sum

Workflow de la Vérification (Suite)

- Lecture de la Plaque d'Immatriculation :**
 - Une fois la plaque d'immatriculation détectée et reconnue par le module OCR, le numéro de plaque est extrait.
- Comparaison avec la Base de Données :**
 - Le numéro de plaque extrait est comparé avec les numéros enregistrés dans la table `vehicles`.
 - Si une correspondance est trouvée, l'accès est autorisé, et un enregistrement est ajouté à la table `historique` avec le statut "autorisé".
 - Si aucune correspondance n'est trouvée, le numéro de plaque est vérifié dans la table `vehicles` pour détecter toute tentative d'accès précédent non autorisée.
- Gestion des Véhicules Non Enregistrés :**
 - Si la plaque n'est pas reconnue et n'a pas été précédemment détectée, une alerte est générée et un enregistrement est ajouté à la table `vehicles`.
 - Le système peut envoyer une alerte aux gardes via la table `gate_keepers` pour les informer d'un véhicule non enregistré tentant d'accéder.
 - Les gardes peuvent ensuite décider d'ajouter manuellement le véhicule et le conducteur dans le système, ou prendre d'autres mesures nécessaires.
- Mise à Jour et Maintenance :**

- Les gardes et les administrateurs peuvent utiliser la plateforme pour mettre à jour les informations des conducteurs et des véhicules, vérifier et résoudre les alertes, et gérer les caméras et les points de contrôle.

3.4.5.2 Flux de Contrôle du Module de Vérification

1. Détection de la Plaque :

- Le système capture l'image du véhicule entrant via la caméra et détecte la plaque d'immatriculation en utilisant le module YOLOv8.

2. Extraction et OCR :

- La plaque d'immatriculation est extraite de l'image et les caractères sont reconnus par PyTesseract après prétraitement de l'image.

3. Comparaison de la Plaque :

- Les données de la base de données des véhicules sont comparées au numéro de plaque.
- En cas de correspondance, le système autorise l'accès et enregistre l'événement dans `historique`.
- En cas de non-correspondance, une alerte est générée, et le véhicule est enregistré dans `vehicles`.

4. Alerte et Enregistrement :

- Les gardes reçoivent une alerte pour les véhicules non enregistrés, et peuvent intervenir pour ajouter le véhicule et le conducteur à la base de données.

Le module de vérification avec la base de données assure que seuls les véhicules autorisés peuvent accéder aux installations. En utilisant une base de données relationnelle robuste et une architecture bien structurée, le système permet une gestion efficace des accès et une réponse rapide aux tentatives d'accès non autorisées. Le processus de vérification s'intègre parfaitement aux autres modules de détection et de reconnaissance, garantissant une solution complète et sécurisée pour le contrôle d'accès basé sur la reconnaissance des plaques d'immatriculation.

b. Diagramme de classe

Voici une représentation des principales classes et de leurs relations. Ce diagramme inclut les entités principales comme les Gardes, les Conducteurs, les Véhicules, les Points de Contrôle, les Caméras, les Journaux d'Accès, et les Alertes.

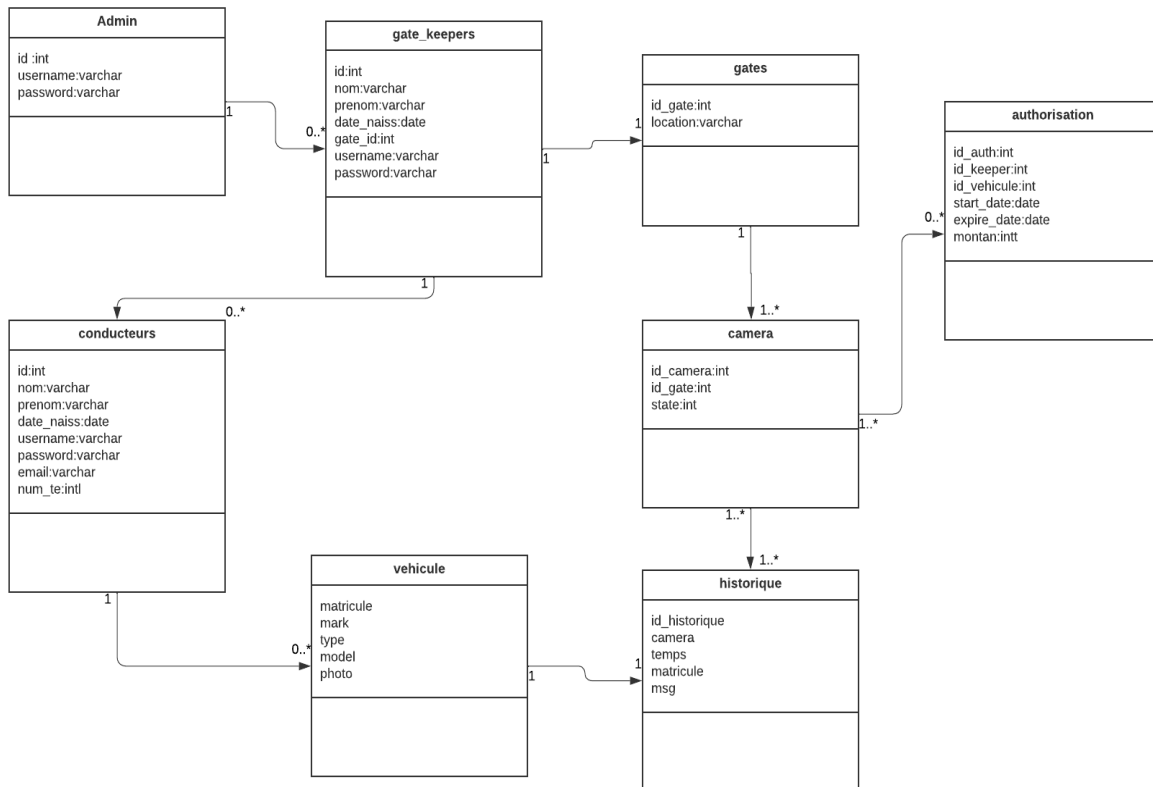


Figure 3.11 diagramme de classe

3.4.5.2 Module IOT

Après avoir reçu le message " la plaque se trouve dans la base de données " ou " la plaque ne se trouve pas dans la base de données " de la base de données, l'ESP32 effectue les étapes suivantes pour contrôler la barrière :

1. **Interprétation de la Réponse :**
 - " la plaque se trouve dans la base de données " : Le véhicule est autorisé à entrer.
 - " la plaque ne se trouve pas dans la base de données " : Le véhicule n'est pas autorisé à entrer.
2. **Action sur le Relais :**
 - Si la réponse est " la plaque se trouve dans la base de données " :
 - L'ESP32 envoie un signal pour activer le relais.
 - Le relais ferme le circuit, alimentant ainsi le moteur de la barrière.
 - La barrière s'ouvre pour permettre au véhicule de passer.
 - Si la réponse est " la plaque ne se trouve pas dans la base de données " :
 - L'ESP32 n'envoie pas de signal au relais.
 - Le relais reste inactif, le circuit reste ouvert.
 - La barrière reste fermée et n'autorise pas le passage.

Ce système assure la sécurité et l'efficacité de la gestion des accès, en n'ouvrant la barrière qu'aux véhicules autorisés sur la base des informations validées par la base de données.

3.5 Métriques

Vrais Positifs (TP) : Cela désigne les cas où le modèle a correctement prédit un échantillon positif (par exemple, un "chat") comme étant positif. Cela correspond aux prédictions exactes de la présence de l'objet.

Faux Positifs (FP) : Cela fait référence aux cas où le modèle a incorrectement prédit un échantillon négatif (par exemple, une catégorie "autre") comme étant positif.

Vrais Négatifs (TN) : Cela concerne les cas où le modèle a correctement prédit un échantillon négatif (par exemple, une catégorie "autre") comme étant négatif.

Faux Négatifs (FN): Cela se réfère aux cas où le modèle a mal prédit un échantillon positif (par exemple, un "chat") comme étant négatif.

Précision Moyenne (mAP): La Précision Moyenne Moyenne est une mesure essentielle pour évaluer les performances des modèles de détection d'objets, comme YOLOv8. Elle évalue la précision du modèle en tenant compte des valeurs de précision et de rappel à différents seuils d'intersection sur union (IoU). Une mAP plus élevée indique de meilleures performances en détection.

Calcul : Moyenne des précisions obtenues à différents seuils d'IoU.

Intersection sur Union (IoU) : L'IoU mesure le chevauchement entre les boîtes englobantes prédites et les boîtes englobantes de la vérité terrain. Elle est calculée comme le rapport de l'aire d'intersection sur l'aire d'union entre les deux boîtes. L'IoU est utilisée pour déterminer si une détection prédite est un vrai positif ou un faux positif.

Précision: La précision mesure la proportion d'objets correctement prédits parmi tous les objets prédits. Elle est calculée comme le rapport des vrais positifs sur la somme des vrais positifs et des faux positifs.

$$Precision = \frac{TP}{TP + FP}$$

Rappel : Aussi appelé sensibilité, le rappel mesure la proportion d'objets correctement prédits parmi tous les objets réels dans l'ensemble de données. Il est calculé comme le rapport des vrais positifs sur la somme des vrais positifs et des faux négatifs.

$$Recall = \frac{TP}{TP + FN}$$

Score F1 : Le score F1 est la moyenne harmonique de la précision et du rappel. Il fournit une seule métrique pour évaluer le compromis entre précision et rappel. Le score F1 est particulièrement utile lorsque les classes positives et négatives sont déséquilibrées dans l'ensemble de données.

$$\text{Score } F1 = \frac{\text{Précision} * \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Ces métriques permettent d'évaluer les performances du modèle YOLOv8 en termes de précision, de rappel et de capacité globale de détection des objets. En analysant ces métriques, on peut évaluer les points forts et les faiblesses du modèle et effectuer les ajustements nécessaires pour améliorer ses performances.

3.6 Conclusion

Dans ce chapitre, nous avons discuté en détail du processus de formation et de configuration du modèle YOLOv8 pour un système conçu pour identifier les plaques d'immatriculation des voitures sur l'autoroute. Voici les principaux points abordés : **(1) Architecture du Système** : Nous avons décrit l'architecture globale du système, divisée en trois étapes clés : identification de l'image et segmentation en véhicules et plaques numérotées, lecture complète de la plaque d'immatriculation, et gestion de l'accès avec ouverture automatique des barrières. **(2) Collecte et Prétraitement des Données** : La préparation des données a été cruciale pour le succès de l'entraînement du modèle. Nous avons utilisé le License Plate Recognition Object Detection Dataset, composé de plus de 24 000 images annotées, et avons appliqué diverses techniques de prétraitement pour améliorer la qualité des données d'entrée. **(3) Configuration du Modèle YOLOv8** : Nous avons expliqué l'architecture du modèle YOLOv8, en détaillant les améliorations apportées par rapport aux versions précédentes, notamment l'utilisation de la détection sans ancre, les nouvelles convolutions et la concaténation directe des caractéristiques. **(4) Procédures d'entraînement** : Le processus d'entraînement du modèle a été décrit, incluant la préparation de l'environnement, l'initialisation du modèle, l'entraînement sur le dataset, et l'évaluation des performances du modèle à l'aide de métriques telles que la précision, le rappel, le score F1 et l'IoU. **(5) Prédiction et Évaluation des Métriques** : Nous avons discuté des différentes métriques utilisées pour évaluer les performances du modèle, telles que les Vrais Positifs, Faux Positifs, Vrais Négatifs, Faux Négatifs, la Précision Moyenne (mAP), l'Intersection sur Union (IoU), la précision, le rappel et le score F1. **(6) Intégration de Modules IoT** : L'intégration du modèle avec les modules IoT a été abordée, notamment l'utilisation de caméras pour capturer les images des véhicules, la détection des plaques d'immatriculation, et la gestion de l'accès avec comparaison des plaques lues avec celles enregistrées dans la base de données.

Ce chapitre a établi une base solide pour comprendre la conception et la mise en œuvre du système de reconnaissance des plaques d'immatriculation basé sur YOLOv8. En suivant les étapes décrites, nous avons réussi à créer un système capable de détecter et de reconnaître avec précision les plaques d'immatriculation des véhicules sur l'autoroute, intégrant efficacement les technologies de vision par ordinateur et les modules IoT pour améliorer la gestion et le contrôle de l'accès.

Dans le prochain chapitre, nous examinerons les résultats obtenus, en analysant les performances du système à l'aide des métriques discutées et en identifiant les points d'amélioration possibles pour augmenter l'efficacité et la précision du système.

Chapitre 4
Implémentation and Résultat

4.1 Introduction

La création de logiciels repose sur l'environnement de développement, qui englobe les outils matériels et logiciels utilisés par les développeurs pour coder et tester leurs applications. L'environnement de développement et les outils logiciels utilisés pour notre application sont examinés dans ce chapitre. Les environnements de développement intégrés (IDE), les éditeurs de texte, les langages de programmation et les frameworks utilisés seront abordés. Nous exposerons également en détail les différentes étapes de la conception et de la mise en place de notre système, en examinant les difficultés rencontrées et en soulignant les résultats et les accomplissements obtenus jusqu'à présent.

4.2 Environnement de développement

4.2.1 Environnement matériel

Google Colab : Google Colab est un environnement de notebook basé sur le cloud proposé par Google, similaire à Google Docs, qui facilite l'édition de documents et offre une intégration transparente avec les bibliothèques de machine learning populaires. Cela permet aux utilisateurs d'incorporer facilement ces bibliothèques dans leurs notebooks. Colab propose trois types de runtimes pour les notebooks : CPU, GPU et TPU, avec un total de 12 heures d'exécution continue. Après cette limite, la machine virtuelle est réinitialisée, obligeant les utilisateurs à recommencer en raison des contraintes d'utilisation des ressources.

- **CPU dans Google Colab :** Le CPU (Central Processing Unit) dans Colab est l'unité de traitement principale responsable de l'exécution des instructions et des calculs au sein de l'environnement de machine virtuelle de Google. Il gère une large gamme de tâches informatiques et exécute le code de manière efficace. La version "x86_64" indique l'utilisation d'une architecture CPU 64 bits.
- **GPU :** Le GPU Tesla T4 est un composant haute performance conçu pour les tâches de calcul accéléré, couramment utilisé dans les centres de données et les plateformes comme Google Colab. Il offre une puissance de traitement parallèle significative, ce qui le rend idéal pour le deep learning, les simulations scientifiques et d'autres charges de travail intensives. Le Tesla T4 est équipé de Tensor Cores dédiés pour des performances optimisées en deep learning et prend en charge des frameworks comme TensorFlow, PyTorch et CUDA pour un calcul efficace sur GPU.

Google Drive : Google Drive est un service de stockage cloud et de synchronisation de fichiers fourni par Google. Il permet aux utilisateurs de stocker et d'accéder à des fichiers depuis n'importe où avec une connexion Internet. Les utilisateurs peuvent télécharger différents types de fichiers, les organiser dans des dossiers et les partager facilement pour la collaboration. Google Drive propose également des outils de productivité en ligne tels que Google Docs, Sheets et Slides pour créer et éditer des documents, des feuilles de calcul et des présentations. Il s'intègre avec d'autres services Google, offre un espace de stockage gratuit et assure un stockage sécurisé des fichiers dans le cloud, en faisant une solution pratique et polyvalente pour gérer et accéder aux fichiers.

Caméra de surveillance haute qualité : Une caméra de surveillance de haute qualité est un appareil électronique qui permet de prendre et de stocker des images et des vidéos dans un but de sécurité. Plusieurs éléments influencent la qualité de la caméra, tels que la résolution, la sensibilité à la lumière et les capacités de traitement de l'image. Prenons l'exemple d'une caméra haute résolution qui peut fournir une image plus claire, ce qui est essentiel pour identifier les individus ou les objets dans une zone surveillée.

ESP32 : L'ESP32 est un microcontrôleur de type SoC (System on Chip) conçu par Espressif Systems. Il est équipé de capacités Wi-Fi et Bluetooth intégrées, ce qui le rend idéal pour les applications de

l'Internet des objets (IoT). L'ESP32 est utilisé pour développer des dispositifs connectés grâce à ses performances élevées, sa faible consommation d'énergie, et ses nombreuses interfaces de communication [39].

Alimentation 5V : Une alimentation de 5 volts est une source d'énergie électrique qui fournit une tension constante de 5 volts. Elle est couramment utilisée pour alimenter des circuits électroniques et des microcontrôleurs comme l'Arduino et l'ESP32 [40].

Relais : Un relais est un dispositif électromécanique ou à semi-conducteurs utilisé pour ouvrir ou fermer un circuit électrique. Il permet de contrôler un circuit de puissance élevée en utilisant un signal de commande de faible puissance [41].

Écran LCD 2x16 : Un écran LCD 2x16 est un affichage à cristaux liquides capable de montrer deux lignes de texte, chaque ligne contenant jusqu'à 16 caractères. Ces écrans sont couramment utilisés dans des projets électroniques pour afficher des informations textuelles simples [42].

Câble électrique : Un câble électrique est un conducteur utilisé pour transporter l'électricité d'un point à un autre. Il est constitué de fils métalliques (souvent en cuivre ou en aluminium) isolés par un matériau non conducteur pour éviter les courts-circuits et les décharges électriques [43].

Alimentation 12V pour la barrière : Une alimentation de 12 volts est utilisée pour fournir l'énergie nécessaire au fonctionnement de systèmes tels que les barrières automatiques. Ces systèmes nécessitent une tension plus élevée pour faire fonctionner des moteurs ou des mécanismes électromécaniques [44].

Point d'accès Wi-Fi : Un point d'accès Wi-Fi est un dispositif qui permet de connecter des appareils sans fil à un réseau local (LAN) les modules ESP32 et à le serveur de base de données. Il agit comme une passerelle entre les appareils sans fil et le réseau câblé [45].

4.2.2 Environnement logiciel :

PHP : PHP (Hypertext Preprocessor) est un langage de script open-source côté serveur conçu principalement pour le développement web. Il est utilisé pour créer des pages web dynamiques et interactives [46].

MySQL : MySQL est un logiciel open-source de gestion de bases de données relationnelles. Il sert à conserver, structurer et récupérer des informations dans des applications en ligne. MySQL est connu pour ses résultats, sa fiabilité et sa convivialité [47].

Python : Python est un langage de programmation polyvalent et très répandu, employé dans différents secteurs tels que le développement web, l'analyse de données, l'apprentissage automatique et le calcul scientifique. Python 3, la version la plus récente, présente des améliorations en termes de syntaxe, de support de l'Unicode, de gestion de la mémoire et de performances. Python est distribué par la Python Software Foundation et dispose d'un large éventail de bibliothèques et de frameworks qui sont adaptés à diverses applications.



Figure 4.1 Python logo

Torch : Torch, ou PyTorch, est un framework d'apprentissage automatique open source principalement utilisé pour l'apprentissage profond. Il offre flexibilité et efficacité dans la construction et la formation de réseaux de neurones grâce à un graphique de calcul dynamique permettant des ajustements en temps réel. Torch est réputé pour son excellent support des GPU, ce qui le rend idéal pour les modèles d'apprentissage profond à grande échelle. Il propose des bibliothèques pour diverses tâches, simplifie la construction de modèles et permet une grande personnalisation. Torch est apprécié pour ses API conviviales, sa documentation complète et sa communauté active. Il est régulièrement mis à jour avec de nouvelles fonctionnalités et améliorations [35] [36].

OpenCV : OpenCV est une bibliothèque open source polyvalente spécialisée dans la vision par ordinateur et l'ingénierie automatique. Elle supporte différentes langues de programmation et offre des fonctionnalités pour des opérations comme le traitement d'images, la reconnaissance d'objets et la calibration de caméras. Afin de s'adapter aux applications en temps réel, OpenCV propose également des modules d'apprentissage automatique. Elle est couramment employée dans des secteurs tels que la robotique, la réalité augmentée, la surveillance, et bien plus encore. OpenCV est accessible gratuitement et est compatible avec la licence BSD. [37]

Importation des bibliothèques :

- **cv2 (OpenCV)** : OpenCV est une bibliothèque de vision par ordinateur open-source. Elle offre une large gamme de fonctionnalités pour la capture, le traitement et l'analyse d'images et de vidéos en temps réel. OpenCV est largement utilisée dans les applications de vision par ordinateur pour la détection d'objets, la reconnaissance faciale, le suivi de mouvement, la segmentation d'image, et bien plus encore. Elle est écrite en C++ mais possède des bindings pour Python (cv2), ce qui la rend populaire pour les développeurs Python travaillant sur des projets de traitement d'images.
- **pymysql** : PyMySQL est une bibliothèque Python qui permet la connexion et l'interaction avec une base de données MySQL. Elle implémente un client MySQL écrit entièrement en Python, facilitant ainsi la gestion des bases de données MySQL directement depuis des scripts Python. PyMySQL est simple à utiliser et bien documenté, ce qui en fait un choix courant pour les applications nécessitant une intégration transparente avec des bases de données MySQL.
- **ultralytics (YOLO)** : YOLO (You Only Look Once) est une série de modèles d'apprentissage profond qui permettent de détecter des objets en temps réel. Dans cette situation, Ultralytics désigne une application particulière ou une version de YOLO créée par Ultralytics. Le fonctionnement de YOLO repose sur la division de l'image en une grille et la prédiction simultanée des boîtes englobantes ainsi que des probabilités de classe pour ces boîtes. Cela offre la possibilité de détecter rapidement et efficacement des objets dans des images ou des flux vidéo en direct. Cette bibliothèque connaît un vif succès dans les domaines de la surveillance vidéo, de la sécurité et de la vision numérique embarquée.
- **sys** : sys est un module Python intégré qui fournit un accès à certaines variables utilisées ou maintenues par l'interpréteur Python et à des fonctionnalités qui interagissent fortement avec l'environnement système. Il est couramment utilisé pour gérer les arguments de la ligne de

commande passés à un script Python, par exemple pour obtenir des options ou des paramètres spécifiés lors de l'exécution du script.

- **time** : time est un module Python intégré qui fournit diverses fonctions liées à la mesure du temps. Il est utilisé pour contrôler le temps d'exécution des scripts, pour introduire des délais, ou pour mesurer la durée d'exécution des différentes parties d'un programme. Cela permet de synchroniser des actions, de gérer les événements dans le temps, ou de calculer la durée de traitement des opérations.
- **PyTesseract** : PyTesseract est une bibliothèque Python spécialisée dans la reconnaissance optique de caractères (OCR). Elle agit comme une interface pour Tesseract-OCR, un moteur avancé développé initialement par HP Labs et maintenu par Google AI. PyTesseract permet de lire et d'interpréter du texte à partir d'images ou de vidéos, en convertissant le contenu textuel présent dans les médias visuels en texte brut manipulable par des scripts Python [38].

4.3 Training and validation

Nous avons réalisé un entraînement et validation. Et voici quelques résultats et métriques :

4.3.1 résultats de l'entraînement :

Il s'agit d'échantillons aléatoires de 16 lots d'ensembles d'entraînement et de validation :

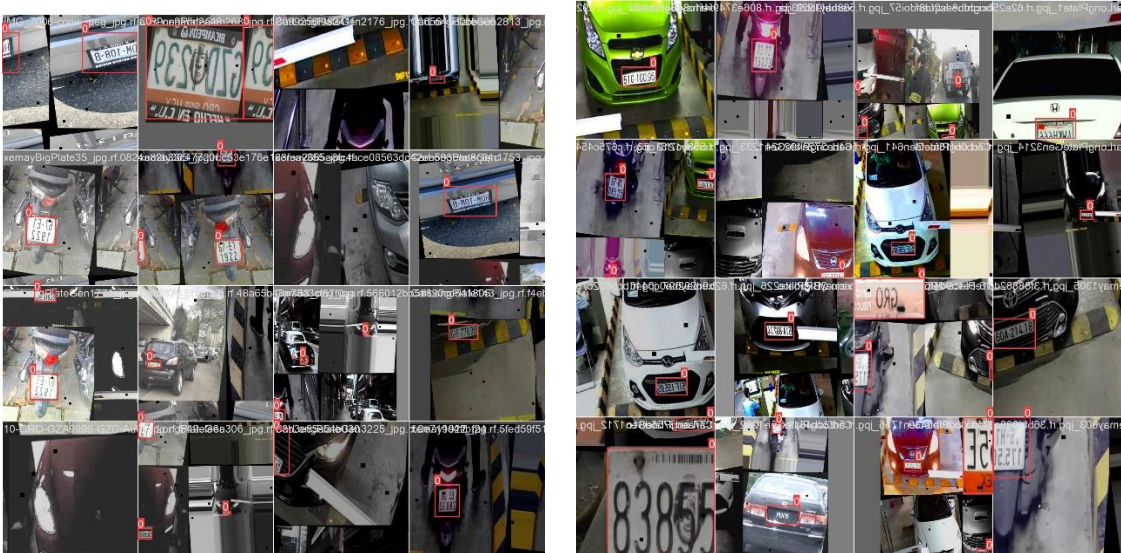


Figure 4.2 Train batches

La méthode **Train Batches** consiste à former un modèle de détection de plaques d'immatriculation en divisant les données en plusieurs sous-ensembles ou **lots**, permettant ainsi une gestion plus efficace des ressources et une accélération du processus d'apprentissage.



Figure 4.3 validation batches

Pendant l'entraînement, à intervalles réguliers, le modèle est évalué sur les lots de validation pour vérifier sa performance. Cela aide à ajuster les hyperparamètres et à prévenir l'overfitting en surveillant la capacité du modèle à généraliser sur des données non utilisées pour l'entraînement.

Métriques train/val

Ces résultats sont alors que le train de 15 époques

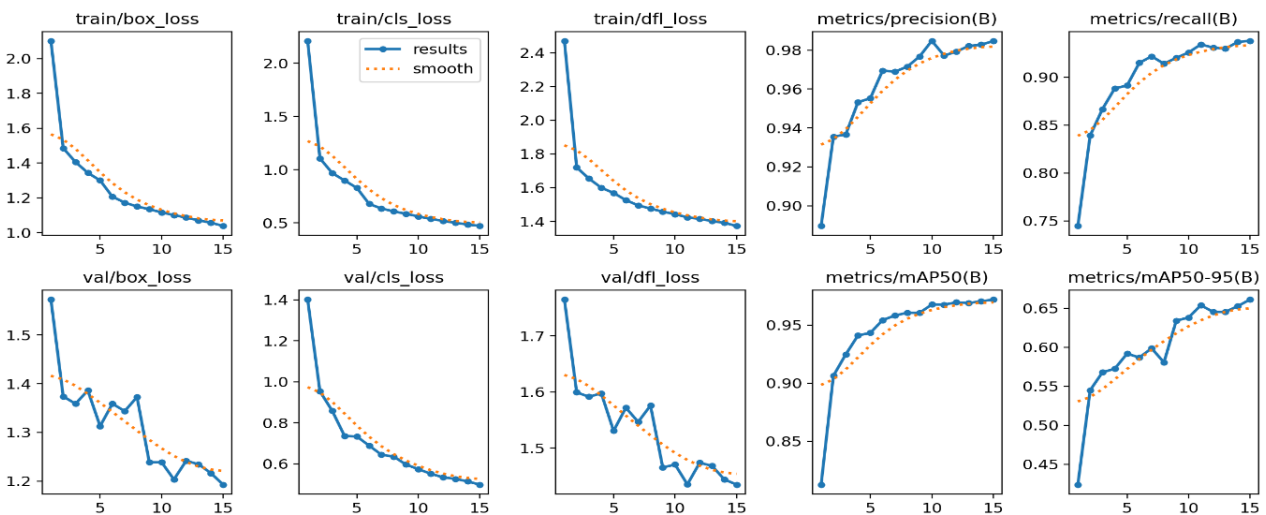


Figure 4.4 Résultat

Ces résultats montrent l'évolution des pertes et des métriques au cours de 15 époques d'entraînement et de validation pour un modèle de détection du plaques d'immatriculation :

1. **Pertes d'entraînement** : Les courbes des pertes de boîte (box_loss), de classification (cls_loss), et de régression (dfl_loss) diminuent régulièrement, indiquant que le modèle s'améliore.
2. **Pertes de validation** : Les pertes de validation (val/box_loss, val/cls_loss, val/dfl_loss) diminuent également, bien que de manière plus fluctuante, suggérant que le modèle généralise bien sur les données de validation.
3. **Métriques de précision** : La précision, le rappel, et les scores de mAP (Mean Average Precision) augmentent avec les époques, montrant une meilleure performance globale du modèle au fil du temps.

Ces graphiques indiquent que le modèle progresse de manière positive dans l'apprentissage des caractéristiques pour la tâche de détection des plaques d'immatriculation.

F1_Curve : "La F1_Curve est une représentation graphique essentielle en apprentissage automatique, illustrant la précision harmonique (F1) pour toutes les classes à un seuil de 0.451, où elle atteint une valeur significative de 0.96."

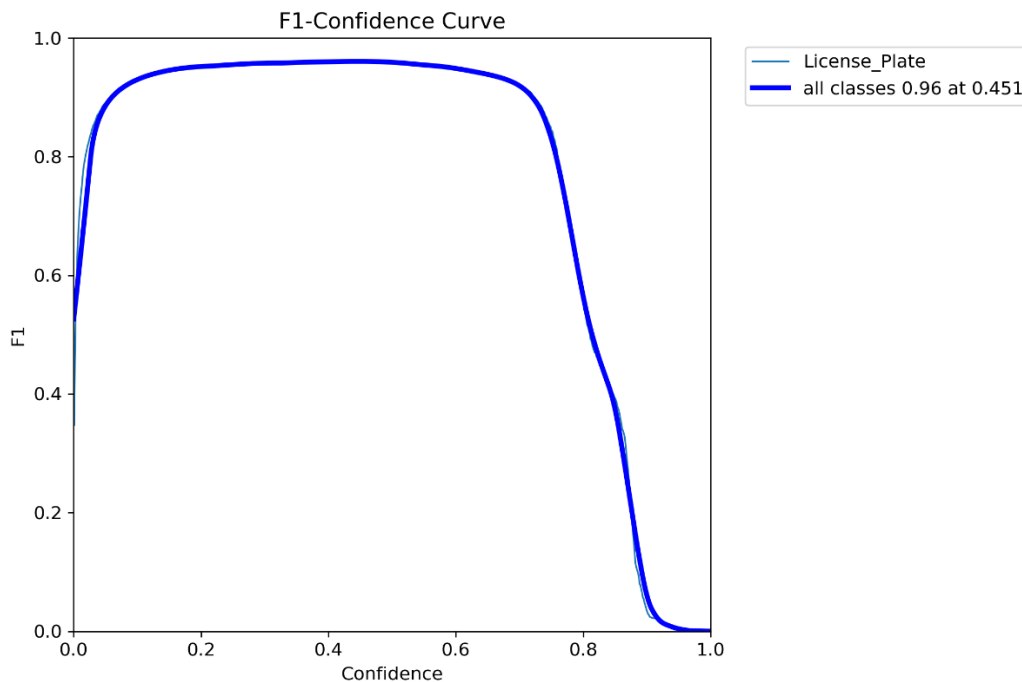


Figure 4.5 F1_Curve (F1)

P_curve : "La P_curve présente un rappel parfait (1.0) pour toutes les classes à un seuil de 0.832, soulignant une sensibilité optimale dans la détection des données."

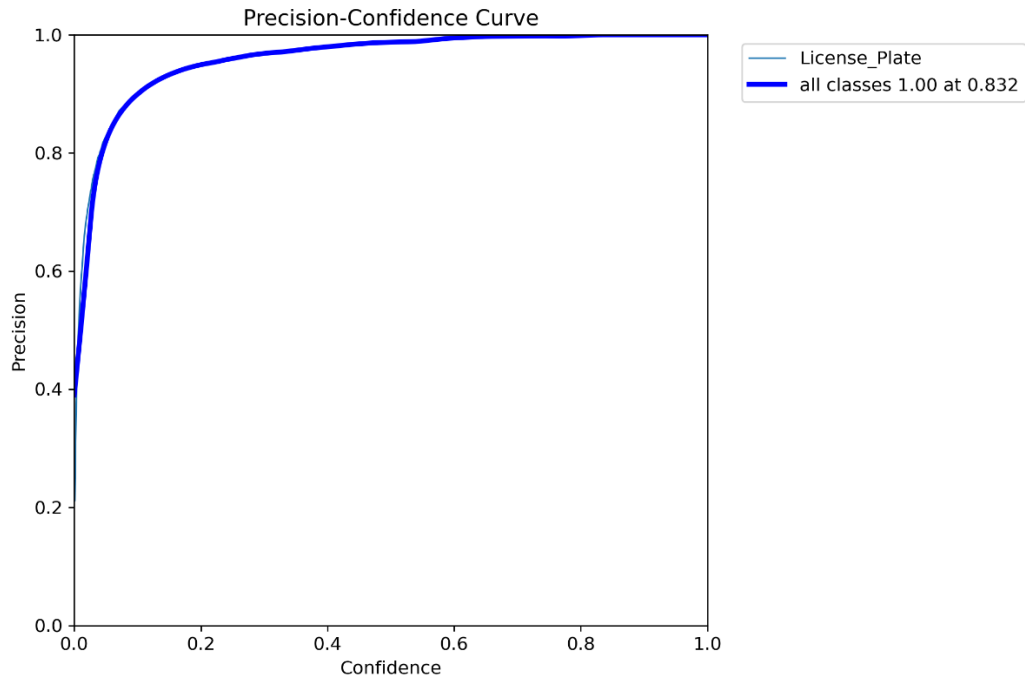


Figure 4.6 P_curve (Precision)

PR_curve: "La PR_curve montre une précision-recall moyenne (mAP) de 0.972 pour toutes les classes à un seuil de 0.5, soulignant une performance élevée dans l'évaluation de la précision et du rappel."

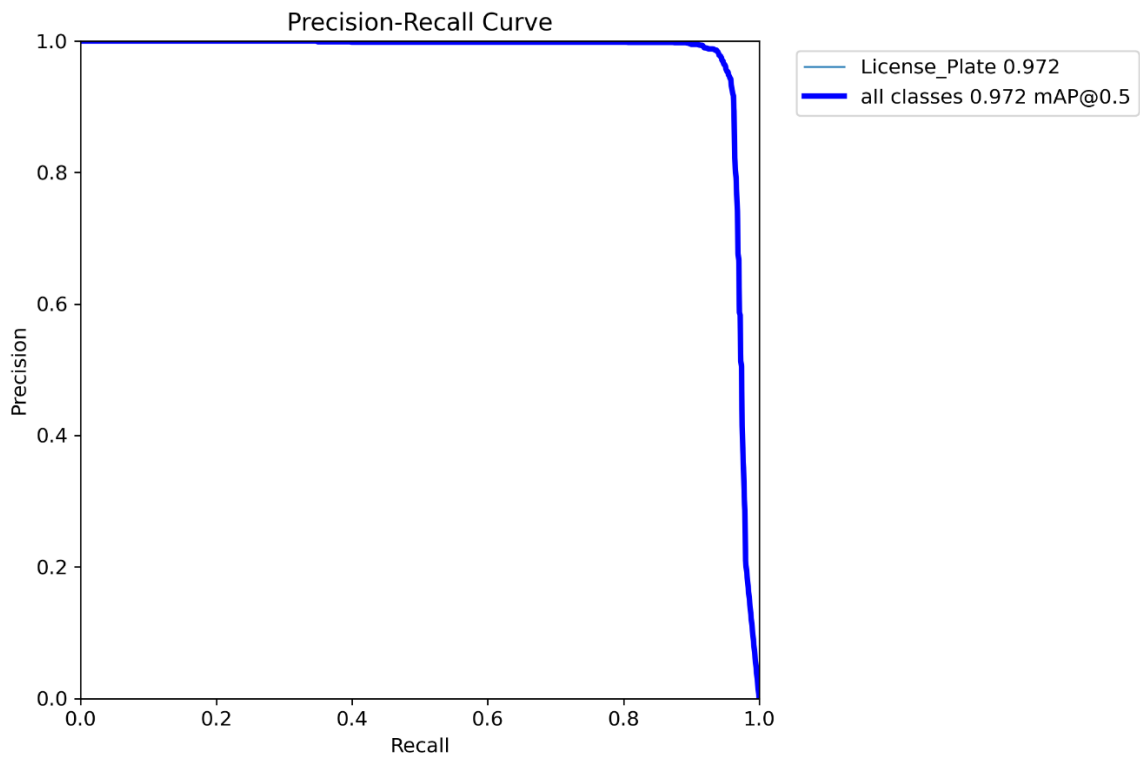


Figure 4.7 PR_curve (Precision_recall)

R_curve : "La R_curve indique un rappel élevé de 0.98 pour toutes les classes à un seuil de 0.0000, mettant en évidence une excellente capacité du modèle à détecter efficacement les instances dans les données."

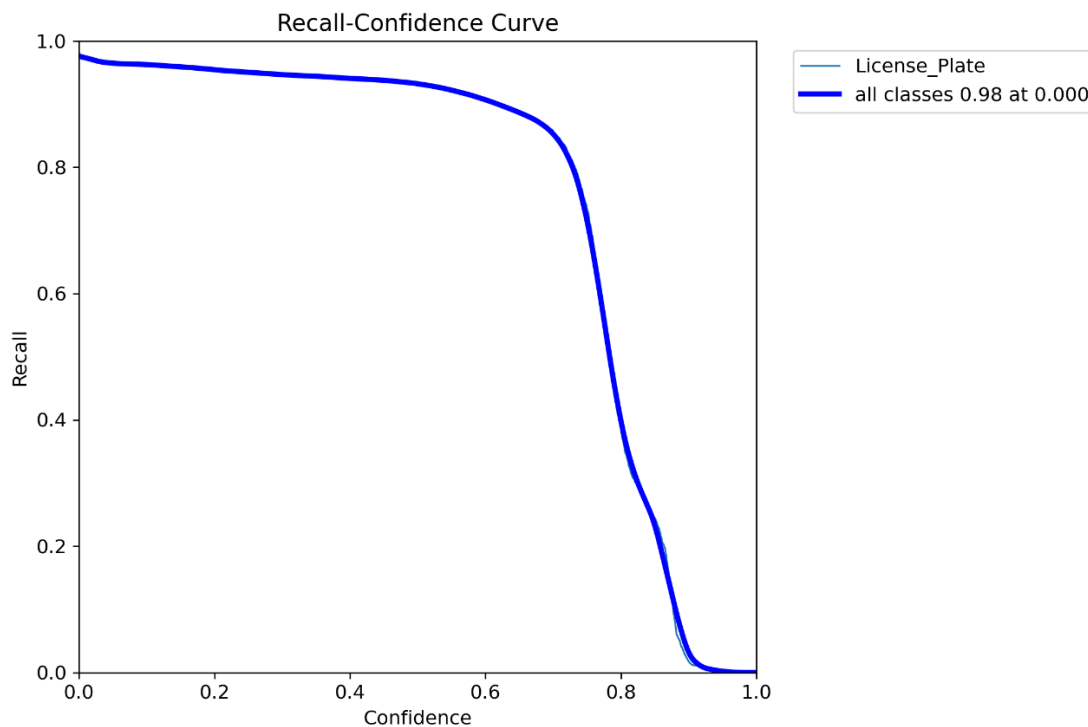


Figure 4.8 R_curve (Recall)

Epoch	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)
1	88.962	0.74469	0.81265	0.4238
2	93.556	0.83912	0.90639	0.5448
3	93.663	0.86679	0.92471	0.56778
4	95.325	0.88837	0.94111	0.57242
5	95.526	0.89138	0.94326	0.59167
6	96.954	0.9151	0.95428	0.58686
7	96.894	0.92186	0.95821	0.59853
8	97.151	0.91417	0.96063	0.5805
9	97.672	0.92026	0.96054	0.63398
10	98.481	0.92589	0.96768	0.63805
11	97.72	0.93433	0.96758	0.65372

12	97.928	0.931	0.96961	0.64528
13	98.216	0.92984	0.9691	0.64532
14	98.278	0.93693	0.97034	0.65261
15	98.469	0.93809	0.97203	0.66115

Tableau 4.1 résultats 01

Explication des Résultats

1. **Précision** : La précision (**metrics/precision(B)**) augmente régulièrement de 88.96% à 98.47%. Cela indique que le modèle devient de plus en plus précis pour détecter correctement les plaques sans fausses alarmes.
2. **Rappel** : Le rappel (**metrics/recall(B)**) passe de 74.47% à 93.81%, montrant que le modèle s'améliore également pour détecter plus de plaques existantes, réduisant les faux négatifs.
3. **mAP50** : La métrique **mAP50(B)**, qui évalue la précision moyenne à un seuil d'Intersection over Union (IoU) de 50%, passe de 81.27% à 97.20%. Cela démontre que la capacité du modèle à localiser les plaques avec une bonne précision est excellente.
4. **mAP50-95** : La métrique **mAP50-95(B)**, plus stricte, qui évalue la précision moyenne à plusieurs seuils d'IoU de 50% à 95%, progresse de 42.38% à 66.12%. Cette amélioration montre que le modèle devient également meilleur pour des localisations plus précises.

Interprétation

- **Progression constante** : Les augmentations régulières dans les quatre métriques au fil des époques montrent une amélioration constante des performances du modèle.
- **Stable après certaines époques** : La précision et les métriques mAP tendent à se stabiliser après certaines époques, suggérant une convergence du modèle.
- **Optimisation réussie** : La progression significative du rappel et des mAP indique que le modèle est bien optimisé pour détecter et localiser les plaques d'immatriculation avec une haute précision.

4.3.2 résultats de l'entraînement de modèle OCR :

Il s'agit d'échantillons aléatoires de 16 lots d'ensembles d'entraînement et de validation :



Figure 4.9 Train batches

Cette figure illustre comment un modèle OCR détecte et reconnaît les caractères sur des plaques d'immatriculation synthétiques, en les entourant de cadres colorés. Les couleurs et les numéros indiquent les détections et classifications réalisées par le modèle. La diversité des images, avec différentes polices et angles, aide à généraliser le modèle pour diverses conditions. Les annotations visuelles servent à valider la qualité des détections et l'exactitude de la reconnaissance des caractères. Ce processus est crucial pour former un modèle capable de lire précisément les plaques d'immatriculation en conditions réelles.



Figure 4.10 validation batches

La figure montre les résultats de la validation batch pour le modèle OCR de reconnaissance de plaques d'immatriculation. À gauche (L), les caractères sont bien détectés et reconnus, indiquant des validations réussies. À droite (P), les erreurs de détection et de reconnaissance apparaissent, révélant des problèmes tels que des caractères mal classifiés ou manquants. Les annotations colorées autour

des caractères montrent les performances et les erreurs du modèle. Ces validations permettent d'identifier les points à améliorer pour une meilleure précision du modèle.

Métriques train/val

Ces résultats sont alors que le train de 10 époques

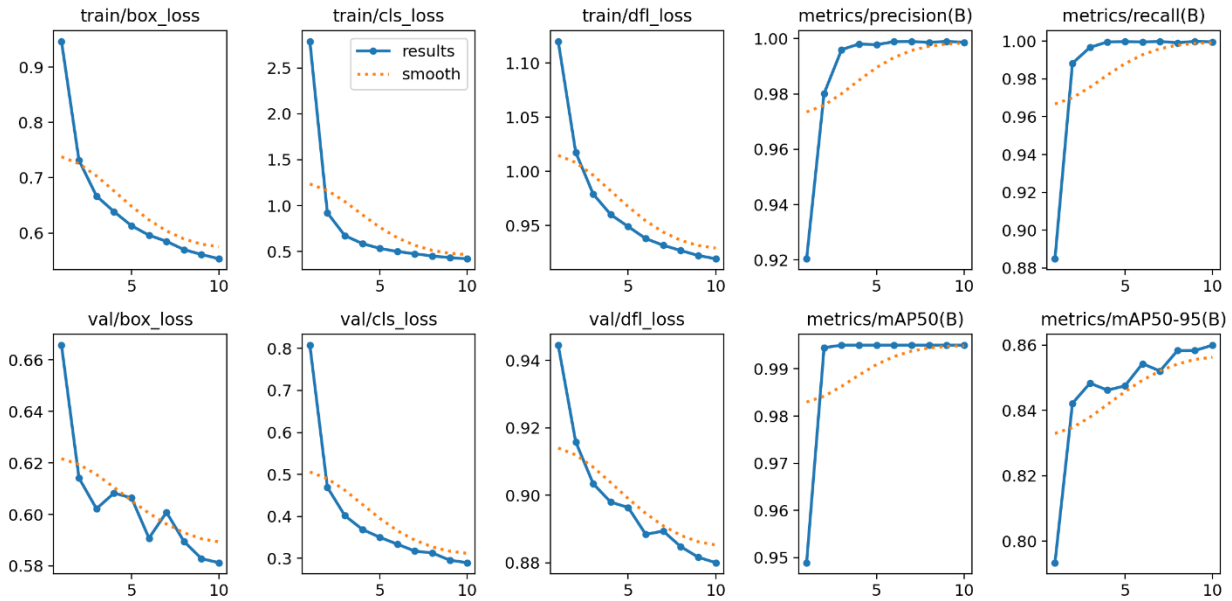


Figure 4.11 Résultat OCR

R_curve : "La R_curve indique un rappel élevé de 1.00 pour toutes les classes à un seuil de 0.0000, mettant en évidence une excellente capacité du modèle à détecter efficacement les instances dans les données."

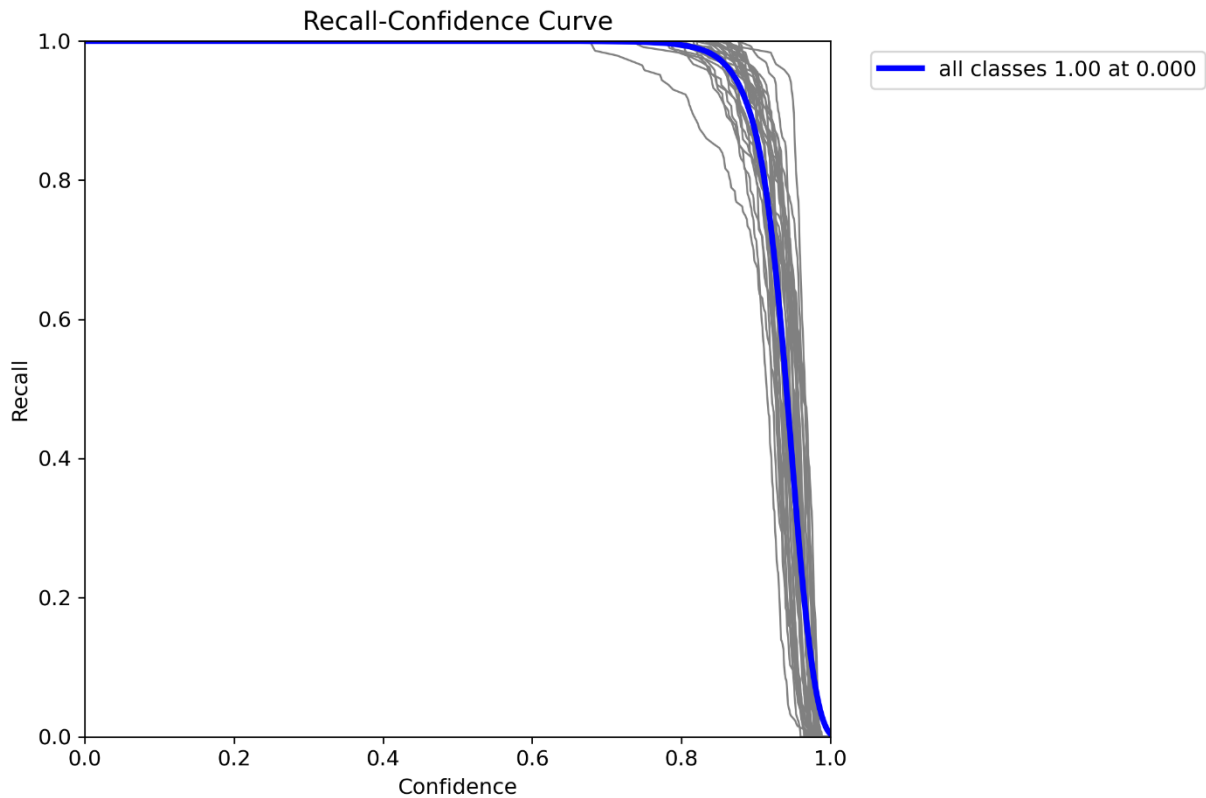


Figure 4.12 R_curve (Recall)

Epoch	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)
1	0.92042	0.8849	0.94893	0.79339
2	0.98014	0.9883	0.99448	0.84218
3	0.99599	0.99668	0.995	0.84833
4	0.99804	0.99957	0.995	0.84619
5	0.99776	0.99974	0.995	0.84752
6	0.99888	0.99955	0.995	0.8543
7	0.99894	0.99983	0.995	0.85206
8	0.99855	0.99907	0.995	0.8583
9	0.999	0.9999	0.995	0.85833
10	0.9986	0.99961	0.995	0.85996

Tableau 4.2 résultats 02

Ces résultats montrent que le modèle OCR atteint des niveaux très élevés de précision et de rappel après seulement quelques époques, avec des valeurs de metrics/precision(B) et metrics/recall(B) approchant 100%. Cependant, metrics/mAP50(B) reste stable autour de 99.5% après l'époque 1, et metrics/mAP50-95(B) n'augmente que légèrement, suggérant que le modèle pourrait sur-apprendre les données d'entraînement. La faible progression de mAP50-95 indique que le modèle peine à généraliser pour des localisations plus précises en validation, malgré ses performances élevées sur les données d'entraînement.

4.4 Test, Résultat et Discussion

4.4.1 Test et Résultat

Le résultat avec le meilleur poids « best.pt » de model détection du matricule :



Figure 4.13 détection et cadrer le matricule

Le résultat avec la fonction PyTesseract

```
0: 480x640 1 truck, 906.0ms
Speed: 15.6ms preprocess, 906.0ms inference, 15.6ms postp

0: 480x640 1 license_plate, 529.4ms
Speed: 15.6ms preprocess, 529.4ms inference, 0.0ms postp

License Plate Image Shape: (93, 338)
License Plate Text: 2477380016
```

Le résultat après la vérification avec la base de données, retourne une acceptation ou un refus d'accès.

```
>>> | La plaque se trouve dans la base de données.
```

4.4.2 Discussion

Après avoir analysé les différentes options de formation et les résultats obtenus, plusieurs observations significatives se dégagent :

Précision dans l'identification de la plaque d'immatriculation

L'objectif principal de notre système était d'améliorer la précision dans l'identification des plaques d'immatriculation des véhicules. Nous avons observé une augmentation substantielle de la précision au fil des itérations d'entraînement. Cette amélioration est clairement démontrée à travers nos analyses graphiques, où nous avons pu visualiser une convergence vers des résultats plus précis au cours du processus d'optimisation du modèle. Comparativement aux travaux précédents dans ce domaine, nos résultats se démarquent par une précision accrue et une capacité améliorée à détecter et à lire les plaques d'immatriculation dans des conditions variées, y compris des environnements lumineux et météorologiques divers.

Lecture des numéros de plaque d'immatriculation

Initialement, lors de la phase de validation, nous avons rencontré des défis significatifs concernant la lecture précise des numéros de plaque d'immatriculation. Les résultats d'entraînement ne correspondaient pas aux attentes, signalant la nécessité d'améliorations supplémentaires. Cependant, grâce à l'intégration de la fonctionnalité Pytesseract, spécialisée dans la reconnaissance optique de caractères (OCR), nous avons pu surmonter ces difficultés. Pytesseract a joué un rôle crucial en améliorant de manière significative la précision de la lecture des numéros de plaque, transformant ainsi des résultats initialement médiocres en performances très satisfaisantes. Cette approche a permis de corriger les erreurs de lecture et d'optimiser le processus de traitement des images capturées par nos caméras de surveillance.

Comparaison avec les plaques d'immatriculation de la base de données

Un aspect essentiel de notre système est sa capacité à comparer efficacement les plaques d'immatriculation détectées avec celles enregistrées dans notre base de données. Cette fonctionnalité

est cruciale pour déterminer si un véhicule est autorisé à accéder à une zone restreinte ou à un service spécifique. Notre méthode de comparaison, intégrée au flux de traitement des données, offre une décision rapide et précise quant à l'acceptation ou au rejet d'un véhicule. Cette approche contribue à renforcer la sécurité et la gestion des accès, tout en réduisant les risques d'erreurs et de faux positifs.

4.5 Conclusion

Dans ce chapitre, nous avons décrit l'implémentation d'un système de détection des plaques d'immatriculation des véhicules utilisant YOLOv8. Ce système est intégré au contrôle d'accès des péages autoroutiers. Nous avons commencé par présenter l'environnement de développement et les outils logiciels utilisés, ainsi que les tests et les résultats du processus d'entraînement. En outre, nous avons mené une discussion détaillée sur les performances du système.

En conclusion, les résultats obtenus reflètent des avancées significatives dans le développement de notre système de reconnaissance de plaques d'immatriculation. Nous avons non seulement amélioré la précision globale de l'identification des plaques d'immatriculation, mais nous avons également introduit des solutions efficaces pour surmonter les défis rencontrés lors de la lecture des numéros de plaque. En intégrant des technologies telles que Pytesseract et en optimisant nos processus de comparaison avec la base de données, nous avons jeté les bases d'un système robuste et fiable pour la gestion des accès et la sécurité.

Pour l'avenir, nous envisageons d'explorer davantage les possibilités d'amélioration continue, notamment en affinant nos modèles d'apprentissage automatique et en explorant de nouvelles techniques de traitement d'image avancées. Ces efforts visent à renforcer encore les performances de notre système et à répondre aux exigences croissantes en matière de sécurité et de gestion des flux de véhicules dans divers environnements.

Conclusion Générale

Ce mémoire a exploré le développement d'un système avancé de contrôle d'accès pour les péages autoroutiers, intégrant la reconnaissance automatique des plaques d'immatriculation et les technologies IoT, avec une emphase particulière sur l'utilisation du modèle de Deep Learning YOLOv8. L'objectif principal de ce projet était de concevoir une solution efficace, évolutive et sécurisée pour améliorer la fluidité du trafic et renforcer la sécurité des infrastructures routières.

Nous avons démontré que le modèle YOLOv8, grâce à son architecture optimisée pour la détection et la reconnaissance d'objets en temps réel, permet une identification précise et rapide des plaques d'immatriculation. En entraînant ce modèle sur des ensembles de données diversifiés et en réalisant des benchmarks pour les plaques d'immatriculation et les numéros de matricules, nous avons assuré une robustesse et une fiabilité accrues du système de reconnaissance.

L'intégration des technologies IoT a joué un rôle crucial en permettant une gestion et une surveillance en temps réel des opérations de péage, offrant une connectivité optimale entre les différents composants du système. Cela a conduit à une réduction significative des temps d'attente, à une diminution des erreurs humaines et des fraudes, ainsi qu'à une amélioration globale de l'efficacité et de la sécurité des processus de péage.

Les résultats obtenus montrent que le système proposé répond efficacement aux défis posés par les méthodes traditionnelles de collecte de péages. En combinant les avancées en Deep Learning et en IoT, ce projet ouvre la voie à des infrastructures de péage autoroutier plus intelligentes, performantes et sécurisées. Cette recherche contribue ainsi de manière significative à l'avancement des systèmes de contrôle d'accès intelligents, tout en soulignant l'importance d'une mise en œuvre éthique et responsable des technologies innovantes.

En conclusion, ce travail démontre que les solutions basées sur l'IA et l'IoT ont le potentiel de transformer les systèmes de péage autoroutier, offrant des avantages considérables en termes de fluidité du trafic, de sécurité et de satisfaction des usagers. Les perspectives d'application de ces technologies sont vastes et prometteuses, et leur adoption généralisée pourrait révolutionner la gestion des infrastructures routières à l'avenir.

Bibliographies

- [1]: European Parliament, & Council of the European Union. (2008). Directive 2008/96/EC of the European Parliament and of the Council of 19 November 2008 on road infrastructure safety management. *Official Journal of the European Union*, L 319, 59-67. Retrieved from <http://data.europa.eu/eli/dir/2008/96/oj>
- [2]: Shafiee, S., Mahmoodian, M., & Aghamohammadi, M. (2014). A review of electronic toll collection systems. *Transportation Research Part C: Emerging Technologies*, 48, 32-47. https://www.researchgate.net/publication/350106626_A_SYSTEMATIC_REVIEW_OF_ELECTRONIC_TOLL_COLLECTION_SYSTEMS
- [3]: Tiwari, S., & Joshi, M. (2022). Automatic toll collection system using ANPR and RFID: A review. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(2), 453-462. <https://www.ijfmr.com/papers/2024/3/19354.pdf>
- [4]: Sun, Z., Wang, Y., & Zhang, H. (2021). License plate recognition based automatic toll collection system for expressway. In *2021 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2134-2139). IEEE. <https://ieeexplore.ieee.org/document/10266114>
- [5]: Russell, S. J., Norvig, P. (2010). *Artificial intelligence: a modern approach*. Pearson Education. <http://repo.darmajaya.ac.id/3800/1/Artificial%20Intelligence%20A%20Modern%20Approach%20%283rd%20Edition%29.pdf%20%28%20PDFDrive%20%29.pdf>. consulté le :01/02/2024
- [6] Alpaydm, Ethem. (2010). *Introduction to Machine Learning* (2nd ed.). The MIT Press: Cambridge, Massachusetts; London, England. ISBN 978-0-262-01243-0.
- [7]: Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- [8]: Mitchell, T. (1997). *Machine Learning*. McGraw Hill. <http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture7.pdf>. consulté le : 2023-12-20.
- [9]: Deep Learning with Graph Convolutional Networks: An Overview and Latest Applications in Computational Intelligence, Uzair Aslam Bhatti, Hao Tang, Guilu Wu, Shah Marjan, Aamir Hussain. <https://www.hindawi.com/journals/ijis/2023/8342104/>. consulté le : 2023-12-20.
- [10]: A Survey of Trendy Financial Sector Applications of Machine and Deep Learning, Nur Indah Lestari, Walayat Hussain, Jose M. Merigo Mahmoud Bekhit . Accessed: 2023-04-20.
- [11]: Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458, 2015.
- [12]: Robin M Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. arXiv preprint arXiv:1912.05911, 2019.
- [13] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1-42, 2021.

- [14]:MS COCO (Microsoft Common Objects in Context)dataset. <https://cocodataset.org/>. consulté le:2024-01-30.
- [15]:PASCAL VOC (ual Object Classes))dataset. <http://host.robots.ox.ac.uk/pascal/VOC/>. consulté le :2024-02-30.
- [16] Open Images Dataset V7dataset. <https://storage.googleapis.com/openimages/web/index.html>. consulté le :2024-01-02.
- [23] : Barbaresso, J., Goodchild, M., & Funderburg, R. (1993). "The effects of toll plazas on traffic flow." *Transportation Research Record*, 1406, 8-16.
- [24] : Goodman, P., Sussman, J., & Boyce, D. (2015). "Evaluation of the E-ZPass electronic toll collection system." *Journal of Transportation Engineering*, 141(4), 04014090.
- [29] : Liu, W., Anguelov, D., Erhan, D., Szegedy, C., & Reed, S. (2016). "SSD: Single shot multibox detector." *European conference on computer vision (ECCV)*, 21(12), 21-37.
- [30]: Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2980-2988 .
- [31]: Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 779-788.
- [32]: Kumar, A., & Verma, A. (2017). "Automatic Number Plate Recognition System using OCR and RFID." *International Journal of Advanced Research in Computer Science*, 8(5), 148-151.
- [33] : Zhou, Q., Zhang, Y., & Shi, Y. (2018). "RFID-based vehicle management system for toll station." *2018 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 187-190.
- [34] : Fernandez-Carames, T. M., & Fraga-Lamas, P. (2018). "A Review on the Use of Blockchain for the Internet of Things." *IEEE Access*, 6, 32979-33001.
- [38] Smith, Ray. "An overview of the Tesseract OCR engine." *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. Vol. 2. IEEE, 2007.
- [39] Espressif Systems. (2024). ESP32 Technical Reference Manual, Version 5.1. Retrieved from <https://www.espressif.com/en/support/download/documents>
- [40] Kuffel, E., Zaengl, W. S., Kuffel, J., & Zaengl, W. S. (2000). *High Voltage Engineering* (2nd ed.). Newnes
- [41] Slade, P. G. (Ed.). (2017). *Electrical Contacts: Principles and Applications, Second Edition*. CRC Press. ISBN 9781138077102.
- [42] Lien, A., & Chen, C. H. (1997). *Liquid Crystal Displays: Fundamental Physics and Technology*. Wiley.
- [43] Moore, G. F. (Ed.). (1997). *Electric Cables Handbook* (3rd ed.). Wiley-Blackwell. ISBN 978-0632040759.

- [44] Short, T. A. (2014). *Electric power distribution handbook*. Boca Raton: CRC Press, Taylor & Francis Group. ISBN 978-1466598652.
- [45] "Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice* (2nd ed.). Prentice Hall.
- [46] Tatroe, K., MacIntyre, P. V. A., & Lerdorf, R. (2013). *Programming PHP*. O'Reilly Media.
- [47] Matthew, V. K., & McDonald, M. W. (2003). *MySQL: The Complete Reference*. McGraw-Hill Osborne Media.
- [48] Simonite, T. (2018). Artificial intelligence vs. machine learning vs. deep learning vs. neural networks: What's the difference? *MIT Technology Review*. Retrieved from <https://www.technologyreview.com/2018/11/17/103781/artificial-intelligence-machine-learning-deep-learning-explained/>
- [52] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd Edition). MIT Press.
- [80]Smith, R. (2007). "An Overview of the Tesseract OCR Engine." Proc. of Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 629-633.
- [55]Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Chap. 9
- [53]Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788.
- [54]Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems, 28, pp. 91-99.
- [56] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). "Simple Online and Realtime Tracking." 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464-3468.
- [57] Danelljan, M., Robinson, A., Khan, F. S., & Felsberg, M. (2016). "Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking." European Conference on Computer Vision (ECCV), pp. 472-488.
- [81]LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning." *Nature*, 521(7553), pp. 436-444.
- [58]Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems, 25, pp. 1097-1105.
- [61]Vinyals, O., & Le, Q. V. (2015). "A Neural Conversational Model." arXiv preprint arXiv:1506.05869.
- [62]Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). "Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models." Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 3776-3784.

- [59] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 4171-4186.
- [60] Brown, T. B., Mann, B., Ryder, N., et al. (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems, 33, pp. 1877-1901.
- [63] Simonyan, K., & Zisserman, A. (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition." International Conference on Learning Representations (ICLR).
- [64] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). "Rethinking the Inception Architecture for Computer Vision." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818-2826.
- [65] He, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.
- [66] Chollet, F. (2017). "Xception: Deep Learning with Depthwise Separable Convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251-1258.
- [67] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). "Densely Connected Convolutional Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708.
- [68] Szegedy, C., Liu, W., Jia, Y., et al. (2015). "Going Deeper with Convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9.
- [69] Howard, A. G., Zhu, M., Chen, B., et al. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv preprint arXiv:1704.04861.
- [70] Tan, M., & Le, Q. V. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 6105-6114.
- [72] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587.
- [73] Girshick, R. (2015). "Fast R-CNN." Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448.
- [74] Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." Advances in Neural Information Processing Systems, 28, pp. 91-99.
- [75] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). "Mask R-CNN." Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2961-2969.

- [76] Tan, M., Pang, R., & Le, Q. V. (2020). "EfficientDet: Scalable and Efficient Object Detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10781-10790.
- [77] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.
- [90] Hamzaoui, A., & Laour, H. (2023). Management of Toll Collection on Algerian Highways: Case Study of Lakhdaria. Springer, *Advances in Intelligent Systems and Computing*, 1313, 185-196.
- [91] Brahimi, H., & Belkacem, D. (2022). Implementation of Toll Systems in the Algerian Highway Network: The El Harrach Case. *Lecture Notes in Civil Engineering*, 122, 345-359.
- [92] Benyahia, K., & Chérif, M. (2022). Analysis of Toll Collection Efficiency on the Algerian Highway Network: Focus on Chlef. Springer, *Studies in Systems, Decision and Control*, 368, 143-155.
- [93] Toumi, A., & Boukhalfa, K. (2023). Challenges in Toll Collection for Highways in Algeria: Constantine as a Case Study. *Lecture Notes in Intelligent Transportation Systems*, 84, 123-136.
- [94] Rahmani, A., & Bensaid, S. (2023). Oran Toll System: Optimization and Implementation. Springer, *Intelligent Systems Reference Library*, 178, 245-258.
- [203] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.
- [204] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 and beyond, 2023
- [205] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.
- [222] Range King. Brief summary of yolov8 model structure 189. <https://github.com/ultralytics/ultralytics/issues/189>. consulté le: 2024-01-28.
- [221] Francesco Jacob Solawetz. What is yolov8? the ultimate guide. <https://blog.roboflow.com/whats-new-in-yolov8/>. consulté le: 2024-01-28.

Web Bibliographies

- [17] <https://ieeexplore.ieee.org/document/10307008> consulté le 12/01/2023.
- [18] <https://ieeexplore.ieee.org/document/10307008> consulté le 12/01/2023
- [19] <https://ieeexplore.ieee.org/document/9752632> consulté le 12/03/2024.
- [20] <https://ieeexplore.ieee.org/document/8322837> consulté le 30/03/2024.
- [21] <https://ieeexplore.ieee.org/document/9752632> consulté le 12/03/2024.

- [22] <https://ieeexplore.ieee.org/document/9752632> consulté le 12/03/2024.
- [35] Pytorch. <https://pytorch.org/>. consulté le: 2024-01-28.
- [36] Pytorch. <https://github.com/pytorch/pytorch>. consulté le: 2024-01-28.
- [37] Opencv. <https://opencv.org/>. consulté le: 2024-01-28
- [49] <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
- [50] https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents
- [51] <https://aws.amazon.com/fr/what-is/gan/>