جامعة سيدي محمد بن عبد الله بفاس
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⵎⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ ⴱ ⴼⴰⵙ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم ظهر المهراز فاس
ⵜⴰⵖⵡⵍⵍⵉⵜ ⵏ ⵜⵎⵙⵙⵓⵔⵉⵜ ⵙⵙⵉⵔⵉⵙ ⵎⵓⵃⵎⴷ ⵀⵎⵓⵔⴰⵣ ⴼⴰⵙ
FACULTE DES SCIENCES DHAR EL MAHRAZ FES

N° d'ordre : 114/ 2023

Centre des Etudes : ***Sciences et Techniques et Sciences Médicales***

Formation Doctorale : ***Sciences et Technologies de l'Information et de la Communication (STIC)***

Discipline : ***Physique***
Spécialité : ***Génie électrique***
Laboratoire : ***Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme (LISAC)***

THÈSE DE DOCTORAT

Vers une Vidéosurveillance Intelligente : Développement et Intégration d'Algorithmes de Deep Learning sur des Plateformes Embarquées

Présentée par :

ELMANAA IMANE

Soutenue le 29/12/2023 devant le jury :

Nom et Prénom	Grade	Etablissement	Qualité
Ali YAHYAOUY	PES	FSDM, USMBA, Fès	Président
FATIMA EL KHOUKHI	PES	FLSH, UMI, Meknès	Rapporteur & examinateur
EL MEHDI ISMAILI ALAOUI	PH	FS, UMI, Meknès	Rapporteur & examinateur
RACHID EL ALAMI	PH	FSDM, USMBA, Fès	Rapporteur & examinateur
EL HABIB NFAOUI	PES	FSDM, USMBA, Fès	Examinateur
YASSINE ABOUCH	PDG	NEXTRONIC, ABA Technology	Invité
AARAB ABDELLAH	PES	FSDM, USMBA, Fès	Co-directeur de thèse
MY ABDELOUAHED SABRI	PES	FSDM, USMBA, Fès	Directeur de thèse

Remerciements

Grâce à Dieu, cette thèse a pu être réalisée, et avec l'aide de nombreuses personnes qui ont donné de leur temps, de leur encouragement, de leur soutien et de leurs connaissances. Merci à tous ceux qui ont rendu ce travail possible.

Je tiens à remercier mes directeurs de thèse, Pr. Abdellah Aarab et Pr. My Abdelouahed Sabri, à la Faculté des Sciences Dhar El Mehraz de Fès, qui m'ont encadré tout au long de cette thèse. Qu'ils soient aussi remerciés pour leurs gentillesses, leurs disponibilités permanentes et pour les nombreux encouragements qu'ils m'ont prodigués.

Ma thèse de doctorat a été réalisée en collaboration avec Nextronic et la Faculté des Sciences Dhar El Mehraz de Fès. Je tiens à exprimer ma profonde gratitude à Mr. Mohamed Benouda, le président de la société, ainsi qu'à Mr. Yassin Abouch, mon directeur, pour leur soutien inestimable et leur aide précieuse dans la réalisation de mes travaux de thèse.

Je tiens à exprimer mes sincères remerciements à Monsieur Ali Yahyaouy, Professeur à l'Université Sidi Mohammed Ben Abdellah, Fès, pour avoir accepté de présider le jury. J'adresse tous mes remerciements à Madame Fatima El Khoukhi, Professeur à la Faculté des Lettres et des Sciences Humaines, Université Moulay Ismail, Meknès, ainsi qu'à Messieurs El Mehdi Ismail Alaoui et Rachid El Alami, Professeurs à l'Université Moulay Ismail, Meknès et à l'Université Sidi Mohammed Ben Abdellah, Fès, respectivement, de l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs et examinateurs de cette thèse.
J'exprime également ma gratitude à Monsieur El Habib Nfaoui, Professeur à l'Université Sidi Mohammed Ben Abdellah, Fès, qui a bien voulu être examinateur.

Enfin, je tiens à témoigner toute ma gratitude à ma famille pour leur soutien constant, leurs encouragements et leur patience tout au long de cette aventure. Je souhaite également exprimer ma sincère reconnaissance à mon mari pour son appui précieux et sa compréhension pendant la rédaction de cette thèse.

Dédicace

Je dédie cette thèse à :

Mes chers parents la source inépuisable de tendresse, de patience et de sacrifice, leur prière et leur bénédiction m'ont été d'un grand secours tout au long de ma vie.

Ma chère sœur, vous avez été à mes côtés pendant toutes les étapes de ce travail, je t'ensuis très reconnaissant.

Mon cher mari, ton soutien inébranlable m'ont guidée. Ma gratitude envers toi est sans fin.

Mon frère bien aimé.

Je vous aime beaucoup

Résumé

Ce travail de thèse porte sur la conception d'un système embarqué pour la supervision et l'analyse en temps réel du trafic routier. Utilisant le Deep Learning, ce système vise à améliorer la gestion des problèmes de congestion. Ce système pivote autour de la détection, la classification et le suivi d'objets en mouvement, ainsi que sur leur implémentation matérielle.

Le premier axe de recherche a abouti au développement d'une nouvelle approche pour la segmentation des objets en mouvement dans une séquence vidéo, essentielle pour leur suivi précis. Nous avons élaboré un algorithme de détection d'objets performant, capable de distinguer efficacement les objets en mouvement des objets statiques, même dans des conditions environnementales changeantes.

Le deuxième axe a mené à la mise en place de deux sous-systèmes embarqués intelligents, distincts et collaboratifs. Le premier, basé sur la carte Allwinner A133, se concentre sur la collecte de données en temps réel. Le second, utilisant la carte Jetson Nano, est chargé du comptage et suivi des véhicules en mouvement. Des algorithmes de Deep Learning ont été adaptés et optimisés pour l'implémentation matérielle.

Ce travail est le fruit d'une collaboration entre le laboratoire de recherche LISAC de la Faculté des Sciences Dhar el Mahraz et la société Nextronic. Les tests sur le terrain ont démontré l'efficacité et la durabilité du système proposé, contribuant ainsi de manière significative à la prise de décision concernant les solutions pour minimiser la congestion du trafic routier dans la région de Casablanca.

Mots clés : Systèmes Embarqués, Supervision et analyse en temps réel, Trafic routier, Deep Learning, Gestion des problèmes de congestion, Détection, classification, et suivi d'objets en mouvement, Implémentation matérielle, Segmentation des objets en mouvement, Carte Allwinner A133, Carte Jetson Nano.

Abstract

This thesis work focuses on the design of a new embedded system for the supervision and real-time analysis of road traffic, utilizing Deep Learning to improve the management of congestion issues. The system centers on the detection, classification, and tracking of moving objects, as well as their hardware implementation.

The first research axis led to the development of a new approach for segmenting moving objects in video sequences, essential for their precise tracking. We developed an efficient object detection algorithm capable of effectively distinguishing moving objects from static ones, even in changing environmental conditions.

The second axis resulted in the establishment of two distinct and collaborative intelligent embedded subsystems. The first, based on the Allwinner A133 board, focuses on real-time data collection. The second, using the Jetson Nano board, is responsible for counting and tracking moving vehicles. Deep Learning algorithms have been adapted and optimized for hardware implementation.

This work is the result of a collaboration between the LISAC research laboratory at the Faculty of Sciences Dhar El Mahraz and Nextronic. Field tests demonstrated the system's effectiveness and durability, significantly contributing to decision-making regarding solutions to minimize road traffic congestion in the Casablanca region.

Keywords: Embedded System, real-time analysis, Road Traffic, Deep Learning, Congestion Management, Detection, Classification, and Tracking of Moving Objects, Hardware Implementation, Segmentation of Moving Objects, Allwinner A133 Board, Jetson Nano Board.

Table des matières

Remerciements *1*

Dédicace *2*

Résumé *3*

Abstract *4*

Table des matières *5*

Liste des figures *9*

Liste des tableaux *12*

Liste des abréviations *13*

Introduction Générale *14*

Chapitre I : Systèmes Embarqués et Infrastructures Matérielles pour le Deep Learning : Synthèse et Revue de Littérature *19*

1. Introduction **19**

2. Techniques et Architectures en Vision par Ordinateur et Deep Learning **20**

2.1. Techniques de Détection des Objets en Mouvement **20**

2.1.1. Approches Traditionnelles pour la Détection des Objets en Mouvement 20

2.1.2. Méthodes de Détection d'Objets Pré-entraînées 22

2.1.2.1. Détection d'objets utilisant le Faster – Rcnn 23

2.1.2.2. Détection d'objets en utilisant le Mask – Rcnn 25

2.1.2.3. Détection de véhicules utilisant le modèle SSD 26

2.1.2.4. YOLO 28

2.2. Suivi d'Objets **30**

2.3. Comptage d'objets **32**

2.3.1. Comptage par régression 33

2.3.2. Comptage par détection 33

2.3.2.1. Utilisation de Lignes Virtuelles pour le Comptage 34

2.3.2.2. Défis et Avancées dans le Comptage de Véhicules 34

3. Matériel et Systèmes Embarqués en Deep Learning **36**

3.1. Architectures de Systèmes Informatiques pour le Deep Learning 36
3.1.1. Architecture Centralisée 37
3.1.2. Architecture Distribuée 37
3.1.3. Architecture Edge Partielle.....38
3.1.4. Architecture Edge Complète 38
3.1.4.1. Adoption de l'Architecture Edge Complète 39
3.2. Infrastructure Matérielle Clé pour le Deep Learning : Ordinateurs, Processeurs et Stockage 39
Ordinateurs et Serveurs pour le Deep Learning 40
Processeurs et GPU : Accélération des Calculs 40
Systèmes de Stockage pour Données d'Entraînement 40
GPU Populaires en Deep Learning.....40
Processeurs pour Applications de Deep Learning.....40
3.3. Les SBC pour l'apprentissage en profondeur : une liste des cartes électroniques les plus populaires 40
3.3.1. Définition des cartes SBC 41
3.3.2. Deep Learning sur les SBC pour l'Edge Complet : Cartes Électroniques Compatibles.....41
3.4. Avancées Technologiques et Innovations 46
4. Conclusion 48
Chapitre II : Détection des objets en mouvement et résultats expérimentaux 51
1. Introduction.....51
2. Détection des objets en mouvement 52
2.1. Soustraction d'arrière-plan.....52
2.2. Différence inter-image.....54
2.3. Flux Optique pour la Détection d'Objets en Mouvement 55
3. Segmentation pour la Détection d'Objets.....56
3.1. Segmentation par seuillage 57
3.2. Segmentation par clustering 57
3.3. Segmentation basée sur les contours.....57
4. Extraction des Caractéristiques.....58
4.1. Caractéristiques Basées sur la Couleur 58
4.2. Caractéristiques Basées sur la Forme.....58
4.3. Caractéristiques Basées sur la Texture 59

4.4. Extraction des Caractéristiques à l'aide de GLCM 59
5. Classification des objets 59
5.1. Réseau Neuronal 61
5.2. Naïf Bayésien 61
5.3. Machine à Vecteurs de Support (SVM).....62
6. Approche Proposée pour la segmentation des objets « Triangulation de Delaunay ».....63
7. Étiquetage des Composants Connectés.....67
8. Phase d'extraction des caractéristiques.....68
9. Approche proposée pour la classification des Objets.....70
10. Résultats Expérimentaux et Évaluation 71
10.1. Présentation des jeux de données utilisés pour évaluer la performance de l'approche.....71
10.2. Entraînement du Classificateur.....72
10.3. Évaluation du Système.....72
11. Conclusion 75
Chapitre III : Implémentation d'un Réseau Neuronal sur une carte embarquée 78
1. Introduction.....78
2. Matériel et méthodes 78
2.1. Site d'étude 78
2.2. Matériel.....79
2.2.1. Le nano-ordinateur « Jatson Nano » 79
2.2.2. Le nano-ordinateur « Allwinner A133 » 82
4.1.1.1. Spécifications du Allwinner A133.....84
2.3. Méthodes.....85
2.3.1. Collecte de données 87
Particularités de la surveillance d'un rond-point 87
2.3.1.1. Acquisition des données.....88
2.3.1.2. Annotation des données 90
2.3.1.3. Division des données 91
2.3.2. Méthodes de Détection des Véhicules 92
2.3.2.1. Choix de YOLOv7 pour la Détection de Véhicules 92
2.3.2.2. YOLOv7-tiny : Un Choix Stratégique.....94
2.3.2.3. YOLOv7 et Jetson Nano 97

A- Paramétrage du Jetson Nano pour YOLOv7-tiny 97
B- Déploiement et Conversion du Modèle sur la Jetson Nano 98
2.3.2.4. Établissement des Objectifs de Performance 99
A- Protocole d'Évaluation de la Détection 99
2.3.3. Suivi des véhicules 101
2.3.4. Évaluation du Trafic aux Carrefours 103
3. Résultats 105
3.1. Résultats de l'entraînement 105
3.2. Résultats de comptage 107
3.3. Évaluation du Système 108
3.4. Visualisation des Données 109
4. Discussion 111
5. Conclusion 111
Conclusion générale et perspectives 113
Liste de publications 115
1. Articles 115
2. Conférences 115
Bibliographie 116

Liste des figures

Figure 1 : Architecture et fonctionnement de Faster R-CNN pour la détection d'objets 25

Figure 2 : Détection de trafic urbain en utilisant Mask R-CNN 26

Figure 3 : Architecture du réseau neuronal Single Shot MultiBox Detector (SSD) 28

Figure 4 : Mécanisme de détection d'objets utilisant YOLO 30

Figure 5 : Détection et comptage de véhicules via une ligne virtuelle 34

Figure 6 : Flux de travail du système de comptage de véhicules 36

Figure 7 : Architecture centralisée pour le traitement et l'analyse de flux de données 37

Figure 8 : Architecture Distribuée pour le traitement et l'analyse de flux de données 38

Figure 9 : Architecture Edge Partielle pour le traitement et l'analyse de flux de données 38

Figure 10 : Architecture Complète pour le traitement et l'analyse de flux de données 39

Figure 11 : Raspberry Pi (RPi) 42

Figure 12 : Carte Jetson Nano 44

Figure 13 : Carte Coral Dev 46

Figure 14 : Techniques de détections 52

Figure 15 : Présentation de la méthode de soustraction d'arrière-plan 54

Figure 16 : Détection de mouvements d'objets via la méthode du flux optique 56

Figure 17 : Techniques de classification des objets 60

Figure 18 : Architecture d'un réseau neuronal 61

Figure 19 : Naïf Bayésien 62

Figure 20 : Principe des SVM 63

Figure 21 : Construction du diagramme de Voronoi à partir de points 2D : Avant (a) et Après (b) 64

Figure 22 : Triangulation de Delaunay (bleu) de Cinque motifs de points et leur VD (lignes orange). 65

Figure 23 : Résultat de la triangulation de Delaunay 67

Figure 24 : Résultats de test de la détection des objets. (a) DT segmentation, (b) Etiquetage des objets connecté(c) Détection des objets dans l'image originale. 68

Figure 25 : Comparaison des performances de classification entre le réseau neuronal, le classifieur bayésien naïf et la SVM 71

Figure 26 : Résultats de détection de notre algorithme sur deux vidéos prises à différents moments 75

Figure 27 : Localisation du site d'étude sur Maps 79

Figure 28 : Jetson Nano pour l'implémentation GPU 80

Figure 29 : Schéma de Bloc du Jetson Nano 81

Figure 30 : Allwinner A133 83

Figure 31 : Diagramme d'application de la carte Allwinner A133 83

Figure 32 : Architecture proposée pour la détection du flux de trafic. La détection, le suivi et le comptage des véhicules sont directement effectués sur le Nvidia Jetson Nano 87

Figure 33 : Vue de la caméra 88

Figure 34 : L'architecture du système utilisé pour la collecte de l'ensemble de données 89

Figure 35 : Exemple de différents ensembles de données 90

Figure 36 : Annotation des véhicules sur une image de rond-point 90

Figure 37 : Véhicule avec caractéristique perdue 91

Figure 38 : Répartition des données 92

Figure 39 : Résultats de YOLOv7 93

Figure 40 : YOLOv7 Architecture. Les changements dans cette architecture incluent les blocs ELAN qui combinent les caractéristiques de différents groupes en les mélangeant et en fusionnant la cardinalité pour améliorer l'apprentissage du modèle, et un RepVGG modifié sans connexion identité. Diagramme basé sur [102] 97

Figure 41 : Identification et suivi des véhicules par Deep SORT 102

Figure 42 : Architecture de l'algorithme Deep SORT 103

Figure 43 : Schéma illustratif des lignes virtuelles utilisées pour compter les véhicules 103

Figure 44 : Mécanisme utilisé pour vérifier si une boîte englobante a été intersectée par la ligne de comptage. 104

Figure 45 : Schéma illustrant la trajectoire possible des véhicules provenant de la Voie 2 105

Figure 46 : Résultats de l'entraînement du modèle pré-entraîné YOLOv7-tiny pour la détection d'objets. (a) Perte de Coordonnées ; (b) Perte d'Objet ; (c) Perte de Classification ; (d) Précision ; et (e) Rappel..... 106

Figure 47 : Courbe de précision-rappel..... 107

Figure 48 : Exemple issu d'une simulation 108

Figure 49 : Résultats du comptage pour la période du 5 décembre 2022 au 13 décembre 2022 110

Figure 50 : Diagrammes circulaires des itinéraires empruntés par les véhicules, 08/12/2022. 111

Liste des tableaux

Tableau 1 : Spécifications Techniques du Raspberry Pi..... 42

Tableau 2 : Spécifications Techniques du Jetson Nano 43

Tableau 3 : Spécifications Techniques de la carte Coral Dev 45

Tableau 4 : Comparaison des taux de reconnaissance selon différentes méthodes. 73

Tableau 5 : Spécifications du Allwinner A133 84

Tableau 6 : Résultats de comptage pour les six voies de l'intersection..... 109

Liste des abréviations

AI	Intelligence Artificielle
CNN	Réseau Neuronal Convolutif
CPU	Unité Centrale de Traitement
DT	Triangulation de Delaunay
ELAN	Réseau Étendu d'Agrégation de Couches Efficace
FLOPs	Opérations en Virgule Flottante par Seconde
FN	Faux Négatif
FP	Faux Positif
FPS	Image par seconde
GLCM	Matrice de Co-occurrence de Niveaux de Gris
GMM	Modèle de Mélanges Gaussiens
GPU	Unité de traitement graphique
HDMI	Interface Multimédia Haute Définition
HSV	Teinte, Saturation, Valeur
MQTT	Transport de Télémesure par File d'Attente de Messages
MOT	Suivi de Multiples Objets
NMS	Suppression non maximale
ONNX	Échange de Réseaux Neuronaux Ouverts-
RAM	Mémoire à Accès Aléatoire
RCNN	Réseaux de Neurones Convolutifs à Régions
RGB	Rouge, Vert, Bleu
RPN	Réseau de Proposition de Régions
Rpi	Raspberry Pi
R-CNN	Réseau de Neurones Convolutifs Basé sur les Régions
SBC	Ordinateur sur une seule carte
SORT	Suivi en Ligne Simple et en Temps Réel
SOT	Suivi d'un seul objet
SSD	Détecteur Multi-Boîtes à Une Seule Reprise
SVM	Machine à Vecteurs de Support
TPU	Unité de traitement de tenseur
VD	Diagramme de Voronoi
YOLO	You Only Look Once

Introduction Générale

Au cours des dernières décennies, l'évolution des systèmes de vidéosurveillance a été remarquable. Initialement conçus pour des applications traditionnelles comme la surveillance des banques, stations de métro et aéroports, ces systèmes sont devenus incontournables dans la gestion du trafic routier. Aujourd'hui, ils jouent un rôle central dans la régulation du trafic, la détection d'infractions et l'assurance de la sécurité des usagers sur nos routes et intersections [1]. Le secteur routier, ayant bénéficié de ces avancées technologiques, a vu des améliorations significatives en matière de sécurité grâce à des investissements substantiels [2].

Cependant, l'efficacité de ces systèmes repose traditionnellement sur la vigilance des opérateurs humains. Une étude du Centre de recherche informatique de Montréal (CRIM) a souligné une baisse notable de l'attention des opérateurs après environ 20 minutes de surveillance continue [3]. Cette limitation met en évidence le besoin impérieux d'automatisation dans le domaine de la surveillance routière.

C'est dans cette perspective que s'inscrit notre étude, qui se donne pour mission essentielle de réduire les erreurs humaines et d'accroître l'efficacité de ces systèmes. Notre attention se porte plus particulièrement sur la détection des objets en mouvement, une pierre angulaire dans la réduction des risques et la préservation de la sécurité sur nos routes. L'automatisation de cette étape cruciale vise à minimiser les éventuelles défaillances liées à la fatigue ou à la distraction humaine, tout en garantissant une surveillance constante et précise des mouvements routiers.

La détection précise des objets en mouvement est essentielle dans le secteur routier, contribuant significativement à la sécurité et à la fluidité de la circulation. Pour atteindre ces objectifs, des systèmes de détection d'une fiabilité sont indispensables. Ils permettent non seulement une identification précise des mouvements, mais contribuent également à la réduction des fausses alertes et à une signalisation adéquate des dangers. En somme, ils constituent la fondation même de l'efficacité des applications déployées sur les routes.

Il convient toutefois de souligner que les méthodes traditionnelles de détection, telles que l'analyse du flux optique, la soustraction d'images et de fond, ainsi que le suivi de points spécifiques, rencontrent des obstacles notables. Ces techniques se révèlent souvent sensibles aux variations d'éclairage et aux scènes dynamiques complexes. De telles limitations entraînent

une diminution de leur fiabilité dans les contextes routiers diversifiés et exigeants, ce qui renforce la nécessité de recourir à des solutions d'automatisation plus sophistiquées et adaptées.

Pour répondre à ces défis, l'élaboration de méthodes plus robustes et précises pour détecter les objets en mouvement dans les applications routières est essentielle. L'accent est mis sur le développement d'une application révolutionnaire pour une détection fine des objets en mouvement sur les voies publiques, en exploitant les dernières techniques avancées. Nous introduisons un algorithme, spécifiquement adapté pour gérer les variations d'éclairage, les ombres et les anomalies, offrant ainsi une solution plus fiable et adaptée aux exigences des environnements routiers.

La deuxième phase de cette thèse a ciblé la lutte contre la congestion routière. L'accent a été mis sur l'intégration de technologies avancées, orientées vers des solutions matérielles pour le comptage et la classification des véhicules. Nous avons utilisé des cartes électroniques avancées comme l'Allwinner A133 pour la collecte de données, et le Nvidia Jetson Nano pour implémenter des techniques de détection basées sur le Deep Learning.

Dans le cadre du projet déployé à Bouskoura, une ville dynamique et en expansion, Une caméra a été stratégiquement positionnée pour observer simultanément les différentes voies d'un rond-point, capturant les véhicules proches de manière plus grande et ceux éloignés de manière plus petite. Pour renforcer l'efficacité du modèle de Deep Learning, une base de données exhaustive de 13 000 images a été constituée. Ces images couvraient un large éventail de scénarios, incluant divers états météorologiques ainsi que des moments de jour et de nuit, pour assurer une détection fiable des véhicules quelles que soient les circonstances. Chaque image a été méticuleusement étiquetée pour améliorer la précision de l'identification et du suivi des véhicules. Cette approche détaillée a permis au modèle d'exceller dans la reconnaissance des véhicules, contribuant ainsi de manière significative à l'amélioration de la gestion du trafic.

En ce qui concerne le choix matériel, le Jetson Nano s'est imposé pour ses capacités avancées de traitement de Deep Learning, tout en étant économe en énergie. Son alimentation a été facilitée dans notre contexte opérationnel spécifique. Un aspect clé de sa configuration a été sa connexion à Internet, établie via un point d'accès utilisant une clé WiFi. Cette méthode de connectivité a assuré une transmission des données à la fois fiable et efficace. En combinant puissance de calcul, efficacité énergétique, et connectivité adaptée, le Jetson Nano a joué un

rôle fondamental dans l'efficacité de notre système de gestion du trafic.

L'impact positif de ce système s'est clairement manifesté dans les mesures prises par la municipalité de Bouskoura. Grâce aux données précises recueillies par notre caméra équipée de YOLOv7-tiny et de l'algorithme Deep SORT, tous deux fonctionnant sur le Jetson Nano, la commune a pu identifier les zones de forte densité de circulation au rond-point. Cette analyse a conduit à une décision d'élargissement des voies les plus encombrées, une mesure significative pour améliorer la fluidité du trafic. Ce résultat témoigne non seulement de l'importance des technologies avancées dans la collecte de données pour la planification urbaine, mais aussi de la manière dont des informations précises peuvent influencer positivement les décisions d'aménagement urbain, même dans des contextes complexes.

La composition du manuscrit est organisée en trois chapitres distincts, chacun structuré de manière hiérarchique :

Le premier chapitre explore l'évolution rapide de la technologie, en se concentrant sur les systèmes embarqués et les infrastructures matérielles essentiels pour le Deep Learning. Il examine leur impact sur la vision par ordinateur et leur rôle croissant dans la gestion du trafic routier, mettant en avant les techniques de détection, de suivi et de comptage d'objets en mouvement. L'objectif est de fournir une compréhension approfondie des innovations technologiques et de leur application pratique, notamment dans l'amélioration de la sécurité et de l'efficacité du trafic.

Le deuxième chapitre se concentre sur la détection d'objets en mouvement, un élément clé de la vision par ordinateur. Il traite des méthodes traditionnelles et de leur évolution vers des techniques basées sur l'apprentissage automatique, comme le SVM (Machine à Vecteurs de Support). L'accent est mis sur les processus de segmentation et d'extraction de caractéristiques, explorant des techniques spécifiques comme la méthode de la triangulation de Delaunay (DT). Le chapitre se conclut par des résultats expérimentaux démontrant l'efficacité de ces méthodes dans divers contextes routiers.

Enfin Le troisième chapitre détaille l'application pratique de technologies avancées pour le comptage des véhicules dans la ville de Bouskoura. Il met en avant l'utilisation de cartes électroniques, notamment l'Allwinner A133 pour la collecte de données et le Nvidia Jetson Nano pour la détection et le comptage des véhicules. L'intégration de YOLOv7-tiny et de

l'algorithme Deep SORT souligne l'efficacité du système dans la gestion du trafic, marquant une avancée dans les infrastructures de mobilité urbaine.

Dans la section finale, intitulée « Conclusion Générale et Perspectives », nous récapitulons les principaux résultats et enseignements tirés de notre thèse. Nous discutons également des orientations futures de notre recherche, en mettant en lumière de nouvelles voies et perspectives à explorer dans la continuité de notre travail.

Chapitre 1

Systèmes Embarqués et Infrastructures Matérielles pour le Deep Learning : Synthèse et Revue de Littérature

Chapitre 1 : Systèmes Embarqués et Infrastructures Matérielles pour le Deep Learning : Synthèse et Revue de Littérature

1. Introduction

Dans un monde où la technologie évolue à un rythme effréné, les systèmes embarqués et les infrastructures matérielles se positionnent comme des piliers essentiels du progrès dans le domaine du Deep Learning. L'avènement du Deep Learning a révolutionné de nombreux secteurs, notamment celui de la vision par ordinateur, ouvrant la voie à des avancées significatives dans la compréhension et l'analyse des données visuelles complexes. Cette révolution technologique est particulièrement pertinente dans le contexte de la gestion du trafic routier, un secteur confronté à des défis croissants liés à l'urbanisation et à la mobilité.

Ce chapitre propose une immersion dans le monde fascinant des systèmes embarqués et des infrastructures matérielles spécifiquement conçus pour le Deep Learning. Nous explorons comment ces technologies transforment la vision par ordinateur, en mettant un accent particulier sur les techniques de détection, de suivi et de comptage d'objets en mouvement. En scrutant les avancées récentes et les architectures innovantes, nous cherchons à comprendre comment elles contribuent à résoudre les problématiques complexes de notre ère, notamment dans la gestion du trafic routier.

L'objectif de ce chapitre est double. D'une part, il vise à fournir une compréhension approfondie des aspects techniques et des innovations en matière de Deep Learning et de vision par ordinateur. D'autre part, il cherche à établir un lien entre ces avancées technologiques et leurs applications concrètes, en examinant comment elles peuvent améliorer la sécurité, l'efficacité et l'impact environnemental dans des domaines aussi critiques que la gestion du trafic. À travers cette revue de littérature, nous aspirons non seulement à dresser un tableau des progrès réalisés, mais aussi à identifier les défis persistants et les opportunités futures dans ce domaine en évolution constante.

2. Techniques et Architectures en Vision par Ordinateur et Deep Learning

2.1. Techniques de Détection des Objets en Mouvement

Dans notre exploration des méthodes de détection d'objets en mouvement, nous nous pencherons sur une gamme de techniques utilisées pour repérer des objets dynamiques au sein de séquences vidéo. Ces approches sont essentielles pour surveiller les activités dans des environnements tels que la gestion du trafic routier. La précision de la détection joue un rôle crucial dans la qualité des données recueillies à partir des zones d'intérêt identifiées. Deux approches principales sont employées pour obtenir ces données : l'utilisation de méthodes telles que la soustraction de l'arrière-plan, la différence entre images successives et l'application du flux optique, qui se basent sur des calculs et des comparaisons pixel par pixel pour repérer les changements dans la scène [4, 5]. La deuxième approche fait appel à des modèles de détection d'objets pré-entraînés ou à des détecteurs spécifiquement conçus pour reconnaître des objets dans des contextes variés [6].

Cette section examinera en détail ces méthodes, en mettant en lumière leurs forces, leurs limites et leurs applications potentielles dans le domaine de la surveillance routière. Nous explorerons comment les méthodes traditionnelles telles que la soustraction de l'arrière-plan peuvent être utilisées pour détecter les objets en mouvement en analysant les variations de pixel entre les images successives. Nous aborderons également comment les approches basées sur des modèles pré-entraînés ou des détecteurs d'objets spécialisés, comme YOLO et Faster R-CNN [7,8], peuvent être mises en œuvre pour une détection plus précise et efficace des objets en mouvement. Cette compréhension approfondie des méthodes de détection d'objets en mouvement constitue une étape cruciale pour le suivi, le comptage et la détection d'anomalies dans le trafic routier.

2.1.1. Approches Traditionnelles pour la Détection des Objets en Mouvement

Lorsqu'il s'agit de détecter des objets en mouvement dans une scène vidéo, une pléthore de méthodes s'offre aux chercheurs. Parmi ces approches, l'une des plus anciennes et des plus couramment utilisées est la "Soustraction de l'Arrière-Plan". Cette technique, fondée sur un concept à la fois simple et puissant, exploite le fait que les objets en mouvement se distinguent généralement de leur environnement en provoquant des variations d'intensité lumineuse au fil du temps.

Le processus de soustraction de l'arrière-plan prend son envol en créant un modèle initial de l'arrière-plan, une représentation statique de la scène sans les objets en mouvement. À chaque image vidéo, une comparaison serrée s'opère entre cette image et le modèle d'arrière-plan. Les zones de l'image où des variations significatives sont identifiées sont considérées comme des objets en mouvement. Cette technique est particulièrement efficace dans des vidéos où les transitions de contexte sont relativement simples, tels que les environnements de surveillance ou de gestion du trafic. Elle produit des "régions d'intérêt", qui correspondent aux zones de l'image contenant des objets en mouvement. Ces régions simplifient grandement leur suivi et leur analyse ultérieurs.

Parmi les approches les plus notoires pour la soustraction de l'arrière-plan figure la méthode de la "Distribution Gaussienne", pionnière grâce à l'œuvre de McKenna et al. En (2000). [9] Cette technique modèle l'arrière-plan à l'aide d'une distribution gaussienne. Chaque pixel de l'image est caractérisé par sa moyenne (μ) et sa variance (σ^2) dans chaque canal de couleur (R, G, B). Lors de chaque itération, un pixel est étiqueté comme "en mouvement" si sa valeur actuelle ($X_{t,i}$) diffère significativement de la moyenne précédente ($\mu_{t-1,i}$) par rapport à trois fois l'écart-type précédent ($3\sigma_{t-1,i}$), où "i" représente un des canaux de couleur (R, G ou B). Dès qu'une de ces conditions est vérifiée pour l'un des canaux, le pixel est classé comme appartenant à la région en mouvement. Ensuite, une mise à jour du modèle de l'arrière-plan s'impose, afin de refléter les variations dans la scène. La méthode de distribution gaussienne permet d'isoler efficacement les zones en mouvement dans une vidéo en continu, un atout précieux pour la surveillance de la circulation routière et bien d'autres applications.

En (2000), Stauffer et Grimson [10] ont proposé une évolution de cette méthode, s'appuyant sur un "Mélange de Distributions Gaussiennes" pour modéliser l'arrière-plan. Cette approche se montre plus résistante face aux bruits et aux mouvements mineurs de l'arrière-plan. Concrètement, chaque pixel est représenté par un ensemble de K distributions gaussiennes, chacune pondérée pour maintenir le modèle d'arrière-plan actuel.

D'autres approches explorent les caractéristiques texturales de l'image, comme la méthode "Détection d'Objets Mobiles Basée sur la Texture" (TBMOD) introduite par Heikkila et Pietikainen en (2006). [11] Cette technique associe à chaque pixel un motif binaire, obtenu par comparaison avec son environnement immédiat. Pour une précision accrue, elle crée des histogrammes liés aux motifs locaux, extraits d'une région circulaire entourant chaque pixel. La

détection s'opère ensuite par la comparaison de ces divers histogrammes.

D'autres chercheurs ont également contribué à l'enrichissement de ces méthodes. El Kah et al. (2019) [12] ont développé une approche qui repose sur la mesure de contraste, montrant une grande résilience face aux perturbations extérieures telles que le mouvement des arbres ou les obstructions. Kavitha et Tejaswini (2012) [13] ont conçu un algorithme amélioré qui surmonte les problèmes liés aux variations de luminosité, tels que les ombres et les éclairages. Lu et al. (2008), [14] dans le domaine des algorithmes en temps réel, ont fusionné les méthodes de flux optique, de différentiation temporelle et de double filtrage des arrière-plans pour maximiser les performances.

Bouwmans et al. (2018) [15] ont contribué en effectuant une revue préliminaire des caractéristiques basées sur les types et les tailles, ainsi qu'en explorant les caractéristiques spectrales, spatiales et temporelles intrinsèques. D'autres chercheurs, comme Devi et al. (2013) [16] et Aqel et al. (2015) [17], ont affiné ces méthodes en comparant les valeurs des pixels entre images successives ou en se basant sur la transition de l'intensité des pixels. Des algorithmes ont également été proposés pour la classification des objets en mouvement Aqel et al. En (2017) [18] et des modèles spécifiques ont été développés pour la surveillance routière Perreault et al., (2019) [19].

Enfin, Giraldo et al. (2020) [20] ont ouvert la voie à une nouvelle ère en enquêtant sur l'application de l'apprentissage profond à la soustraction de l'arrière-plan. Leur étude a débuté par une comparaison des différentes architectures utilisées, tout en offrant une discussion approfondie sur les besoins spécifiques des applications. Cette transition vers l'apprentissage profond marquera notre transition vers la "Détection d'Objets Pré-entraînés", une ère où des réseaux de neurones sophistiqués seront nos outils privilégiés pour traquer des objets en mouvement dans des vidéos complexes.

2.1.2. Méthodes de Détection d'Objets Pré-entraînées

La détection d'objets en vision par ordinateur constitue une tâche complexe qui vise à repérer, localiser et classifier les objets présents dans une image. Cette procédure se décompose généralement en trois étapes fondamentales : la reconnaissance des objets, l'identification de leur emplacement, et leur catégorisation dans des classes spécifiques. Récemment, l'application réussie de techniques d'apprentissage profond dans ce domaine a généré des résultats prometteurs, tant sur des ensembles de données standards que dans le cadre de défis en vision

par ordinateur.

Les détecteurs d'objets pré-entraînés se basent sur des réseaux neuronaux convolutifs (CNN) qui ont été préalablement entraînés sur de vastes ensembles de données contenant divers types d'objets [21]. L'idée sous-jacente est que ces modèles ont déjà acquis la capacité de reconnaître une variété d'objets dans différentes situations. Ainsi, il est possible de les adapter pour détecter spécifiquement des objets en mouvement dans des vidéos de surveillance routière ou d'autres applications similaires.

Il est également important de noter que les réseaux neuronaux convolutifs génèrent des cartes de caractéristiques contenant par défaut plusieurs régions d'intérêt. Ces cartes de caractéristiques proviennent de différentes couches du CNN, avec une tendance à réduire leur taille et à augmenter leur complexité à mesure que l'on progresse vers les couches plus profondes. Les couches profondes traitent des informations à une échelle plus large et construisent des représentations abstraites, tandis que les couches peu profondes se concentrent sur des détails plus fins [22].

En somme, l'utilisation de détecteurs d'objets pré-entraînés basés sur des CNN offre une approche efficace pour résoudre la tâche complexe de détection d'objets, en exploitant les avantages des réseaux profonds et en capitalisant sur leur capacité à extraire des caractéristiques pertinentes à partir des données visuelles.

2.1.2.1. Détection d'objets utilisant le Faster – Rcnn

Les chercheurs se sont profondément investis dans l'optimisation de la détection d'objets en réponse à divers défis tels que les variations lumineuses, les conditions météorologiques et les fluctuations du trafic. Dans ce contexte, l'apprentissage profond a émergé comme une solution privilégiée.

Dans le domaine de la détection d'objets, deux principales catégories de méthodes ont émergé. La première catégorie repose sur une approche en deux étapes : la génération de régions candidates suivie de leur classification. R-CNN, proposé par Girshick et ses collaborateurs en (2014) [23], l'utilisation du CNN a été précurseur pour cette tâche. Il s'appuyait sur l'extraction des caractéristiques des régions candidates avant leur classification. SPP-net [24] a introduit une amélioration avec une méthode de mise en commun des pyramides spatiales.

Fast R-CNN est venu ensuite avec une formation en une seule étape, introduisant une fonction

de perte multitâche et une régression de boîte englobante. Cependant, une avancée majeure a été réalisée avec Faster R-CNN, où Ren et al. En (2016) [7] ont intégré un mécanisme de recherche sélective. Cette intégration a non seulement amélioré la vitesse de détection – Faster R-CNN s'avérant environ 10 fois plus rapide que Fast R-CNN et 250 fois que R-CNN – mais a aussi conservé une précision similaire. Toutefois, avec une fréquence de détection inférieure à 10 images par seconde (FPS) [25], Faster R-CNN reste insuffisant pour des applications en temps réel.

Outre les avancées dans le développement de ces architectures, les chercheurs ont aussi exploré des applications spécifiques. Les CNN, par exemple, ont prouvé leur efficacité dans la reconnaissance d'objets routiers, notamment les panneaux de signalisation [26] et les piétons [27]. Concernant la détection des véhicules, les méthodes sont allées au-delà de l'utilisation de CNN simples, avec des techniques avancées telles que Faster R-CNN. Une étude notable [28] a même proposé une méthode Faster R-CNN multi-tâches capable de fournir des informations de coordonnées 3D d'un objet à partir d'une seule image, facilitant ainsi la conduite autonome.

La figure 1 illustre le processus de détection d'objets utilisant Faster R-CNN. À gauche, une image d'une autoroute sert d'entrée, qui est ensuite traitée par le réseau VGG16 pour produire une carte de caractéristiques. Cette carte est introduite dans le Réseau de Proposition de Régions (RPN) pour identifier les régions candidates d'intérêt. Ces régions subissent un pooling pour obtenir une taille fixe et sont finalement traitées pour déterminer la classe d'objet et la position de la boîte englobante. À droite, l'image originale est montrée avec les boîtes englobantes générées, indiquant la détection réussie des véhicules sur la route.

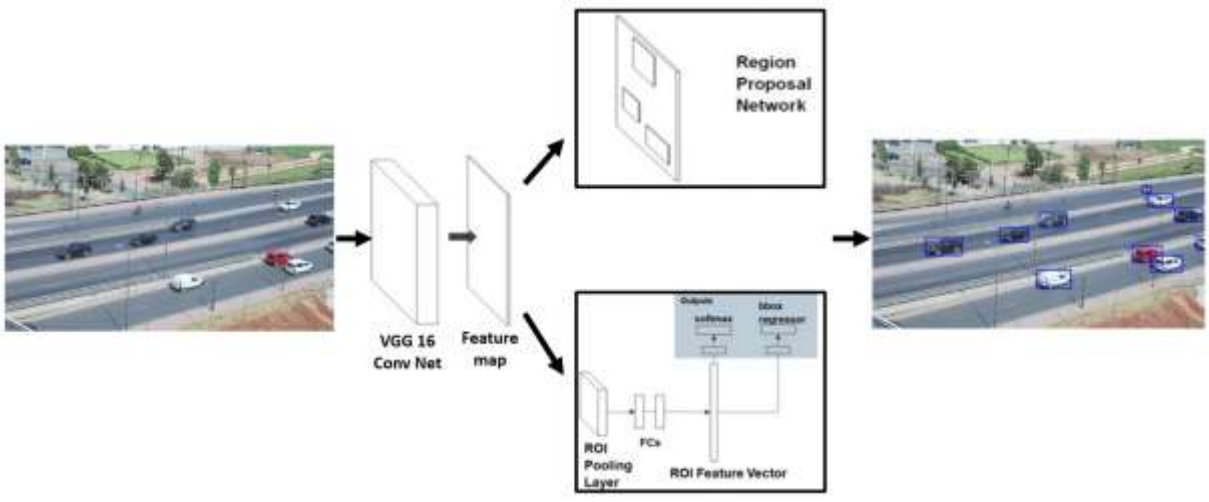


Figure 1 Architecture et fonctionnement de Faster R-CNN pour la détection d'objets

2.1.2.2. Détection d'objets en utilisant le Mask – Rcnn

Concernant la détection d'objets avec le Mask R-CNN, une technique précise pour détecter des véhicules dans des vidéos en exploitant cette méthode est présentée. La détection des véhicules se fait à intervalles réguliers de N images en utilisant le modèle de segmentation d'instance Mask R-CNN. Ce modèle se distingue des autres modèles d'apprentissage profond en fournissant un masque de segmentation pour chaque objet détecté, ce qui le rend particulièrement efficace en cas d'occlusion. Les expériences menées sur diverses vidéos complexes ont révélé des performances exceptionnelles [29].

Mask R-CNN représente une avancée significative par rapport à d'autres méthodes telles que RCNN (Réseaux de Neurones Convolutifs à Régions), FRCNN, SSD [30]. ResNet et RetinaNet en termes de précision de la détection. L'ajout d'une branche de prédiction de masque permet d'améliorer considérablement la capacité de cette méthode à segmenter précisément les objets dans une image.

Cependant, il est important de noter que les approches basées sur Mask R-CNN peuvent être sensibles à de grandes rotations dans les images, ce qui peut avoir un impact négatif sur leur précision de détection.

Malgré leurs performances exceptionnelles en termes de précision, les méthodes telles que Mask R-CNN présentent un coût de calcul élevé, ce qui peut limiter leur utilisation pour des applications nécessitant une détection en temps réel. Pour des contextes en temps réel, il peut être nécessaire de privilégier des méthodes plus rapides, telles que celles basées sur YOLO, qui, bien que légèrement moins précises, offrent une meilleure efficacité en termes de vitesse de détection.

En somme, Mask R-CNN est une méthode puissante pour la détection d'objets avec une précision de segmentation élevée, mais elle peut être moins adaptée aux applications nécessitant une détection en temps réel en raison de son coût de calcul plus élevé.



Figure 2 Détection de trafic urbain en utilisant Mask R-CNN

2.1.2.3. Détection de véhicules utilisant le modèle SSD

Dans le contexte de la détection d'objets dans des environnements complexes et étendus, il est courant de se heurter à un défi délicat : comment équilibrer la précision et la vitesse des systèmes de détection multi-objets, en particulier ceux basés sur l'apprentissage profond et l'exploitation de données massives. Pour aborder cette question, une étude approfondie de la littérature s'impose, mettant en lumière les avancées significatives dans le domaine de la détection d'objets, en mettant l'accent sur les travaux de Wang et de son équipe (2018) [31].

Dans leur recherche novatrice, Wang et son groupe ont adopté une approche stratégique pour améliorer la performance de la méthode (Détecteur Multi-Boîtes à Une Seule Reprise) SSD dans des contextes de trafic exigeants. Ils ont initié une série d'innovations visant à repenser le fonctionnement de SSD. Tout d'abord, ils ont remplacé le noyau de convolution de bas niveau du modèle SSD par un noyau de convolution conçu pour l'extraction de caractéristiques. Cette modification s'est avérée cruciale pour améliorer la capacité de SSD à extraire des informations significatives des images complexes.

De plus, Wang et son équipe ont introduit une approche astucieuse consistant à associer leur méthode de détection d'image unique à des réseaux de convolution à mémoire à court et long terme. Cette intégration a permis de raffiner les caractéristiques extraites des images tout en facilitant leur propagation entre les différentes images d'une séquence. Cette stratégie a renforcé

la capacité du modèle à discerner et à suivre efficacement les objets dans des scènes dynamiques.

Une autre avancée notable a été la substitution d'une méthode de seuillage floue au seuil de confiance statique traditionnel de SSD. Cette approche adaptative a permis au modèle de mieux s'adapter aux variations de la scène, en particulier dans des situations de trafic complexes où les conditions d'éclairage, les occlusions et les autres facteurs peuvent fluctuer considérablement.

Enfin, Wang et ses collaborateurs ont développé un réseau spécialisé dans l'amplification dynamique des régions, visant à améliorer la détection des objets de petite taille dans des images à résolution limitée. Cette composante a été essentielle pour garantir que le modèle SSD puisse identifier efficacement les détails subtils, même lorsque les objets sont peu visibles.

L'ensemble de ces améliorations a conduit à des résultats impressionnants, avec un modèle SSD amélioré atteignant une précision moyenne de 91,83 % sur l'ensemble de données KITTI, tout en maintenant une vitesse de traitement exceptionnelle de 32 images par seconde. En comparaison avec les cadres traditionnels tels que Faster R-CNN, DSOS300, DSSD513 et YOLOv2 544, cette approche s'est démarquée en affichant un score mAP supérieur, la plaçant ainsi au rang de la deuxième méthode de détection d'objets la plus rapide. Ces avancées offrent des perspectives prometteuses pour la détection d'objets dans des scènes de trafic exigeantes, où la précision et la vitesse sont essentielles pour des applications telles que la conduite autonome.

En parallèle, Liu et son équipe (2016) [32] ont proposé une approche novatrice avec le détecteur à boîte unique (SSD), remplaçant le RPN par une méthode d'extraction uniforme. Cette amélioration a permis d'atteindre des vitesses de détection allant jusqu'à 19 FPS (SSD512) et 46 FPS (SSD300).

L'image 4 représente l'architecture du réseau neuronal Single Shot MultiBox Detector (SSD). Contrairement à d'autres méthodes qui génèrent des propositions de région avant la classification, le SSD réalise la détection et la classification simultanément sur différentes échelles de l'image.

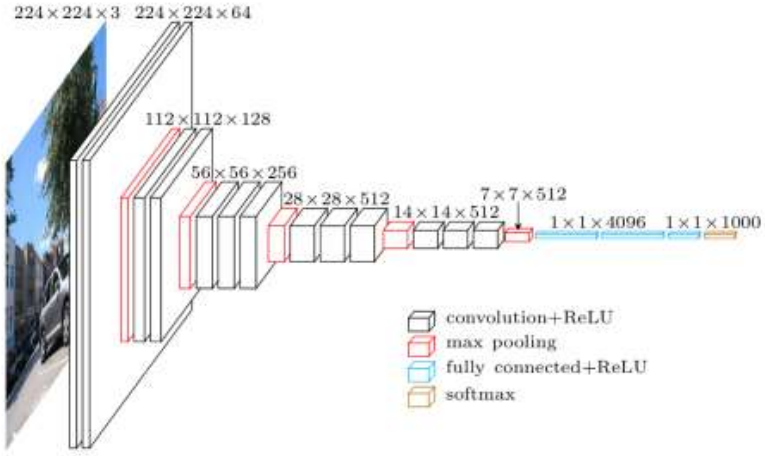


Figure 3 Architecture du réseau neuronal Single Shot MultiBox Detector (SSD)

2.1.2.4. YOLO

YOLO, acronyme de "You Only Look Once", est le précurseur des méthodes de détection d'objets en une seule étape, introduites en 2016. Cet algorithme a innové en prédisant simultanément les coordonnées des boîtes englobantes et les probabilités de classe pour chaque objet dans une image en une seule passe [33]. Cette approche découpe l'image en grilles de taille S x S, prédit les boîtes englobantes et leurs probabilités associées, puis applique un seuil pour éliminer les boîtes de faible confiance. YOLO peut être divisée en deux catégories principales d'approches : les méthodes en deux étapes, qui génèrent d'abord des boîtes candidates avant de les classifier, et les méthodes en une étape, qui prédisent et classent directement les boîtes candidates sans étapes intermédiaires.

Dans cette section, nous passerons en revue les travaux précédents sur la détection d'objets basée sur CNN, en mettant l'accent sur les contributions de Joseph Redmon à travers l'évolution des algorithmes, il a joué un rôle prépondérant dans le développement de la série d'algorithmes (YOLO) [33-36]. Ces algorithmes ont révolutionné la détection d'objets en combinant vitesse et précision.

Néanmoins, YOLO a rencontré des difficultés dans la détection d'objets de petite taille et d'objets avec des caractéristiques inhabituelles, et il présentait des limites en termes de localisation par rapport à des modèles plus avancés, tels que le Fast R-CNN.

Pour relever ces défis, Joseph Redmon et son équipe ont continué à perfectionner YOLO au fil du temps. YOLOv2 a été introduit en 2017 [8], apportant des améliorations notables grâce à l'utilisation de boîtes d'ancrage, la normalisation par lots dans les couches convolutives, et un classificateur haute résolution.

YOLOv3, sorti l'année suivante, a remplacé le réseau principal de YOLOv2 par Darknet-53 et intégré des éléments tels que l'entropie croisée binaire, la régression logistique pour prédire le "score d'objectivité" des boîtes englobantes, et l'extraction de caractéristiques à trois échelles différentes inspirée de FPN [37].

En début d'année 2020, YOLOv4 a surpassé YOLOv3 en termes de précision et de vitesse d'inférence [35]. Ce modèle fonctionne deux fois plus rapidement que d'autres modèles récents de détection d'objets tout en maintenant une précision comparable. Il se distingue également par sa capacité à être entraîné sur un seul GPU (unité de traitement graphique) conventionnel, ce qui renforce son efficacité.

Au fil du temps, différentes variantes de YOLO ont été développées pour répondre à des besoins spécifiques, notamment YOLOv4-tiny, axée sur la vitesse et l'efficacité de calcul, et YOLOv4-CSP, conçue pour maximiser la précision.

Une autre branche de recherche a vu le jour avec YOLOv5. YOLOv5, implémenté en Python et exploité grâce au framework de Deep Learning PyTorch [38], propose différentes échelles pour son modèle : Nano, Small, Medium, Large et Extra-large. Ces échelles se distinguent principalement par la profondeur et la largeur du modèle, tout en conservant la structure globale de YOLO. Dans certaines expérimentations, le modèle de base le plus petit et le plus rapide, YOLOv5n, a été choisi, puis personnalisé en ajoutant des couches supplémentaires.

L'ajout de ces couches visait à améliorer la précision de la détection des petits véhicules observés sur les autoroutes, tout en maintenant un modèle léger adapté à un déploiement sur des dispositifs embarqués.

En 2022, un modèle ultrarapide, capable de détecter des images à des vitesses allant de 5 FPS à 160 FPS, avec une précision moyenne (AP) de 56,8 % a été lancé. YOLOv7 repose sur le Framework PyTorch de Python, facilitant ainsi la transition de la recherche au déploiement en production grâce à une puissance de calcul accrue [39].

Il est à noter que malgré ces avancées, la détection d'objets reste confrontée à des défis tels que

le scintillement et les problèmes de détection en cas d'occultation ou d'éclairage difficile. Ainsi, l'association d'un système de suivi d'objets est souvent nécessaire pour garantir des mesures précises du nombre d'objets, notamment en cas d'échec de la détection.

La figure 4 illustre les étapes clés du processus de détection d'objets avec YOLO (You Only Look Once). Elle commence par une image d'entrée représentant une scène routière. Cette image est ensuite divisée en une grille de dimensions S x S. Pour chaque cellule de la grille, YOLO génère des boîtes englobantes avec des scores de confiance associés. Finalement, une détection réussie d'une voiture est mise en évidence avec une boîte englobante précise, démontrant l'efficacité de YOLO pour localiser les objets dans une image.

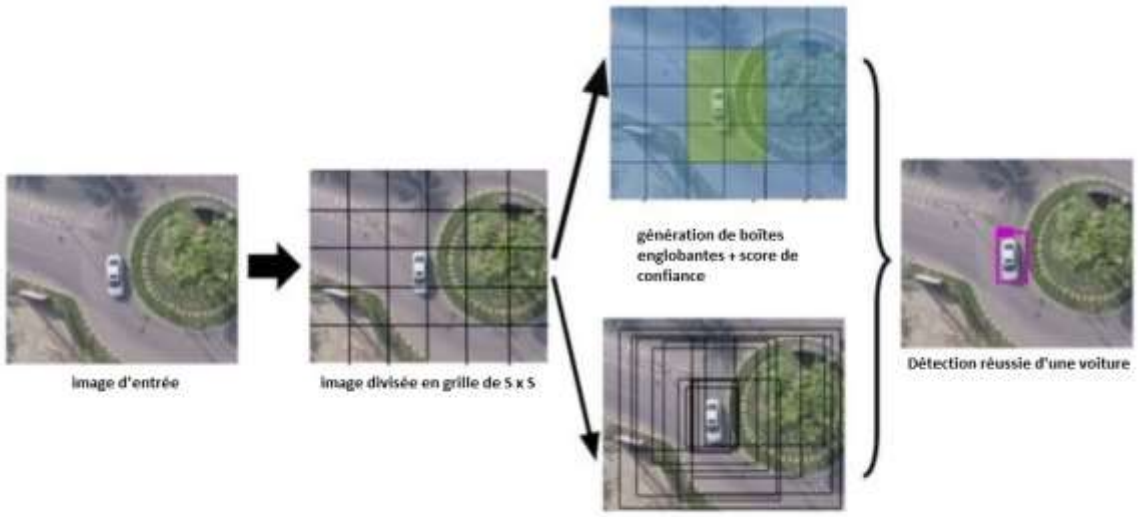


Figure 4 Mécanisme de détection d'objets utilisant YOLO

Dans la suite de cette étude, nous explorerons comment ces avancées en matière de détection d'objets, sont utilisées pour la détection, le suivi et la classification en temps réel des véhicules, notamment sur les ronds-points.

2.2. Suivi d'Objets

La transition de la phase de "détection" à celle de "suivi" constitue une étape essentielle dans notre processus. En effet, notre objectif principal lors du suivi d'objets détectés dans une image est d'assurer un suivi efficace de leurs mouvements. Cette tâche de suivi doit s'effectuer à un rythme bien plus rapide et de manière plus efficiente que la détection initiale, car elle maintient la surveillance des objets au fil du temps tout en pouvant réactiver le détecteur d'objets

ultérieurement. Cette séquence se répète en continu. En général, un algorithme de suivi d'objets opère en tandem avec un détecteur d'objets moins performant. Le rôle du détecteur d'objets est de localiser initialement un objet, tandis que le traqueur d'objets est responsable du suivi continu de chaque objet en leur attribuant des numéros d'identification (ID) et en maintenant ces ID au fil du temps.

Pour un suivi robuste des objets, diverses approches ont été proposées dans la littérature. Zhang et al. (2017) [40] : ont présenté un algorithme de corrélation multitâche qui repose sur la stratégie de l'échantillonnage de particules et exploite l'indépendance entre différentes caractéristiques pour comprendre conjointement les filtres de corrélation. Shantaiya [41] et al. (2015) ont développé une méthode basée sur le filtre de Kalman et la méthode de flux optique pour le suivi en temps réel des objets. Cette approche offre une précision élevée tout en minimisant le temps de calcul. Chantara [42] et al. (2015) ont introduit une approche d'appariement de modèles adaptatifs pour le suivi d'objets, offrant la meilleure localisation des objets dans l'image. Bergmann et al. En (2019) [43] ont développé le modèle de suivi d'objets Tracktor en utilisant un réseau neuronal formé pour la détection d'objets, obtenant des précisions de suivi d'objets multiples élevées. Bochinski et al. (2017) [44] ont introduit le tracker IOU, compétitif en termes de performances tout en ayant un faible coût de calcul. Ont étendu le tracker IOU avec le tracker V-IOU, incorporant un suivi visuel d'objet unique pour résoudre certains problèmes du tracker IOU liés à la durée d'occlusion. Wojke et al. (2017) [45] ont développé DeepSORT, une amélioration du tracker SORT, en remplaçant la métrique d'association par CNN pour le suivi d'objets multiples, résolvant ainsi les problèmes liés aux occlusions.

Avec les progrès récents en matière de détection d'objets, un nombre croissant de recherches se sont orientées vers une approche de suivi basée sur la détection. Cette approche commence par une détection initiale des objets, suivie de leur association. Les méthodologies adoptées dans ce contexte peuvent être segmentées en deux catégories majeures : les méthodes hors ligne et les méthodes en ligne. Les premières s'appuient sur les détections à travers l'ensemble des séquences d'images pour effectuer une optimisation globale. Ces méthodes englobent des techniques basées sur des graphes ainsi que des approches hiérarchiques. Les méthodes en ligne, quant à elles, se concentrent sur les détections de l'image actuelle et des images antérieures, abordant le suivi comme une association de données. Un élément clé de ces méthodes est l'évaluation de la similitude entre chaque objet suivi et chaque détection. Cette

évaluation s'appuie sur divers mécanismes, allant de l'attention à l'utilisation de réseaux neuronaux avancés. Il est également essentiel de distinguer entre le suivi d'un seul objet (SOT) et le MOT (Suivi de Multiples Objets). Alors que le SOT se concentre sur un objet spécifique dès le départ, le MOT s'aligne sur le paradigme du suivi par détection. DeepSORT est l'une des approches les plus notables dans le domaine du MOT, chacune offrant des caractéristiques distinctes pour l'association des données. Le rôle du suivi d'objets dans les systèmes de transport intelligent est indéniable. Historiquement, l'accent a été mis sur le suivi d'objets individuels en utilisant des méthodes traditionnelles. Cependant, avec l'émergence de l'apprentissage profond, de nouvelles méthodes, telles que CCOT (Traqueur d'Opérateur de Convolution Continue) et ECO (Opérateurs de Convolution Efficaces), ont vu le jour, bien que leur adoption pratique reste limitée par rapport aux méthodes basées sur le filtrage par corrélation. Chacune de ces approches offre des avantages spécifiques en fonction des besoins et des contraintes d'application, allant du suivi d'objets généraux au suivi de véhicules dans des scénarios de trafic.

En somme, le suivi d'objets joue un rôle prépondérant dans la vision par ordinateur, offrant une multitude de méthodologies et d'approches pour répondre aux besoins spécifiques de diverses applications. Alors que nous avons exploré en profondeur les nuances du suivi, il est essentiel de souligner que ce n'est qu'une facette de la surveillance routière. Dans les sections suivantes, nous nous pencherons sur deux autres aspects cruciaux : le comptage précis des véhicules et la détection des anomalies sur la route. Ces éléments, combinés au suivi d'objets, forment un ensemble complet d'outils pour assurer une surveillance routière efficace et proactive.

2.3. Comptage d’objets

À l'ère de la numérisation et de l'évolution technologique, les systèmes de comptage basés sur caméra ont émergé comme des outils incontournables pour diverses applications, allant de la gestion du trafic à la surveillance des foules. Ces systèmes, grâce à leur capacité à offrir une surveillance précise et en temps réel, ont été l'objet de nombreuses recherches et développements. La présente revue de littérature vise à explorer l'évolution, les mécanismes, les innovations et les défis associés à ces systèmes. Nous plongerons dans les travaux clés qui ont façonné ce domaine, mettant en lumière les avancées significatives et les domaines nécessitant encore des recherches approfondies.

Avec l'avènement de l'apprentissage automatique, de nombreuses solutions ont été proposées pour aborder cette tâche. Selon la classification de Sindagi and Patel, (2018) [46], les approches

de comptage peuvent être catégorisées en deux méthodes principales : le comptage par régression et le comptage par détection.

2.3.1. Comptage par régression

Le comptage par régression est une approche qui vise à établir une relation directe entre les caractéristiques extraites d'une image et le nombre d'objets qu'elle contient Lempitsky et Zisserman, (2010) [47]. Plutôt que d'identifier chaque objet individuellement, cette méthode se concentre sur l'estimation globale du nombre d'objets. Voici quelques points clés concernant cette méthode :

Extraction de caractéristiques : Les caractéristiques pertinentes sont extraites de l'image, qui peuvent inclure des éléments tels que la texture, la couleur, la forme ou d'autres attributs pertinents.

Cartes de densité : Au lieu de fournir un nombre direct, certaines méthodes de régression estiment une carte de densité Zhang et al., (2016) [48]. Chaque point de cette carte représente la probabilité de la présence d'un objet.

Avantages : Cette méthode est particulièrement efficace dans des scénarios où les objets sont nombreux, superposés ou partiellement occultés Oñoro-Rubio et López-Sastre, (2016) [49].

Limitations : La précision de cette méthode peut être affectée dans des situations où les objets varient considérablement en taille ou en apparence.

2.3.2. Comptage par détection

Le comptage par détection, comme son nom l'indique, repose sur la détection individuelle de chaque objet dans une image ou une séquence vidéo Amato et al., (2018) [50]. Une fois tous les objets détectés, ils sont simplement comptés pour obtenir le nombre total. Voici quelques points clés concernant cette méthode :

Détection d'objets : Des algorithmes spécifiques sont utilisés pour identifier et localiser chaque objet dans l'image. Ces algorithmes peuvent être basés sur des techniques classiques de traitement d'image ou sur des méthodes d'apprentissage profond Ciampi et al., (2018) [51].

Gestion des occlusions : Une attention particulière est accordée à la gestion des objets qui peuvent être partiellement ou totalement occultés par d'autres objets.

Avantages : Cette méthode est précise dans des scénarios où les objets sont clairement visibles et bien séparés. Elle est également utile lorsque des informations supplémentaires, telles que la

position ou la trajectoire des objets, sont nécessaires.

Limitations : La détection individuelle peut être coûteuse en termes de temps et de ressources, en particulier dans des images avec un grand nombre d'objets.

2.3.2.1. Utilisation de Lignes Virtuelles pour le Comptage

Les lignes de comptage sont des outils essentiels dans les systèmes de comptage basés sur la vidéo. Ces lignes, souvent virtuelles, sont tracées à des endroits stratégiques dans la vidéo, comme on peut le voir sur la figure 5. Chaque fois qu'un véhicule traverse l'une de ces lignes, il est compté. Cette méthode est simple, mais elle nécessite une calibration précise pour s'assurer que la ligne est placée à un endroit optimal pour le comptage. Le comptage par zone est une autre méthode utilisée pour compter les véhicules dans des scénarios complexes. Plutôt que de compter les véhicules lorsqu'ils traversent une ligne spécifique, cette méthode divise la vidéo en plusieurs zones et compte les véhicules lorsqu'ils entrent ou sortent de ces zones.



Figure 5 Détection et comptage de véhicules via une ligne virtuelle

2.3.2.2. Défis et Avancées dans le Comptage de Véhicules

Le comptage de véhicules à partir de vidéos présente plusieurs défis. L'un des principaux défis est la qualité de la vidéo. Une vidéo de faible qualité peut rendre difficile la détection précise des véhicules, surtout si elle est granuleuse ou floue. De plus, les conditions météorologiques, comme la pluie ou la neige, peuvent également affecter la qualité de la vidéo et la capacité du système à détecter les véhicules avec précision. Un autre défi est la perspective. Dans les vidéos prises à partir d'un angle élevé, les véhicules peuvent sembler plus petits et être plus rapprochés entre eux. Rendant la détection et le suivi plus difficiles. Enfin, les ombres et les reflets peuvent également poser des problèmes, car ils peuvent être confondus avec des véhicules par le système.

Pour surmonter les défis inhérents au comptage de véhicules à partir de vidéos, une gamme de méthodes a été développée, utilisant des techniques avancées de traitement d'image et d'apprentissage automatique pour améliorer la précision de la détection et du suivi. Bien que ces avancées aient été significatives, le comptage de véhicules demeure un domaine de recherche dynamique, offrant de nombreuses opportunités pour de nouvelles innovations et améliorations. En complément, une revue des systèmes basés sur la détection et ceux basés sur la densité révèle des différences notables : les méthodes basées sur la densité [52, 53] estiment le nombre d'objets à partir de caractéristiques extraites des images, tandis que les méthodes basées sur la détection [54-59] se concentrent sur l'identification précise des objets au sein des images.

Dai et al. (2019) [60] ont mis en avant une méthode structurée en trois étapes pour le comptage de véhicules. La première étape consiste en la détection des véhicules. Cette détection est cruciale car elle sert de base pour les étapes suivantes. La détection des véhicules peut être réalisée à l'aide de diverses techniques, allant de la simple soustraction d'arrière-plan à des méthodes plus avancées basées sur l'apprentissage automatique. Une fois les véhicules détectés, la deuxième étape consiste à les suivre tout au long de la vidéo. Le suivi est essentiel pour s'assurer que chaque véhicule est compté une seule fois et pour éviter les doubles comptages. Enfin, la troisième étape traite des trajectoires des véhicules pour obtenir un compte précis. Cette étape est souvent réalisée à l'aide de lignes virtuelles ou de zones d'intérêt où les véhicules sont comptés lorsqu'ils les traversent.

Liu et al. (2017) [61] ont adopté une approche différente en utilisant la boucle virtuelle et la ligne de détection pour le comptage de mouvements multiples. Cette méthode est particulièrement utile pour les scénarios où les véhicules peuvent changer d'identité lors du suivi sur de longues distances, comme dans les intersections ou les zones de forte congestion.

Le comptage de véhicules est essentiel pour la gestion du trafic et la planification urbaine [62]. La figure 6 [63] illustre le processus général d'un système de comptage basé sur la vidéo. Xiang et al (2018) [64] ont adopté cette approche pour le comptage de véhicules à partir de vidéos aériennes, une méthode particulièrement utile pour surveiller le trafic sur de grandes étendues, comme les autoroutes ou les zones rurales. Song et al. Ont combiné le puissant algorithme de détection YOLOv3 avec l'algorithme ORB pour le suivi [65], offrant une solution robuste pour le comptage dans des conditions variées.

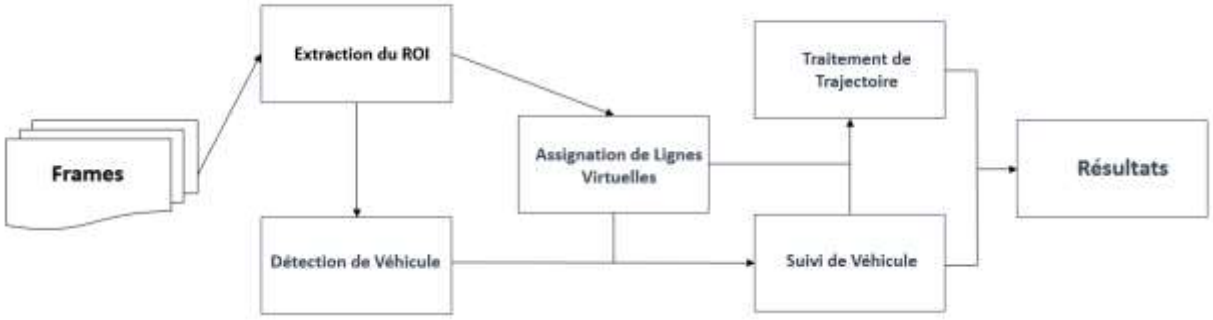


Figure 6 Flux de travail du système de comptage de véhicules

Liu et al (2017) [61] ont introduit un modèle adaptatif, offrant une flexibilité pour différents scénarios urbains.

Rosas-Arias et al. (2019) [66] ont contribué à la conception d'un système de suivi inter-images pour le comptage de véhicules traversant des barrières virtuelles, en utilisant une méthode basée sur la décomposition en valeurs singulières pour classifier les vecteurs propres.

Lucking et al. (2020) [67] ont développé un système de comptage de véhicules en temps réel en utilisant des réseaux neuronaux profonds intégrés dans des dispositifs embarqués, avec des taux de rappel et de précision élevés.

Pham et al. (2020) [68] ont élaboré un processus complet pour détecter, classifier et compter différents types de véhicules, atteignant un taux de détection élevé en conditions diurnes.

En conclusion, le comptage de véhicules est un domaine en constante évolution. Avec l'innovation continue et l'adoption croissante de l'apprentissage profond, on peut s'attendre à voir des méthodes encore plus précises et robustes à l'avenir.

3. Matériel et Systèmes Embarqués en Deep Learning

3.1. Architectures de Systèmes Informatiques pour le Deep Learning

Dans l'univers du deep learning, différentes architectures sont adoptées en fonction des exigences spécifiques du projet. Une compréhension claire de ces architectures - centralisée, distribuée, edge complète et edge partielle - est essentielle pour déterminer la plus adaptée à un contexte donné. Cette section présente une étude comparative de ces architectures, mettant en lumière leurs avantages et inconvénients.

3.1.1. Architecture Centralisée

L'architecture centralisée traite et stocke toutes les données sur un serveur ou un cluster de serveurs central.

Avantages : Meilleure gestion des ressources, mise à jour facile, et centralisation de la maintenance.

Inconvénients : Latence élevée, risques liés à la sécurité des données, et coûts élevés de bande passante.

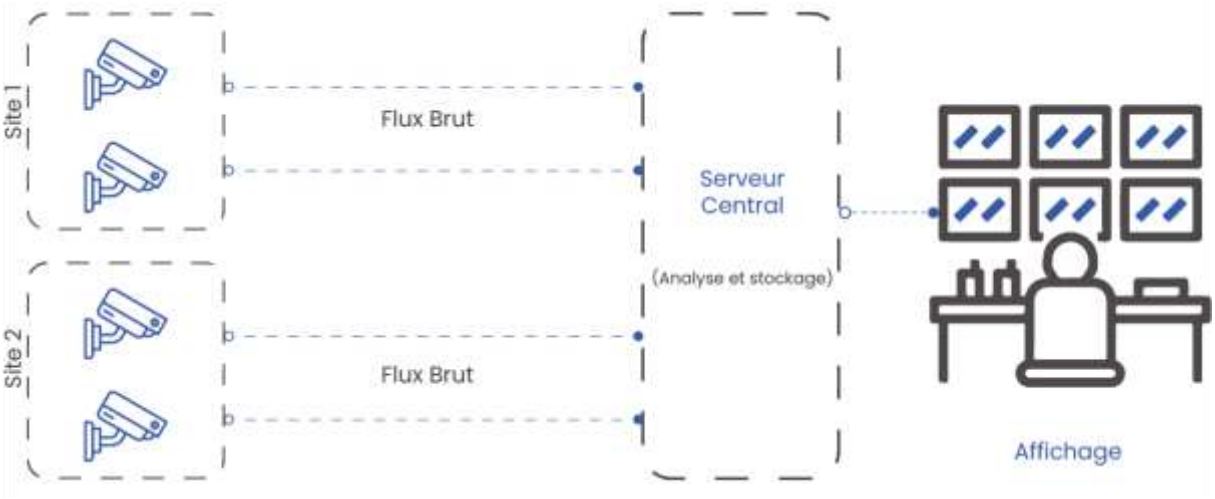


Figure 7 Architecture centralisée pour le traitement et l'analyse de flux de données

3.1.2. Architecture Distribuée

Dans cette approche, les calculs et le stockage des données sont répartis sur plusieurs emplacements ou nœuds.

Avantages : Réduction de la latence, amélioration de la résilience et meilleure gestion du trafic de données.

Inconvénients : Complexité de gestion et de maintenance, et défis de synchronisation des données.

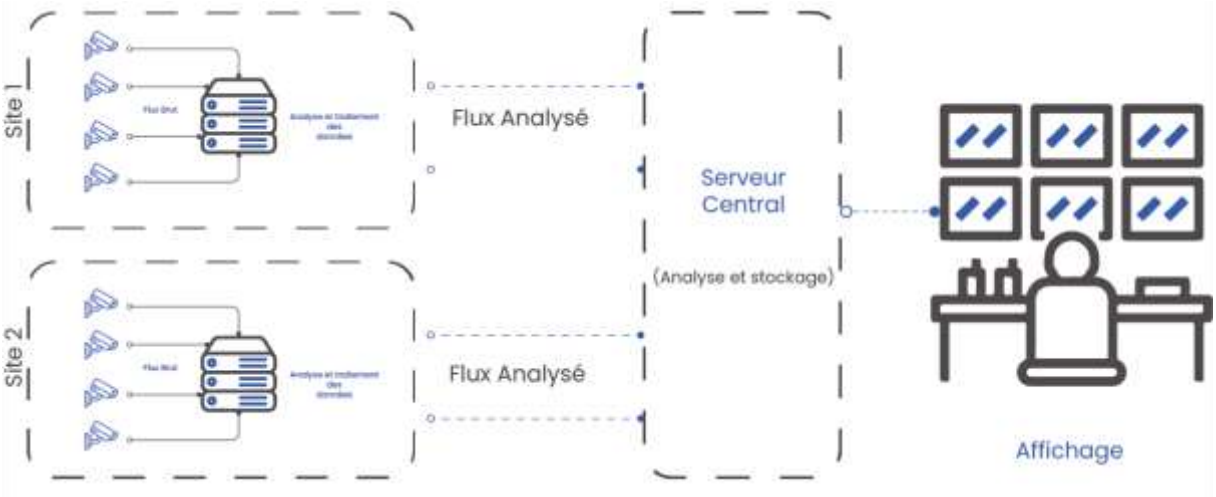


Figure 8 Architecture Distribuée pour le traitement et l'analyse de flux de données

3.1.3. Architecture Edge Partielle

Une approche hybride où une partie du traitement est effectuée à la périphérie, tandis que des tâches plus lourdes sont gérées par un serveur central ou distribué.

Avantages : Équilibre entre réactivité et puissance de traitement, et optimisation des coûts de bande passante.

Inconvénients : Complexité de mise en œuvre et nécessité d'une coordination efficace entre l'Edge et le centre.

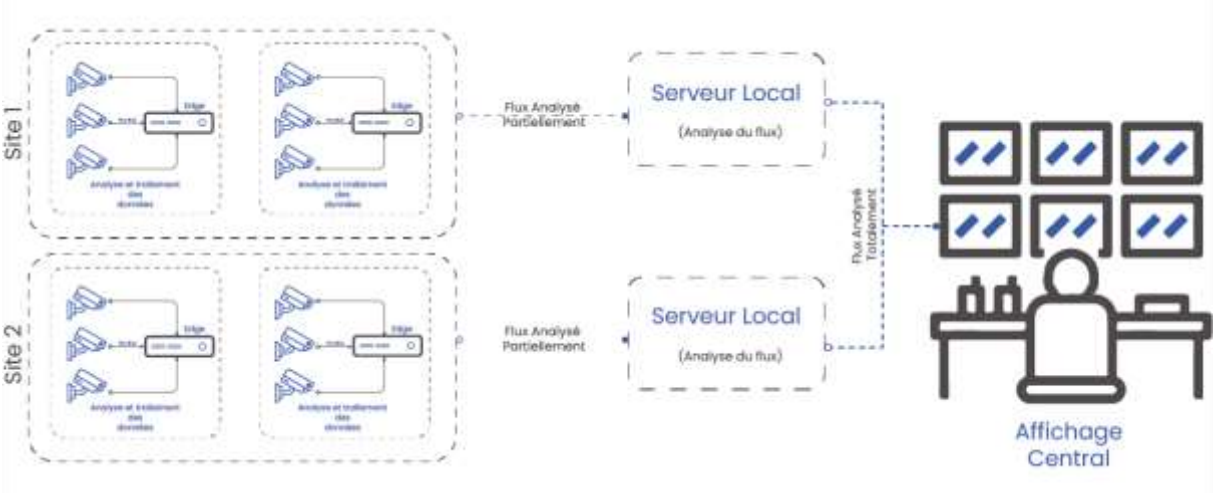


Figure 9 Architecture Edge Partielle pour le traitement et l'analyse de flux de données

3.1.4. Architecture Edge Complète

L'architecture Edge complète traite toutes les données à la périphérie du réseau, près de la source des données.

Avantages : Réactivité accrue grâce au traitement en temps réel sur le terrain. Sécurité renforcée car les vidéos brutes restent sur le site, Économies significatives de bande passante.

Inconvénients : Capacité de traitement et de stockage limitée, et défis de gestion de la sécurité.

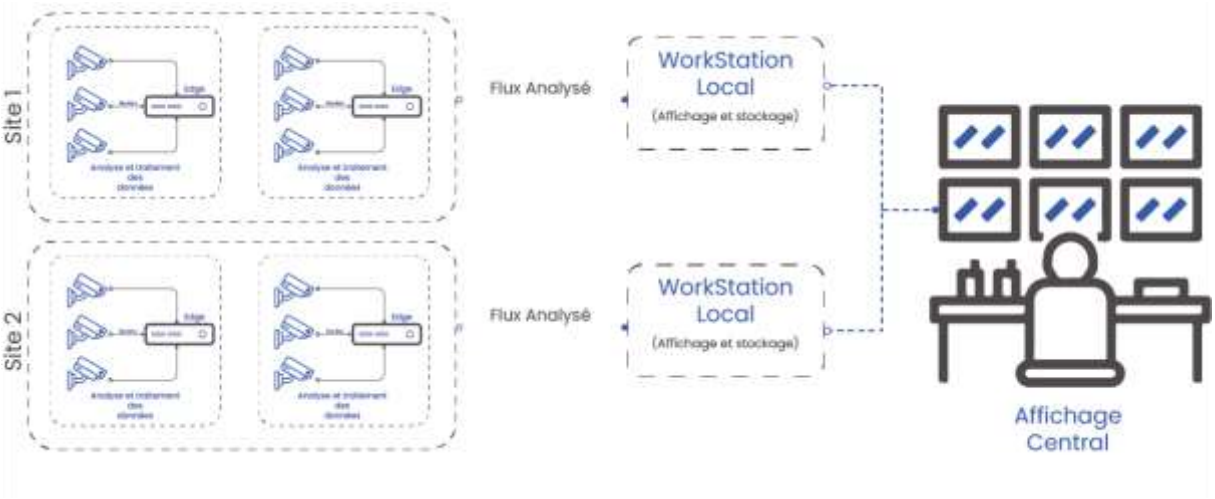


Figure 10 Architecture Complète pour le traitement et l'analyse de flux de données

3.1.4.1. Adoption de l'Architecture Edge Complète

Après analyse comparative, notre choix se porte sur l'architecture Edge complète pour le déploiement de solutions basées sur le Deep Learning. Les principaux avantages de cette approche incluent la capacité à traiter les données en temps réel avec une latence minimale, ce qui est crucial dans les applications nécessitant des réponses rapides, telles que les systèmes de reconnaissance en temps réel ou les applications de surveillance intelligente.

Bien que cette architecture présente des défis, notamment en termes de capacité de traitement et de stockage, les avantages en termes de réactivité et de réduction des coûts de transmission de données l'emportent dans le contexte de mon projet. De plus, les progrès récents dans la technologie des systèmes embarqués ont considérablement augmenté les capacités de traitement à la périphérie, rendant cette option plus viable que jamais.

3.2. Infrastructure Matérielle Clé pour le Deep Learning : Ordinateurs, Processeurs et Stockage

Après avoir examiné les différentes architectures de déploiement pour les solutions basées sur le Deep Learning, il est crucial de se pencher sur l'infrastructure matérielle qui sous-tend ces technologies. Cette section se concentre sur les composants matériels clés qui rendent possibles ces architectures avancées, en mettant en lumière leur rôle et leur importance dans la mise en œuvre de solutions de Deep Learning performantes et efficientes.

Dans le domaine du Deep Learning, la sélection et l'utilisation efficace du matériel informatique sont cruciales pour le succès des projets. Cette section se consacre aux différents outils matériels qui forment le socle des solutions de Deep Learning, en mettant l'accent sur les ordinateurs, les processeurs, les GPU et les systèmes de stockage, essentiels à l'entraînement et à l'inférence des modèles de Deep Learning.

Ordinateurs et Serveurs pour le Deep Learning

Exploration des types d'ordinateurs utilisés dans le Deep Learning, allant des ordinateurs de bureau, adaptés aux projets de petite envergure, aux serveurs et clusters de serveurs, nécessaires pour les projets à grande échelle demandant d'importantes ressources informatiques.

Processeurs et GPU : Accélération des Calculs

Discussion sur le rôle des processeurs (CPU) et des cartes graphiques (GPU) dans l'accélération des calculs de Deep Learning. Tandis que les CPU sont utilisés pour les tâches générales, les GPU sont privilégiés pour les opérations parallèles complexes, rendant l'entraînement des modèles plus rapide et plus efficace.

Systèmes de Stockage pour Données d'Entraînement

Analyse des systèmes de stockage nécessaires pour gérer les grandes quantités de données d'entraînement. Des solutions comme les systèmes de stockage en cluster (exemple : Hadoop) et les systèmes de stockage en mémoire (exemple : Redis) sont discutées pour leur efficacité dans le stockage et l'accès aux données.

GPU Populaires en Deep Learning

Présentation des GPU les plus couramment utilisés dans le Deep Learning, avec un accent particulier sur les produits de NVIDIA, tels que les séries GeForce, Tesla et Quadro, ainsi que les cartes graphiques AMD Radeon. L'importance de ces GPU dans divers projets de recherche et applications est soulignée à travers des exemples d'études scientifiques.

Processeurs pour Applications de Deep Learning

Examen des différents types de processeurs utilisés dans le Deep Learning, y compris les séries Intel Xeon, AMD Ryzen, ARM et IBM Power. Leur rôle dans les tâches de prétraitement des données, la gestion de la mémoire et la distribution des calculs est discuté.

3.3. Les SBC pour l'apprentissage en profondeur : une liste des cartes électroniques les plus populaires

3.3.1. Définition des cartes SBC

Les ordinateurs à carte unique (Single Board Computers, SBC) sont des systèmes informatiques intégrés sur un seul circuit imprimé. Compacts et économiques, ils intègrent un processeur, de la mémoire, du stockage, ainsi que des ports d'entrée/sortie et une connectivité réseau. Ces dispositifs offrent une solution abordable et peu encombrante par rapport aux ordinateurs de bureau ou portables traditionnels.

Leur taille réduite et leur coût avantageux rendent les SBC idéaux pour des projets de bricolage, de robotique, d'automatisation domestique, et d'autres applications nécessitant un encombrement minimal et une efficacité maximale. Ils trouvent aussi leur place dans des contextes industriels pour la surveillance et l'automatisation des processus.

Ces ordinateurs peuvent supporter divers systèmes d'exploitation comme Linux, Android ou Windows. Ils disposent typiquement de ports USB, Interface Multimédia Haute Définition

(HDMI), Ethernet et audio pour une connectivité étendue. Certains modèles incluent également des processeurs graphiques, les rendant aptes à des applications de traitement d'images et de Deep Learning.

3.3.2. Deep Learning sur les SBC pour l'Edge Complet : Cartes Électroniques Compatibles

Le Raspberry Pi 4 représente un choix populaire de Single Board Computer (SBC) pour les projets de Deep Learning à petite échelle. Bien qu'il ne rivalise pas en puissance avec certaines cartes électroniques spécialement conçues pour l'apprentissage profond, il demeure capable de gérer des tâches telles que la reconnaissance d'images, la détection d'objets et la classification de texte.

Il est possible d'y installer des frameworks de Deep Learning tels que TensorFlow, PyTorch et Keras. Toutefois, étant donné ses limitations en termes de puissance de calcul, il est conseillé de privilégier des modèles de Deep Learning plus légers et de réduire le volume des itérations d'entraînement.

Le Raspberry Pi 4 trouve son utilité dans diverses applications d'apprentissage en profondeur, telles que :

Reconnaissance d'Images : Il peut être employé pour identifier des panneaux de signalisation, reconnaître des visages ou interpréter des gestes.

Détection d'Objets : Utilisé pour repérer des objets dans des images ou des séquences vidéo, le Raspberry Pi 4 offre une plateforme pratique pour ce type d'applications [69].

Composant	Details
SOC	Broadcom BCM2711B0 quad-core A72 (ARMv8-A) 64-bit @ 1.5GHz
GPU	Broadcom VideoCore VI
Networking	2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN
RAM	1GB, 2GB, or 4GB LPDDR4 SDRAM
Bluetooth	Bluetooth 5.0, Bluetooth Low Energy (BLE)
GPIO	40-pin GPIO header, populated
Storage	microSD
Ports	2 x micro-HDMI 2.0, 3.5 mm analogue audio-video jack, 2 x USB 2.0, 2 x USB 3.0, Gigabit Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
Dimensions	88 mm x 58 mm x 19.5 mm, 46 g

Tableau 1 Spécifications Techniques du Raspberry Pi



Figure 11 Raspberry Pi (RPi)

Le NVIDIA Jetson Nano est un ordinateur monocarte (SBC) conçu pour les applications d'intelligence artificielle (AI) et d'apprentissage profond. Il est équipé d'un CPU ARM Cortex-A57 quadricœur 64 bits et d'un GPU NVIDIA Maxwell avec 128 cœurs CUDA. Cela permet au Jetson Nano d'exécuter efficacement les modèles d'apprentissage profond, offrant des performances élevées tout en maintenant une faible consommation d'énergie. Compatible avec un vaste éventail de cadres d'apprentissage profond tels que TensorFlow, PyTorch, Keras, MXNet et Caffe, les développeurs peuvent entraîner et déployer des modèles d'apprentissage

profond directement sur le Jetson Nano. Cet appareil est particulièrement adapté aux tâches d'apprentissage profond, y compris la reconnaissance d'objets, la détection d'objets, la segmentation sémantique, la reconnaissance vocale et la navigation autonome. Il est utilisé dans divers projets de développement tels que la robotique, les drones et les caméras intelligentes. Par exemple, une application de l'apprentissage profond sur le Jetson Nano est le contrôle gestuel basé sur la vision pour les drones. Dans ce projet, le Jetson Nano traite un algorithme d'apprentissage profond pour reconnaître les gestes à travers des entrées visuelles, permettant la navigation du drone pilotée par gestes [70].

Composant	Spécification
GPU	GPU 128 cœurs à architecture NVIDIA Maxwell™
CPU	Processeur quad-core ARM® Cortex®-A57 MPCore
Mémoire	4 Go 64 bits LPDDR4
Stockage	MicroSD (carte non incluse)
Encodage vidéo	1x 4K @ 30
Décodage vidéo	1x 4K @ 60
Mise en réseau	Gigabit Ethernet, M. 2 Key E
Caméra	Doubles connecteurs de caméra MIPI CSI-2 à 15 broches et 2 canaux
Affichage	1x HDMI 2.0, 1x DP 1.2
Autres E/S	Connecteur à 40 broches (UART, SPI, I2S, I2C, PWM, GPIO) Connecteur d'automatisation à 12 broches Connecteur de ventilation à 4 broches Connecteur POE à 4 broches Prise d'alimentation CC
USB	4 connecteurs USB 3.0 Type-A Connecteur USB 2.0 Micro-B
Boutons	Alimentation, de récupération forcée et de réinitialisation
Spécifications mécaniques	100 mm x 79 mm x 30,21 mm (La hauteur inclut la carte de référence, le module et la solution thermique.)
Prix	À partir de 99$

Tableau 2 Spécifications Techniques du Jetson Nano



Figure 12 Carte Jetson Nano

La carte Coral Dev est un ordinateur monopuce (SBC) conçu spécialement pour exécuter des tâches d'inférence dans le domaine de l'apprentissage profond. Doté d'une unité de traitement de tenseur (TPU) de Google Edge, ce processeur est optimisé pour réaliser de l'inférence en apprentissage profond tout en conservant une efficacité énergétique élevée.

Ce matériel est compatible avec des frameworks de pointe dans le domaine de l'apprentissage profond, tels que TensorFlow et TensorFlow Lite, offrant aux développeurs la possibilité de former et d'implémenter des modèles de deep learning directement sur le Coral Dev Board.

Le board se prête particulièrement bien aux tâches d'inférence d'apprentissage profond, comme la reconnaissance d'images, la détection d'objets, la classification de texte et la reconnaissance vocale, et trouve son utilité dans le développement de dispositifs tels que des caméras intelligentes, des systèmes de sécurité et des robots.

Quelques applications pratiques incluent :

Estimation de la pose avec MoveNet : Cette application démontre l'utilisation du modèle MoveNet pour identifier les postures humaines à partir d'images. Elle est compatible avec le modèle 'Lightning' pour une détection rapide ou le modèle 'Thunder' pour une précision accrue.

Suivi d'objet en vidéo : Cet exemple utilise un flux vidéo pour suivre les objets détectés, attribuant à chacun un identifiant unique persistant. Le script de détection permet de choisir le programme de suivi désiré, incluant l'option de suivi par tri.

Détection d'objet avec MediaPipe : Utilise MediaPipe pour la détection d'objets en temps réel via une caméra [71].

Composant	Spécifications
CPU	NXP i.MX 8M SoC (quad Cortex-A53, Cortex-M4F)
GPU	Integrated GC7000 Lite Graphics
ML accelerator	Google Edge TPU coprocessor: 4 TOPS (int8); 2 TOPS per watt
RAM	1 or 4 GB LPDDR4
Flash memory	8 GB eMMC, MicroSD slot
Wireless	Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz) and Bluetooth 4.2
USB	Type-C OTG; Type-C power; Type-A 3.0 host; Micro-B serial console
LAN	Gigabit Ethernet port
Audio	3.5mm audio jack (CTIA compliant); Digital PDM microphone (x2); 2.54mm 4-pin terminal for stereo speakers
Video	HDMI 2.0a (full size); 39-pin FFC connector for MIPI-DSI display (4-lane); 24-pin FFC connector for MIPI-CSI2 camera (4-lane)
GPIO	3.3V power rail ; 40 - 255 ohms programmable impedance; ~82 mA max current
Power	5V DC (USB Type-C)
Availability	Australia, Japan, New Zealand, Taiwan, European Union (except France, Czech Republic), Ghana, Hong Kong, India, Oman, Philippines, Singapore, South Korea, Thailand, United States
Prix	$129.99

Tableau 3 Spécifications Techniques de la carte Coral Dev

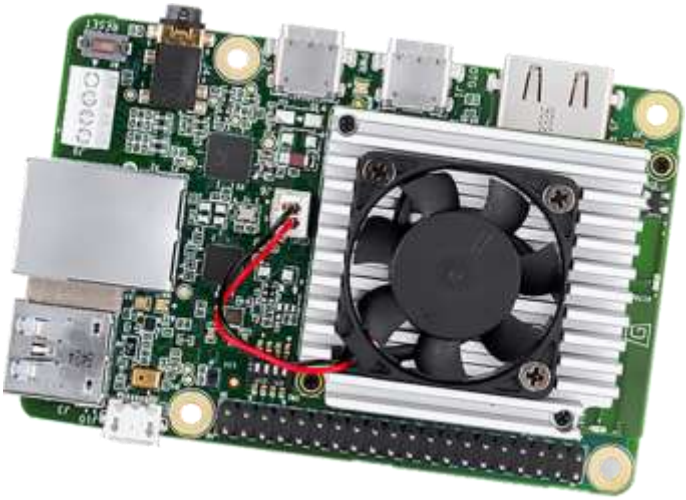


Figure 13 Carte Coral Dev

Après avoir analysé les différentes cartes électroniques embarquées et leur capacité à supporter l'apprentissage profond, nous allons maintenant découvrir comment ces technologies sont mises en œuvre dans les dernières avancées et innovations. Nous verrons des exemples concrets de recherches actuelles qui utilisent ces cartes pour faire progresser le champ de l'AI.

3.4. Avancées Technologiques et Innovations

Dans l'univers complexe des infrastructures routières, les systèmes embarqués émergent comme des éléments fondamentaux pour la gestion et la régulation du trafic. Ces dispositifs, alliant rapidité de traitement et adaptabilité, jouent un rôle essentiel dans la supervision et la prévention des infractions routières. De plus, grâce à leur capacité de comptage précis des véhicules, ils fournissent une vision claire du flux de circulation, facilitant ainsi la détection des congestions et l'ajustement des mesures de contrôle.

L'évolution croissante de ces systèmes, associée à l'innovation constante des algorithmes de détection, ouvre la voie à une nouvelle ère de surveillance routière. Cette combinaison permet une identification quasi instantanée des contrevenants tout en autorisant une régulation dynamique du trafic. L'intégration de ces technologies dans nos réseaux routiers témoigne de notre ambition : celle de créer des routes où la sécurité prime, où chaque infraction est rapidement repérée et gérée.

Lorsque l'on se penche sur le comptage des véhicules, il est clair que les systèmes embarqués ont joué un rôle central dès les débuts de cette discipline. Grâce à eux, des mécanismes automatisés ont vu le jour, générant des statistiques de trafic fiables et détaillées. Au fil du temps, cette mine d'informations s'est avérée cruciale pour optimiser les infrastructures,

renforcer la sécurité et approfondir notre compréhension des mouvements routiers.

Ce survol historique du comptage routier illustre parfaitement la révolution apportée par les systèmes embarqués en matière de surveillance. Ces innovations ont non seulement modifié notre approche de la gestion du trafic, mais elles ont également ouvert la voie à des avancées technologiques continuelles, redéfinissant la mobilité et la sécurité sur nos routes.

De précédentes recherches et projets liés à cette problématique ont été identifiés dans la littérature, et nous aimerions en mentionner quelques-uns.

Abuelgasim et al. (2023) [72] ont présenté un système de comptage de véhicules qui utilise l'algorithme YOLOv5 pour la détection d'objets et Deep SORT pour le suivi. Leur système a été évalué sur une carte Nvidia Jetson Nano, capable de traiter 15 images par seconde en temps réel. Cependant, il convient de noter que leur évaluation s'est limitée à des vidéos, alors que des conditions réelles incluent des variations d'éclairage et météorologiques. Par conséquent, pour garantir une performance fiable, il est essentiel de former un modèle robuste capable de gérer ces diverses conditions.

Duc-Liem et al. En (2021) [73] ont présenté un système rentable et efficace intégrant des modèles de détection d'objets pour la détection, le suivi et le comptage des véhicules. Ils ont créé un ensemble de données représentant les conditions de circulation au Vietnam et ont évalué divers modèles d'apprentissage profond sur des plates-formes telles que la carte Coral Dev (avec TPU) et la carte Nvidia Jetson Nano (avec GPU). Cependant, leur étude a identifié la nécessité d'améliorer le temps d'inférence du modèle pour une performance optimale. Ainsi, notre travail vise à optimiser le temps d'inférence pour rendre le système plus pratique pour le contrôle du trafic.

Valladares et al. En (2021) [74] ont évalué les performances des cartes Nvidia pour le comptage en temps réel des véhicules à Quito, en Équateur, en utilisant l'apprentissage automatique. Ils ont considéré des facteurs tels que l'utilisation des ressources (GPU et CPU), la consommation d'énergie, la température de la carte et l'utilisation de la RAM (Mémoire à Accès Aléatoire) pendant le traitement. Ils ont également souligné la nécessité d'améliorer la précision du modèle en prenant en compte divers paramètres, comme la position de la caméra, l'éclairage, la taille des véhicules et le flux de circulation, pour s'adapter à toutes les conditions.

Yan Han et al. En (2017) [75] ont développé un système d'aide à la conduite basé sur des cartes

embarquées, telles que la Nvidia Jetson. Le système a utilisé des techniques de traitement d'image pour localiser et classer les panneaux de signalisation. Bien que le système ait atteint une précision de détection élevée, sa vitesse de rafraîchissement était limitée à 1,6 (FPS). Par conséquent, un modèle plus rapide était nécessaire pour une utilisation avec la carte Jetson.

Zhihui Wang et al. En (2020) [76] ont proposé une solution intégrée combinant détection, suivi et modélisation de trajectoire pour la détection et le comptage des véhicules. Ils ont utilisé une approche de modélisation à rebours de type GMM (Modèle de Mélanges Gaussiens), ainsi qu'un détecteur basé sur l'apprentissage automatique pour améliorer la précision. Cependant, leur algorithme présentait encore des limites, notamment des détections manquantes et des faux positifs dus à divers facteurs. Ainsi, des algorithmes plus robustes étaient nécessaires pour surmonter ces problèmes.

4. Conclusion

Ce chapitre nous a conduits à une compréhension exhaustive des techniques de détection d'objets et des infrastructures nécessaires au Deep Learning. A travers ce chapitre, nous avons pu identifier les méthodes de détection d'objets en mouvement à la lumière des exemples de la littérature, ce qui a attiré notre attention pour chercher et proposer des algorithmes adaptés à nos objectifs de recherche.

L'analyse menée afin d'identifier les limites des architectures traditionnelles, balayant un spectre complet d'infrastructures — centralisées, distribuées, et autres modalités intermédiaires — et qui nous a mené à l'Edge-computing. Cette décision a été suite à des réflexions, émergeant non seulement de l'examen des besoins pratiques mais aussi de la projection des tendances technologiques futures.

De manière subtile mais déterminante, nous avons également sondé les profondeurs des composantes matérielles indispensables à l'apprentissage en profondeur. La réflexion sur les ordinateurs, processeurs et systèmes de stockage a renforcé notre conviction en choisissant le NVIDIA Jetson Nano, une plateforme qui incarne l'alliance entre la performance de calcul et l'agilité requise pour le Deep Learning en Edge computing.

En résumé, ce chapitre consolide notre choix de méthodes pour la recherche. Ce choix résulte d'un examen attentif et d'une comparaison des différentes techniques de Deep Learning, ainsi que des études de cas qui illustrent leur mise en pratique. Notre objectif a été d'identifier et de choisir des outils et des approches qui tirent le meilleur parti du Deep Learning pour les

systèmes embarqués modernes.

Chapitre 2

Détection des objets en mouvement et résultats expérimentaux

Chapitre 2 : Détection des objets en mouvement et résultats expérimentaux

1. Introduction

La détection d'objets en mouvement dans les séquences vidéo est devenue une partie fondamentale de la vision par ordinateur, avec des implications profondes dans des domaines aussi variés que la surveillance urbaine, l'assistance à la conduite automobile et les interfaces homme-machine avancées. Ces applications nécessitent une précision et une fiabilité élevées, car les décisions basées sur cette détection peuvent entraîner des conséquences significatives, allant de la sécurité des personnes à la fluidité du trafic urbain.

Historiquement, les méthodes traditionnelles, telles que la soustraction de l'arrière-plan et la différence de frame, ont été largement adoptées en raison de leur simplicité conceptuelle et de leur facilité de mise en œuvre. Cependant, elles ont montré leurs limites, notamment en termes de sensibilité aux variations d'éclairage et aux mouvements rapides. De plus, bien que le flux optique soit une méthode puissante pour détecter le mouvement, sa complexité computationnelle la rend moins adaptée aux applications en temps réel.

Devant ces enjeux, les méthodes basées sur l'apprentissage automatique sont apparues comme une solution robuste et innovante. Le recours à des algorithmes de classification s'est révélé particulièrement efficace pour identifier les objets en mouvement par rapport aux éléments statiques de la scène. Grâce à sa capacité à gérer des espaces de caractéristiques complexes et à établir des limites de décision solides, plusieurs algorithmes d’apprentissage automatique se sont imposés comme des outils de choix pour relever ce défi.

Dans ce contexte, nous nous sommes intéressés à l’utilisation de la DT pour la segmentation des scènes de trafic. Cette méthode, axée sur la géométrie, divise l'espace visuel en un réseau de triangles, offrant une approche unique et détaillée pour analyser l'espace urbain. Contrairement aux techniques traditionnelles de segmentation, telles que la méthode basée sur les niveaux de gris Otsu's Thresholding, la DT se concentre sur la structure spatiale elle-même. Cette spécificité permet une décomposition précise de la scène en régions d'intérêt, adaptée aux défis complexes de l'analyse du trafic en milieu urbain. Grâce à cette technique, notre système

gagne en précision et en capacité à discerner les différents éléments dans l'espace urbain, ce qui est essentiel pour une gestion efficace du trafic.

Après la segmentation, l'étape d'extraction des caractéristiques transforme chaque région d'intérêt en un ensemble de données que l'algorithme de classification SVM va utiliser comme entrée pour classifier ces régions. Ces caractéristiques peuvent inclure des informations texturales, des gradients de couleur et d'autres éléments visuels qui aident l'algorithme de classification identifier les objets en mouvement.

Ce chapitre détaillera chaque étape de cette approche, des fondements théoriques aux choix pratiques effectués lors de la mise en œuvre. De plus, une série de résultats expérimentaux sera présentée, illustrant la performance de cette méthode dans diverses conditions et scénarios routiers.

2. Détection des objets en mouvement

Chaque technique de suivi nécessite d'abord une phase de détection de l'objet, soit à chaque image, soit lors de la première apparition de l'objet dans la séquence vidéo. Ce volet du suivi d'objet vise à repérer les entités pertinentes dans la vidéo et à assembler les pixels correspondant à ces entités. Étant donné que les éléments en déplacement sont souvent la clé de l'analyse, la majorité des approches privilégient la détection de ces éléments en mouvement.

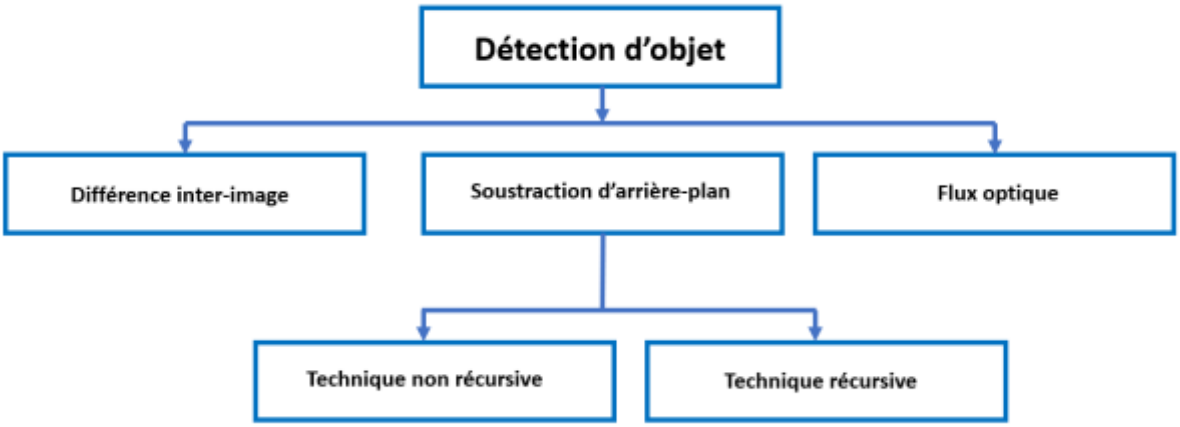


Figure 14 Techniques de détections

2.1. Soustraction d'arrière-plan

La soustraction d'arrière-plan est une technique couramment utilisée en traitement d'images et vidéos pour distinguer les objets en mouvement (premier plan) de l'arrière-plan statique. Cette approche est centrale à la détection et à la segmentation d'objets en mouvement dans une

séquence vidéo.

La formule fondamentale de la soustraction d'arrière-plan repose sur la différence entre une image vidéo courante I (x, y, t) et un modèle d'arrière-plan B (x, y, t) :

$$D(x, y, t) = |I(x, y, t) - B(x, y, t)| \tag{2.1}$$

Où D (x, y, t) représente l'image de différence à la position (x, y) au temps t.

En fonction du seuil τ défini, on peut déterminer si un pixel appartient à un objet en mouvement ou à l'arrière-plan :

$$f(x) = \begin{cases} 1, & D(x, y, t) \geq \tau \\ 0, & \text{si non} \end{cases} \tag{2.2}$$

Le modèle d'arrière-plan B (x, y, t) peut être soit statique (constant dans le temps) ou dynamique (s'adaptant aux changements dans la scène). Les techniques dynamiques peuvent être classées en méthodes récursives et non récursives [77] :

a) Techniques récursives : Ces techniques mettent à jour continuellement le modèle d'arrière-plan basé sur les frames récents, sans nécessiter de mémoire tampon pour stocker plusieurs frames précédents.

b) Techniques non récursives : Les techniques non récursives emploient généralement une stratégie de fenêtre glissante pour l'évaluation de l'arrière-plan. Elles conservent un ensemble de frames vidéo antérieures dans un tampon et déduisent l'image d'arrière-plan en examinant la fluctuation temporelle de chaque pixel au sein de ce tampon. Ces méthodes sont particulièrement adaptatives car leur analyse se limite aux frames conservés dans le tampon, sans s'appuyer sur un historique plus ancien. Cependant, une telle approche peut nécessiter une importante capacité de stockage, notamment si un vaste tampon s'avère indispensable face à des mouvements lents.

Selon cette approche, un pixel à la position (a, b) de l'image actuelle est considéré comme appartenant au premier plan si la condition de l'équation (2.3) est remplie :

$$|IT(a, b) - BT(a, b)| \geq Th \tag{2.3}$$

Ici, Th représente un seuil préétabli, BT désigne l'image d'arrière-plan et IT l'image en cours. Après avoir déterminé les pixels du premier plan, une opération de fermeture morphologique est appliquée, suivie de la suppression des zones de petite dimension (voir figure 15).

Bien que ces techniques parviennent généralement à identifier la majorité des pixels associés aux mouvements, même lorsqu'ils sont stationnaires, elles demeurent sensibles aux variations dynamiques. Par exemple, lorsqu'un objet immobile révèle l'arrière-plan (comme une voiture quittant une place de stationnement) ou en cas de changement d'éclairage abrupt.

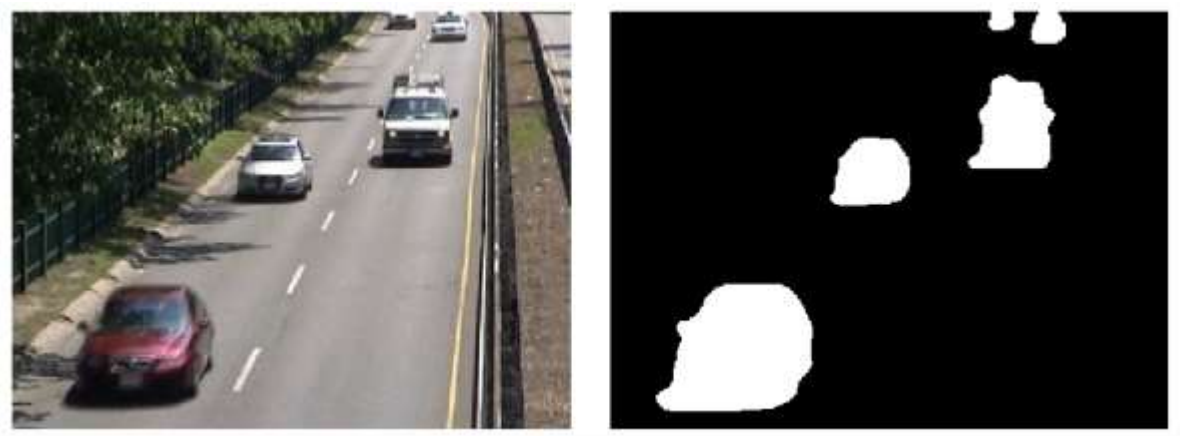


Figure 15 Présentation de la méthode de soustraction d'arrière-plan

2.2. Différence inter-image

La différence inter-image, aussi connue comme la différenciation temporelle, est l'une des techniques les plus fondamentales et intuitives pour détecter le mouvement dans une séquence vidéo. Elle est basée sur la prémisse simple que tout changement dans l'intensité d'un pixel entre deux images consécutives indique un mouvement.

Cette approche a été mise en évidence dans des travaux antérieurs, comme référencé dans [78]. Fondamentalement, la première image de la séquence, souvent appelée image de différence de premier ordre (FODP), est distincte des autres images. Elle représente typiquement la partie immobile ou statique de la scène.

L'avantage majeur de cette technique est qu'elle ne nécessite pas de modèle d'arrière-plan préétabli. Au lieu de cela, elle se concentre sur l'analyse des variations temporelles de l'intensité lumineuse de chaque pixel entre deux images successives. Mathématiquement, cette différence est exprimée par l'équation :

$$\Delta(x,y) = |I(x,y,t) - I(x,y,t-1)| \quad (2.4)$$

Où Δ représente la différence en intensité pour un pixel donné à la position (x, y) entre deux moments successifs t et t-1.

Toutefois, compte tenu des variations inhérentes d'intensité lumineuse et des bruits introduits

par les conditions environnementales ou les imperfections des capteurs de la caméra, il est souvent nécessaire d'introduire un seuil, désigné par τ. Ce seuil permet de déterminer si le changement observé est suffisamment significatif pour être considéré comme un mouvement. L'utilisation d'un seuil est toutefois délicate, car son choix peut influencer les résultats. Ainsi, l'image résultante M (x, y), qui identifie les zones en mouvement, est définie par :

$$M(x,y) = \begin{cases} 1 si \Delta(x,y) \geq \tau \\ \text{si non } 0 \end{cases} \quad (2.5)$$

La différence temporelle est particulièrement efficace dans des environnements dynamiques où les changements sont fréquents. Cependant, elle présente des limitations. Par exemple, elle peut ne pas détecter des mouvements très lents ou saccadés, ou encore des arrêts temporaires d'un objet en mouvement. De plus, en cas de redondance dans les frames de la vidéo, la robustesse de cette méthode peut être mise à l'épreuve.

2.3. Flux Optique pour la Détection d'Objets en Mouvement

Le flux optique se réfère à la représentation du mouvement apparent des objets, des surfaces et des bords dans une scène visuelle. Ce mouvement est provoqué par la variation relative entre un observateur (ou une caméra) et la scène elle-même. En termes plus simples, il décrit comment chaque pixel d'une image se déplace d'un cadre à l'autre.

Le concept de flux optique est fondamental pour la détection d'objets en mouvement, car il fournit une description précise du mouvement local dans une séquence vidéo. Lorsque l'on calcule l'arène du flux optique de l'image, on analyse le mouvement à chaque point de l'image. Ensuite, on regroupe ces informations en fonction des apparences distinctes dans l'image, permettant ainsi d'isoler des régions ou des objets spécifiques en mouvement.

Mathématiquement, la méthode du flux optique est basée sur l'approximation de Taylor, utilisant des dérivées partielles par rapport aux coordonnées spatiales et temporelles. Si I (x, y, t) représente l'intensité de l'image à la position (x, y) et à l'instant t, la constance de la luminance au fil du temps nous donne [101] :

$$I(x,y,t) = I\left(x + \delta_{x,y} + \delta_{y,t} + \delta_{t}\right) \quad (2.6)$$

En développant cela avec une série de Taylor, on peut arriver à l'équation fondamentale du flux optique :

$$I_{xu} + I_{yv} + I_t = 0 \qquad (2.7)$$

Où I x et I y sont les dérivées spatiales de l'image et u et v sont les composantes du flux optique en x et y respectivement.

Malgré sa capacité à fournir des informations de mouvement détaillées et à isoler efficacement les objets en mouvement de l'arrière-plan, la méthode du flux optique présente certains défis. En raison de la nécessité de calculer des dérivées, cette méthode nécessite une grande quantité de calculs, la rendant parfois inadaptée aux applications en temps réel. De plus, elle est sensible au bruit dans l'image, et son rendement peut être affecté dans des conditions de faible éclairage ou lorsque le bruit est particulièrement élevé.

Cependant, lorsqu'elle est utilisée dans des conditions appropriées et avec des prétraitements adéquats pour réduire le bruit, la technique du flux optique peut offrir des résultats précis et détaillés pour la détection d'objets en mouvement.

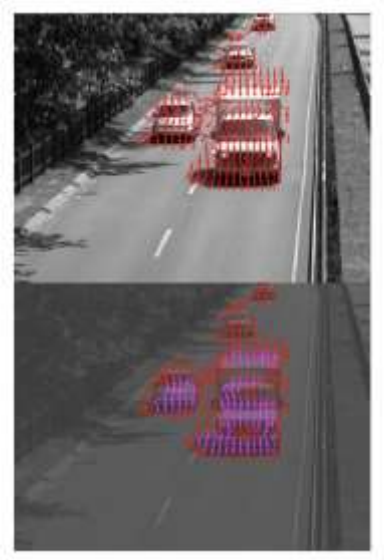


Figure 16 Détection de mouvements d'objets via la méthode du flux optique

3. Segmentation pour la Détection d'Objets

La segmentation d'images est une étape cruciale dans de nombreuses applications de vision par ordinateur, permettant d'isoler des régions d'intérêt ou des objets spécifiques au sein d'une image. La méthode appropriée dépend des caractéristiques de l'image et des objectifs de l'analyse.

3.1. Segmentation par seuillage

Une des méthodes les plus basiques, elle convertit une image en une version binaire basée sur un seuil de pixel défini. L'algorithme d'Otsu, par exemple, cherche à déterminer un seuil optimal en maximisant la variance entre deux classes de pixels. Il est particulièrement utile pour des images où les objets à détecter se démarquent nettement du fond, comme des objets sombres sur un fond clair. Pour des images plus complexes avec différents niveaux de luminosité, la variante "Otsu N thresholding" segmente l'image en plusieurs régions en utilisant plusieurs seuils [79].

La segmentation d'images est un domaine fascinant qui englobe une multitude de techniques. Bien que de nombreuses méthodes soient spécifiques aux images en niveaux de gris, elles peuvent souvent être adaptées aux images en couleur. L'une des approches les plus élémentaires est le seuillage basé sur l'histogramme de l'image. Cette technique identifie une valeur de seuil en observant les creux entre les pics de l'histogramme. Si l'histogramme est bimodal, le seuillage peut segmenter efficacement l'image. En explorant différents espaces colorimétriques issus des transformations RGB (Rouge, Vert, Bleu), on peut obtenir divers points de seuil pour une même image.

3.2. Segmentation par clustering

Cette technique consiste à regrouper les pixels ayant des caractéristiques similaires. Une approche courante est l'algorithme K-means, qui segmente les pixels basés sur des attributs tels que la couleur ou l'intensité. Imaginez une image avec plusieurs objets colorés distincts ; K-means pourrait identifier et segmenter chaque objet en regroupant les pixels en fonction de leur couleur [80].

Les techniques basées sur les régions, telles que la croissance, la division et la fusion de régions, analysent les valeurs des pixels à partir d'une graine initiale. Par exemple, la croissance de région débute avec un pixel et évalue les pixels adjacents en fonction de leur luminosité. La manière dont la graine est choisie et le processus de balayage sont essentiels à l'efficacité de cette méthode.

3.3. Segmentation basée sur les contours

La détection des contours se concentre sur l'identification des transitions nettes d'intensité ou de couleur dans une image, marquant souvent les limites des objets. Le détecteur de Canny, par exemple, est une méthode populaire qui utilise des dérivées pour repérer ces transitions. Dans

une situation où vous avez un objet bien défini, comme un objet rond sur un fond uniforme, la méthode Canny pourrait identifier précisément le bord circulaire de cet objet.

La détection des bords, quant à elle, se concentre sur les variations soudaines d'intensité dans l'image. Plusieurs opérateurs, tels que Sobel, Prewitt, Laplacien et Canny, ont été développés pour cette tâche, chacun ayant ses propres spécificités [81].

En explorant ces techniques, il est essentiel de souligner que chaque méthode possède ses avantages et ses contraintes. Leur performance est fortement liée au type d'image et au contexte d'application. Par exemple, le seuillage d'Otsu pourrait ne pas fonctionner aussi bien sur des images avec des éclairages variables, tandis que K-means pourrait avoir du mal avec des objets de couleurs similaires. La clé est de comprendre les spécificités de chaque méthode et de l'adapter à la situation donnée.

4. Extraction des Caractéristiques

L'extraction des caractéristiques joue un rôle crucial dans le traitement d'images, car elle permet de transformer les données brutes en un ensemble de caractéristiques significatives. Ces caractéristiques sont ensuite utilisées pour des tâches ultérieures telles que la classification ou la reconnaissance.

L'objectif principal de l'extraction des caractéristiques est de capturer les éléments essentiels de l'image tout en éliminant les informations redondantes. Cela facilite non seulement le traitement ultérieur, mais réduit également le coût computationnel.

4.1. Caractéristiques Basées sur la Couleur

Histogramme des Couleurs : Il donne une représentation globale de la distribution des couleurs dans une image.

Moments Statistiques de Couleur : Ils fournissent des informations sur la distribution des couleurs, comme la moyenne, la variance et le skewness.

Espaces Colorimétriques : Outre le RGB, d'autres espaces colorimétriques comme HSV, LAB et YCbCr peuvent être utilisés pour extraire des informations de couleur [82].

4.2. Caractéristiques Basées sur la Forme

Descripteurs de Fourier : Ils capturent les informations de forme en transformant le contour de l'objet en une série de Fourier.

Descripteurs de Hu : Ce sont des moments invariants qui fournissent une description de la forme qui est indépendante de la taille, de la position et de l'orientation [83].

4.3. Caractéristiques Basées sur la Texture

Matrice de Co-occurrence de Niveaux de Gris (GLCM) : Elle examine la distribution des niveaux de gris co-occurrents dans une image. Les caractéristiques telles que le contraste, la corrélation, l'énergie et l'homogénéité peuvent être extraites de la GLCM.

Transformée de Fourier : Elle permet d'analyser la texture en examinant la fréquence des motifs dans l'image.

Transformée en ondelettes : Elle décompose l'image en différentes échelles pour capturer les détails fins et grossiers de la texture [84].

4.4. Extraction des Caractéristiques à l'aide de GLCM

La Matrice de Co-occurrence de Niveaux de Gris (GLCM) est une méthode statistique qui examine la distribution des niveaux de gris co-occurrents dans une image. Pour chaque paire de pixels, un élément de la GLCM est incrémenté, ce qui permet de capturer la texture de l'image.

Contraste : Il mesure la variation locale dans la GLCM, indiquant la présence de motifs fréquents dans l'image.

Corrélation : Elle mesure la dépendance linéaire des niveaux de gris dans la GLCM.

Énergie : Elle fournit la somme des carrés des éléments de la GLCM, indiquant l'uniformité de la texture.

Homogénéité : Elle mesure la proximité des éléments de la GLCM à sa diagonale, indiquant la régularité de la texture.

L'extraction des caractéristiques est une étape essentielle qui influence directement la performance des algorithmes ultérieurs. En choisissant les bonnes méthodes et en combinant plusieurs types de caractéristiques, il est possible d'obtenir une représentation robuste et informative de l'image.

5. Classification des objets

La détection et la classification des objets en mouvement sont essentielles dans de nombreux domaines, notamment la surveillance vidéo, la reconnaissance d'objets et la navigation

autonome. Les objets en mouvement peuvent être de diverses natures, tels que les humains, les véhicules, les oiseaux, les nuages en déplacement, les arbres balançant au vent, etc. Pour distinguer ces objets les uns aux autres, il est nécessaire d'utiliser différentes caractéristiques et méthodes de classification. Voici quelques approches couramment utilisées pour la classification des objets :

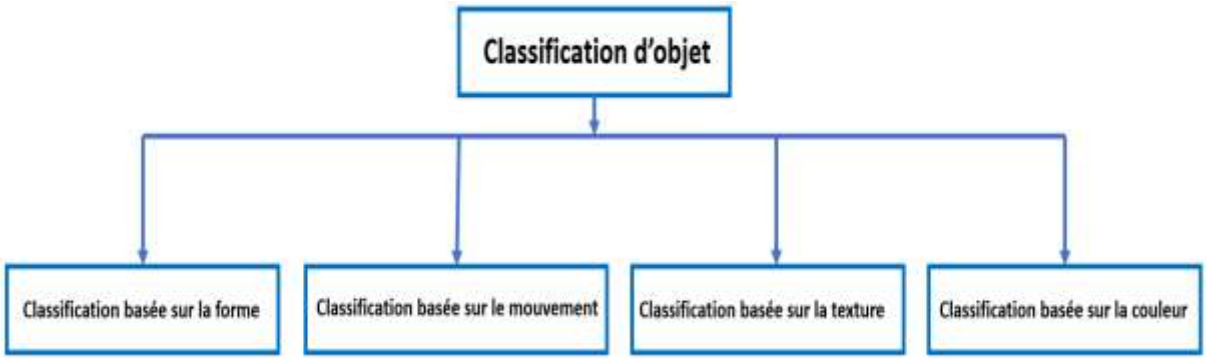


Figure 17 Techniques de classification des objets

Classification basée sur la forme : La forme d'un objet est l'une de ses caractéristiques les plus distinctives. Différentes descriptions de la forme, telles que les représentations par points ou par boîtes, sont utilisées pour classer les objets. Ces caractéristiques sont souvent combinées avec d'autres informations, telles que la couleur ou la texture, pour améliorer la précision de la classification.

Classification basée sur le mouvement : Certains objets, en particulier ceux qui ne sont pas rigides, présentent un mouvement périodique. Cette propriété peut être exploitée pour la classification. Par exemple, la démarche d'un humain ou le vol d'un oiseau ont des motifs de mouvement distinctifs qui peuvent être identifiés à l'aide de techniques telles que l'analyse du flux optique.

Classification basée sur la texture : La texture d'un objet fait référence à la répétition régulière de motifs ou de couleurs sur sa surface. Les techniques basées sur la texture analysent la distribution et l'orientation des motifs pour classer les objets.

Classification basée sur la couleur : La couleur est une caractéristique facilement identifiable et peut être utilisée pour distinguer rapidement différents objets, surtout lorsque la forme ou la texture ne sont pas distinctives. Des techniques telles que l'analyse d'histogrammes de couleur sont couramment utilisées pour cette approche.

5.1. Réseau Neuronal

Les réseaux neuronaux sont des algorithmes inspirés par la structure des neurones dans le cerveau humain. Ces systèmes apprennent à partir de grandes quantités de données et sont particulièrement performants dans la reconnaissance de modèles complexes.

- Fonctionnement :

Un réseau neuronal est composé de couches successives de neurones qui traitent des entrées et transmettent l'information à la couche suivante. La "formation" d'un réseau est réalisée par un processus itératif d'ajustement des poids associés à chaque connexion, généralement en utilisant une méthode appelée rétropropagation.

- Applications :

Bien qu'ils soient largement utilisés dans de nombreux domaines tels que la reconnaissance d'images ou la prédiction de séquences, les réseaux neuronaux sont également efficaces pour la classification, capable de gérer des frontières de décision non linéaires.

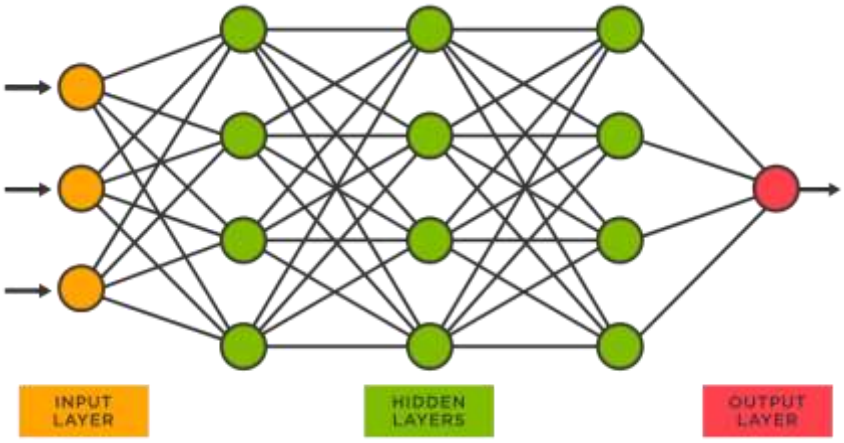


Figure 18 Architecture d'un réseau neuronal

5.2. Naïf Bayésien

Il s'agit d'un classificateur probabiliste basé sur le théorème de Bayes, avec une hypothèse de simplicité selon laquelle chaque paire de caractéristiques est indépendante [85].

Fonctionnement :

Le classificateur naïf bayésien estime la probabilité qu'une instance appartienne à une classe particulière en fonction de ses caractéristiques. Bien que cette indépendance supposée entre les caractéristiques ne soit pas toujours réaliste, le classificateur se révèle souvent étonnamment

efficace en pratique.

Applications :

Traditionnellement employé pour la classification de textes, comme le filtrage des spams, il est également efficace pour d'autres types de tâches de classification.

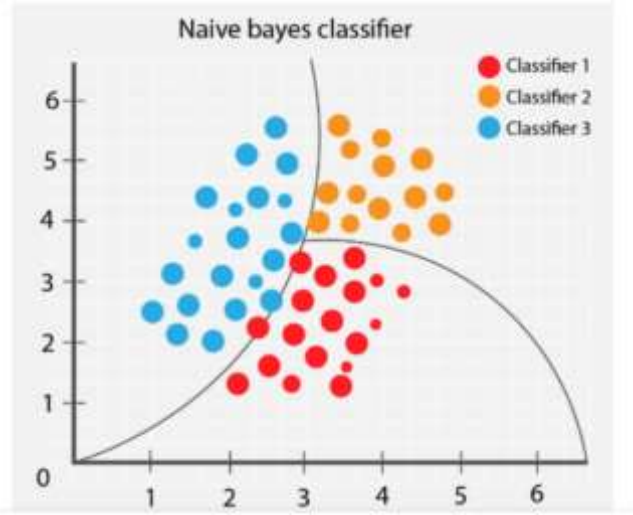


Figure 19 Naïf Bayésien

5.3. Machine à Vecteurs de Support (SVM)

Les SVM sont conçus pour trouver l'hyperplan qui sépare le mieux les données de deux classes en maximisant la marge entre elles [86].

Fonctionnement :

L'hyperplan est déterminé par un sous-ensemble des points de données, appelé vecteurs supports. La marge est la distance entre l'hyperplan et ces vecteurs supports. Les SVM peuvent aussi être étendus pour traiter des problèmes non linéaires en utilisant ce que l'on appelle le "truc du noyau", qui projette les données dans un espace de dimension supérieure où elles deviennent linéairement séparables.

Applications :

Les SVM sont largement reconnus pour leur robustesse dans les problèmes de classification binaire, notamment dans des situations où les données ne sont pas facilement séparables linéairement. Ils sont également employés dans des domaines tels que la reconnaissance d'images, la détection d'anomalies et la bio-informatique.

Chacune de ces méthodes possède ses avantages propres en fonction du type de données et de

la nature du problème à résoudre. Bien que les SVM soient souvent privilégiés pour leur capacité à gérer des problèmes non linéaires, les réseaux neuronaux sont idéaux pour des tâches nécessitant la reconnaissance de modèles complexes, et le naïf bayésien est remarquable pour sa simplicité et son efficacité, en particulier dans les problèmes où la dimensionnalité des données est élevée.

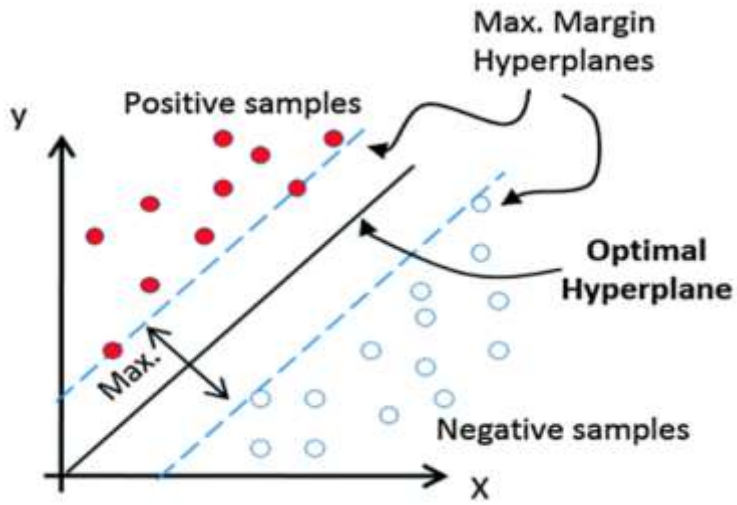


Figure 20 Principe des SVM

6. Approche Proposée pour la segmentation des objets « Triangulation de Delaunay »

Après avoir examiné diverses techniques de segmentation d'images, il est évident que chaque méthode possède ses propres mérites. Les techniques traditionnelles, comme le seuillage, le clustering, et en particulier l'algorithme K-means, ont été largement étudiées. L'algorithme K-means, introduit par Forgey et MacQueen, est une méthode de clustering qui vise à minimiser la distance moyenne au carré de chaque point de données à son centre le plus proche. Bien qu'efficace, cette méthode, comme d'autres techniques basées sur les limites et les régions, peut ne pas être idéale pour toutes les applications ou tous les types d'images.

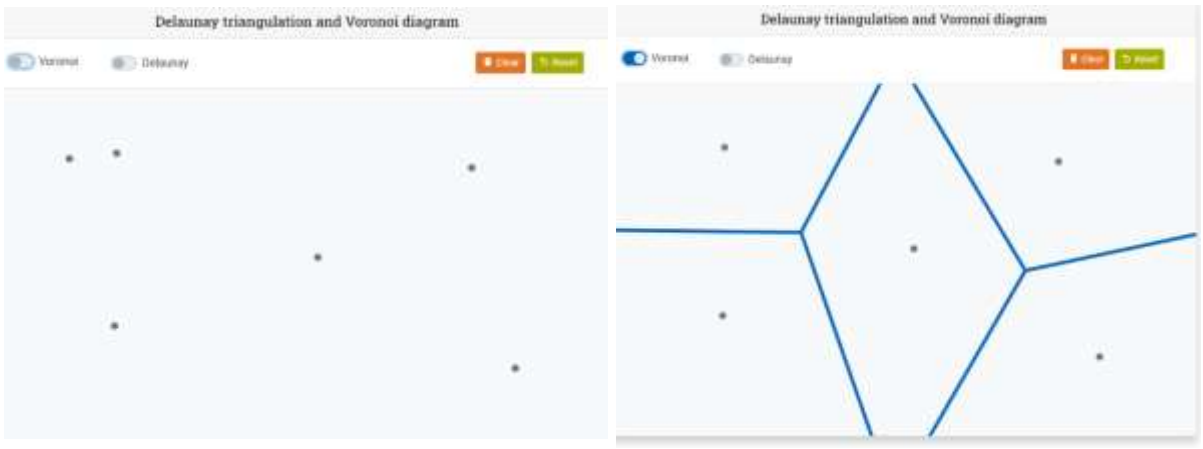
C'est ici que la DT se présente comme une alternative intéressante. Au lieu de se baser uniquement sur des critères de proximité ou de similarité comme le fait K-means, la DT utilise la géométrie computationnelle pour segmenter l'image. Chaque point d'intérêt de l'image est représenté comme un sommet d'un triangle, et ces triangles sont organisés de manière à refléter les relations géométriques entre les points.

La relation entre K-means et la DT peut être vue sous l'angle de la représentation des données.

Tandis que K-means regroupe les points en fonction de leur proximité à un centre, la DT établit des connexions entre les points en fonction de leur disposition géométrique. Les deux méthodes cherchent à segmenter l'image en régions cohérentes, mais la DT offre une perspective basée sur la structure géométrique, ce qui peut être avantageux dans certaines situations. Ainsi, en tenant compte des avantages distincts de la DT et de sa complémentarité avec des méthodes comme K-means, il est justifié de considérer cette technique pour des applications de segmentation d'images avancées.

Dans notre approche, nous utilisons le Diagramme de Voronoi (VD), une structure géométrique polyvalente, pour la segmentation d'images. Le VD est basé sur l'extraction de points caractéristiques qui définissent ensuite les centres des cellules de Voronoi.

Pour définir le VD : Considérons un ensemble de points 2D (voir figure 21(a)). La région de Voronoi pour un point P_i est l'ensemble de tous les points plus proches de P_i que de tout autre point. Formellement, soit $S = \{P_1, P_2, \dots P_n\}$ un sous-ensemble fini de R^m et d'une métrique sur R^m. La région de Voronoi $VR(P_i)$ pour un point P_i est définie par $VR(P_i) = \{P \text{ dans } R^m \mid d(P, P_i) \leq d(P, P_j)$ pour tout $j\}$, c'est-à-dire que $VR(P_i)$ contient tous les points au moins aussi proches de P_i que de tout autre point de S. L'ensemble de toutes ces régions forme le VD (S) (voir figure 21(b)) [87].



(a) (b)

Figure 21 Construction du diagramme de Voronoi à partir de points 2D : Avant (a) et Après (b)

Une structure étroitement liée au VD est la DT. La DT est en réalité le dual du VD. Dans la DT, deux points sont connectés par un segment si et seulement si leurs cellules de Voronoi partagent une arête. Cette relation entre le VD et la DT est fondamentale pour de nombreuses applications,

y compris la nôtre, où nous utilisons ces structures pour la segmentation d'images.

La dualité entre le VD et la DT est telle que les sommets du VD correspondent aux faces de la DT, tandis que les cellules de Voronoi correspondent aux sommets de la DT. Une fois le VD construit, il est facile de créer la DT en reliant deux points dont les polygones de Voronoi partagent un bord. En particulier, trois points distincts forment un triangle de Delaunay s'il n'y a pas d'autre point à l'intérieur du cercle circonscrit à ce triangle. Cette idée est illustrée dans la Figure 22.

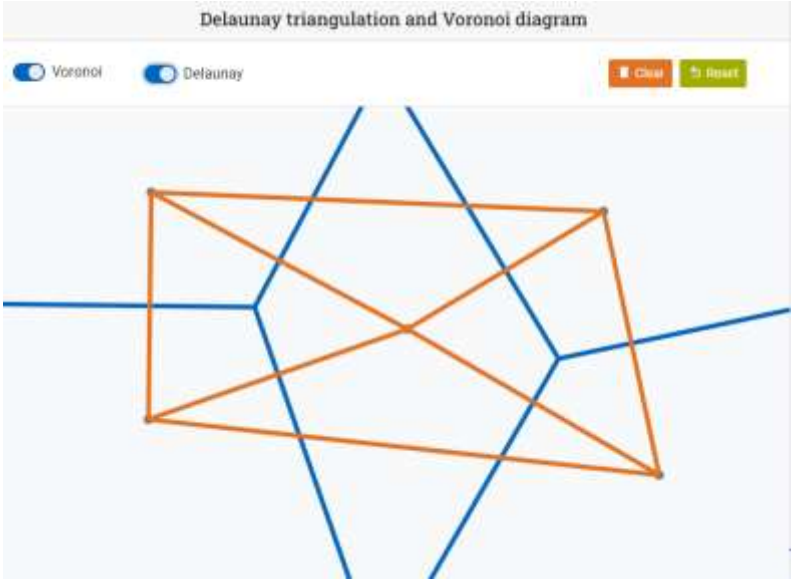


Figure 22 Triangulation de Delaunay (bleu) de Cinque motifs de points et leur VD (lignes orange).

Au lieu d'utiliser le VD directement sur toute l'image, nous choisissons certains "points clés" (<=255) de l'image pour construire le VD. Ces points clés sont choisis en fonction des variations d'intensité de couleur dans l'image. En d'autres termes, nous regardons où les couleurs changent le plus et utilisons ces points pour notre VD.

Une fois que nous avons ces points clés, nous créons un histogramme, qui est essentiellement un graphique montrant la fréquence de chaque intensité de couleur dans l'image. Avec cet histogramme, nous utilisons le VD et la DT pour repérer les principales intensités de couleur (les pics) et les moins présentes (les minima).

Après avoir identifié ces points importants, nous les organisons dans un certain ordre pour nous aider à décider comment segmenter l'image. Les points qui sont sur les bords extérieurs de la DT sont particulièrement utiles car ils nous montrent les couleurs les plus dominantes de l'image.

En résumé, au lieu d'analyser chaque pixel de l'image, nous sélectionnons et utilisons des points clés pour nous aider à comprendre et segmenter l'image de manière plus efficace.

La méthode proposée présente des similitudes avec l'approche de seuillage dynamique pour la segmentation. Cependant, elle se distingue par sa manière de prendre des décisions pour la fusion et la division.

La distribution des couleurs dans les scènes routières se distingue nettement de l'arrière-plan, ce qui en fait un élément essentiel de notre analyse. Nous utilisons le VD pour élaborer les DT à partir de points issus de l'histogramme de l'image. La frontière externe du DT correspond principalement à l'Enveloppe Convexe (CV) des points distinctifs. De cela, les deux maximas principaux sont identifiés en sélectionnant les deux valeurs les plus élevées de la liste des sommets du DT, qui correspondent aux pics de l'histogramme. Pour localiser les minima situés entre ces pics, une série d'étapes est mise en œuvre :

- Produire l'histogramme de l'image principale.
- Formuler le VD/DT pour extraire la liste des sommets et identifier les deux pics.
- Annuler les points en dessous du pic initial et ceux dépassant le pic secondaire.
- Attribuer une valeur équivalente au maximum des deux pics (pic1, pic2) aux points qui sont à zéro.
- Calculer de nouveaux points à travers la formule :

$$Valnouveau(x) = \left|Val(x) - max\left(Val(x)\right)\right| \tag{2.8}$$

Cette méthodologie induit une inversion localisée dans l'histogramme de l'image. En conséquence, les points minima sont intégrés dans l'enveloppe convexe formée par le DT.

Par la suite, ces points caractéristiques distincts sont organisés séquentiellement en fonction de leurs valeurs de luminosité. Cette séquence forme un vecteur (V) qui englobe des valeurs déterminant les critères à la fois pour les décisions de fusion et de division. Cette stratégie est ensuite exécutée.

En résumé, notre système est capable de gérer la segmentation d'images de différentes complexités. Contrairement à d'autres techniques comme le clustering K-Means, notre méthode est entièrement automatisée et ne nécessite aucun paramétrage. Elle se base uniquement sur l'histogramme de l'image, ce qui est un avantage considérable. Bien que la segmentation par

découpe d'histogramme soit simple et produise une image binaire, la méthode VD offre des résultats plus détaillés, même sur des images avec des arrière-plans complexes ou un éclairage inégal. L'objectif est de représenter l'image avec le moins de couleurs possible tout en conservant les détails essentiels.



Figure 23 Résultat de la triangulation de Delaunay

7. Étiquetage des Composants Connectés

L'étiquetage des composants connectés, consiste à attribuer à chaque pixel d'un composant connecté une étiquette unique. Cette démarche est essentielle pour distinguer les différents objets présents dans une image binaire. De plus, elle sert de prérequis pour l'analyse d'image et l'extraction de caractéristiques.

En ce qui concerne l'algorithme DT évoqué précédemment, il a pour fonction de réduire le nombre de valeurs de gris dans l'image originale. Cet algorithme renvoie les groupes de chaque frame. Par conséquent, l'étiquetage des composants connectés nécessite une image binaire pour fonctionner. Ainsi, pour chaque valeur de gris (ou groupe), nous avons effectué une binarisation afin de transformer l'image en format binaire. Ceci permet ensuite d'étiqueter et d'extraire chaque région de l'image.

Ci-dessous, nous présentons l'algorithme d'étiquetage proposé pour les groupes obtenus à partir de DT :

- Lire l’image obtenue par Delaunay
- Binariser l’image
- Appliquer les opérations morphologiques

- Analyser les Composants Connectés
- For i in range (1, TotalLabels) avec TotalLabels=nombre des objets dans l'image

- Un masque est créé pour ce composant.

- Les coordonnées de la boîte englobante de ce composant sont extraites.

- Une boîte englobante est dessinée autour de ce composant sur l'image originale.

- Extraction de l'objet d'intérêt.

Cette méthode garantit que chaque objet ou région distincte dans l'image est correctement identifié et étiqueté (voir figure 24), permettant ainsi une analyse plus approfondie et une classification ultérieure.

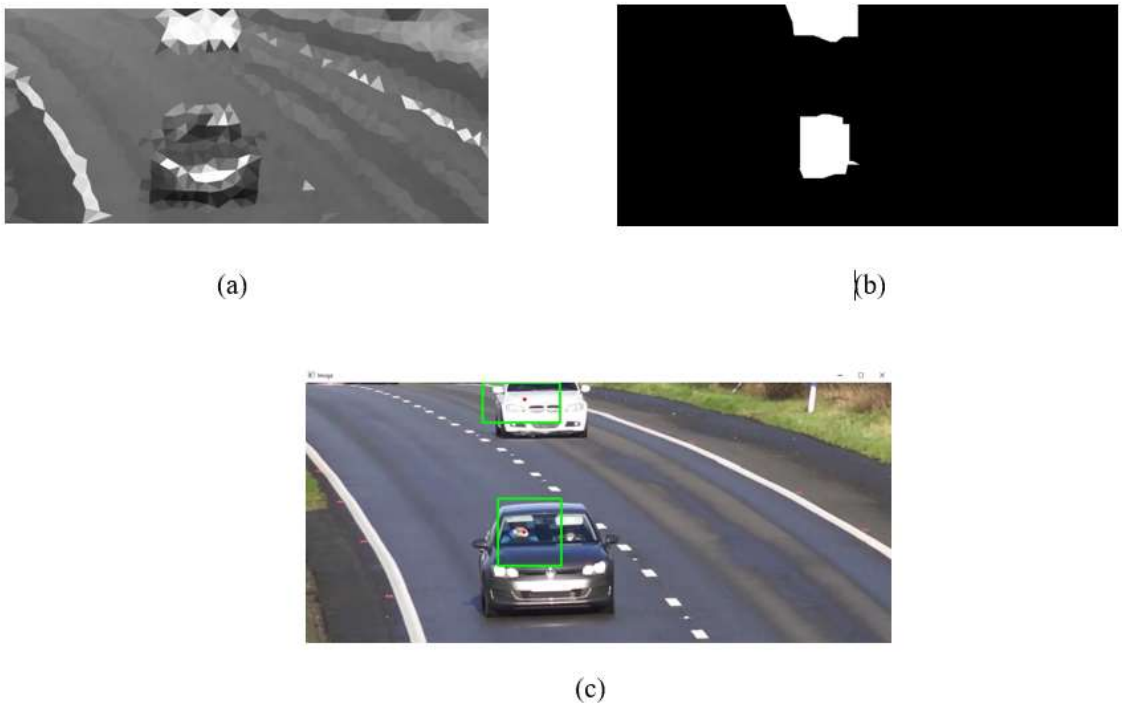


Figure 24 Résultats de test de la détection des objets. (a) DT segmentation, (b) Etiquetage des objets connecté(c) Détection des objets dans l'image originale**.**

8. Phase d'extraction des caractéristiques

Un objet peut être défini par un ensemble de fonctions de plusieurs mesures, nommées caractéristiques. Dans le domaine du traitement d'images, ces données volumineuses sont transformées par un processus, appelé extraction de caractéristiques, en vecteurs contenant des informations pertinentes comme la texture, la couleur et la forme. Ces vecteurs servent ensuite

d'entrée pour la méthode de classification.

L'objet est souvent réduit à sa forme géométrique, ce qui peut entraîner des omissions ou inclure une partie du fond. Pour résoudre ce problème, on se base uniquement sur les caractéristiques de couleur et de texture pour représenter l'objet. Les caractéristiques de texture aident à identifier les régions d'intérêt et fournissent des informations sur la répartition de l'intensité dans une image. Nous utilisons l'approche statistique GLCM pour extraire ces caractéristiques. Les éléments extraits avec GLCM comprennent l'énergie, le contraste, l'homogénéité et la corrélation.

Algorithme GLCM :

- Charger l'image RGB
- La convertir en niveau de gris
- Créer la GLCM à partir de cette image
- Calculer certaines propriétés statistiques de la GLCM : [Contraste, homogénéité, énergie, corrélation].

La couleur est un descripteur d'image puissant car elle est invariante aux occlusions partielles ou transformations géométriques. Cependant, elle est sensible aux conditions d'éclairage. Les caractéristiques de couleur sont obtenues à partir de l'histogramme HSV.

Algorithme HSV :

- Charger l'image RGB
- La convertir en HSV
- Extraire la teinte et calculer sa moyenne
- Extraire la saturation et sa moyenne
- Enfin, extraire la valeur et sa moyenne.

Pour chaque région, nous fusionnons les vecteurs de caractéristiques GLCM et HSV, V = [couleur, texture], qui sont ensuite utilisés pour la phase de classification.

Pour notre base de données, nous avons utilisé nos propres vidéos, converties en images. Chaque image est traitée avec notre algorithme, et les vecteurs de caractéristiques sont extraits pour chaque région. Ensuite, un étiquetage manuel est effectué pour déterminer si la région est

en mouvement ou statique.

9. Approche proposée pour la classification des Objets

Selon une étude réalisée en 2006 par Kim [88], une expérimentation a été menée pour comparer la performance de plusieurs classificateurs, notamment le réseau neuronal, le naïf bayésien et le SVM. Pour évaluer la précision de chaque classificateur, une technique de validation croisée en cinq plis a été employée et répétée à dix reprises.

La Figure 25 illustre les résultats obtenus. Ces courbes, connues sous le nom de courbes ROC, montrent le taux de détection en fonction du taux de fausses alertes pour chaque classificateur. Plus la courbe s'approche du coin supérieur gauche du graphique, plus le classificateur est performant. À la lumière de cette figure, il est évident que le SVM surpasse les deux autres techniques en termes de performance, atteignant les taux de détection les plus élevés pour les plus faibles taux de fausses alertes.

Le classificateur SVM a donc été identifié comme étant le plus performant pour cette tâche. Le SVM cherche à séparer les données en classes en identifiant l'hyperplan qui maximise la distance, ou "marge", entre elles. Hsu et al. (2003) [89] ont également mis en évidence la puissance du SVM dans ce contexte. Pour la détection d'objets en mouvement, les exemples positifs proviennent d'images d'objets en déplacement, tandis que les exemples négatifs englobent les éléments statiques de l'image. À partir d'exemples d'entraînement générés manuellement, le SVM est utilisé pour identifier l'hyperplan optimal pour séparer ces deux classes.

En somme, cette étude, corroborée par les résultats de la Figure 25, justifie pleinement le choix du SVM comme méthode de classification pour cette tâche spécifique.

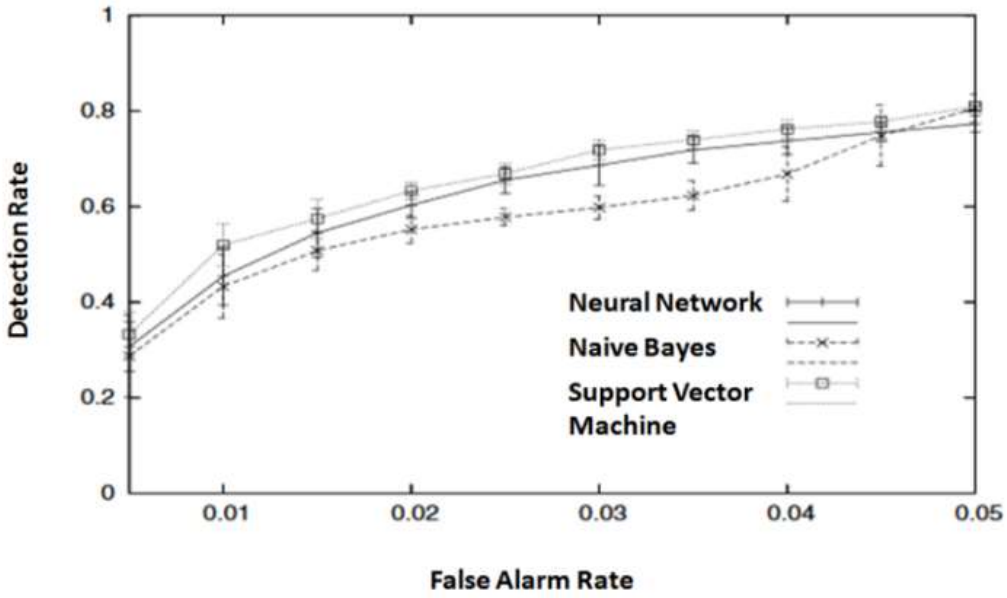


Figure 25 Comparaison des performances de classification entre le réseau neuronal, le classifieur bayésien naïf et la SVM

10. Résultats Expérimentaux et Évaluation

Dans cette recherche, nous avons exploité des vidéos capturées à l'aide de notre propre dispositif de caméra pour des analyses expérimentales. Les aspects de calcul et de programmation de notre travail ont été menés sur MATLAB R2013a. Une étape cruciale de notre évaluation a consisté à comparer la performance de notre système, en termes de reconnaissance d'objets, avec des algorithmes référencés dans la littérature actuelle.

10.1. Présentation des jeux de données utilisés pour évaluer la performance de l'approche

D'un point de vue empirique, notre ambition était de concevoir un classificateur capable de distinguer les objets mobiles des objets statiques. Pour y parvenir, nous avons élaboré un jeu de données varié, extrait de nombreuses vidéos d'une caméra de surveillance routière. Cela a nécessité une segmentation des images vidéo pour isoler les régions d'intérêt, essentielles pour l'extraction ultérieure des caractéristiques destinées à notre classificateur SVM.

Notre jeu de données propriétaire englobe une multitude d'entités, notamment des piétons, des véhicules tels que des voitures et des camions, des vélos et des éléments statiques comme des arbres et des panneaux de signalisation. La sélection du jeu de données a été méticuleusement effectuée, garantissant des angles de caméra diversifiés (frontal, arrière et latéral) et des scénarios d'éclairage variés. L'annotation des données, une étape essentielle, a été effectuée

manuellement.

10.2. Entraînement du Classificateur

Nous avons formé notre jeu de données à l'aide du classificateur SVM linéaire. Nous avons institué une division où 20% du jeu de données étaient réservés pour les tests, le reste étant destiné à la formation. Cette division a été effectuée de manière aléatoire. L'issue du processus de formation était la création d'un classificateur, optimisé pour les phases d'évaluation ultérieures.

Pour assurer l'efficacité de notre algorithme, nous avons adopté une approche rigoureuse dans la formation de notre jeu de données en utilisant le classificateur SVM linéaire. Ce choix s'est basé sur les propriétés intrinsèques du SVM, particulièrement adapté à la séparation linéaire des données, offrant ainsi une capacité robuste de discrimination entre les classes.

Afin de garantir une évaluation précise et fiable de la performance du classificateur, nous avons décidé d'adopter une stratégie de division du jeu de données. 20% de celui-ci a été méticuleusement séparé pour servir de jeu de données test, garantissant ainsi que le classificateur serait évalué sur des données qu'il n'avait jamais vues auparavant. Les 80% restants ont été utilisés comme ensemble d'apprentissage pour permettre au modèle de comprendre et de reconnaître les nuances et les motifs intrinsèques des données.

L'assignation aléatoire était cruciale pour prévenir toute prédisposition ou ajustement excessif du classificateur à un segment spécifique des données. Ce processus veillait à ce que le modèle acquière une perspective équilibrée et équitable de toutes les données disponibles.

Le processus de formation a abouti à un classificateur SVM sophistiqué qui va au-delà de la mémorisation des données d'entraînement. Il a acquis la capacité de reconnaître des schémas complexes et de faire des prédictions précises sur des données nouvelles, témoignant de sa capacité à généraliser et à maintenir des performances fiables face à de nouveaux défis.

10.3. Évaluation du Système

Afin d'apprécier la performance de notre algorithme proposé dans la détection des objets en mouvement, nous avons choisi d'utiliser deux vidéos capturées sous des angles distincts : frontal et latéral, et ce, à différents moments. Ces vidéos servent de terrain d'essai pour évaluer la finesse et la précision de notre méthode face à des scénarios diversifiés.

Afin d'ancrer notre évaluation dans un contexte comparatif, nous avons juxtaposé notre

algorithme à plusieurs autres méthodes prévalentes dans la littérature, notamment la méthode de différence de frame, la soustraction de l'arrière-plan, ainsi que l'algorithme plus rapide, le RCNN. Chacun de ces algorithmes a été testé et formé à l'aide de nos données, spécifiquement conçues pour détecter une unique catégorie d'objets : les 'objets en mouvement'. Notre échantillon test se composait d'une vidéo de 100 frames, contenant 215 objets en mouvement.

Confrontés à ces multiples essais, les résultats se sont avérés révélateurs. La méthode de différence de frame a permis de reconnaître 185 objets en mouvement, soit un taux de 86,04%. La technique de soustraction de l'arrière-plan a identifié 192 objets, équivalent à un taux de 89,30%. Le RCNN plus rapide a montré une légère amélioration avec la détection de 197 objets, soit un taux de 91,62%. Cependant, notre méthode proposée a surpassé toutes les autres avec la reconnaissance de 203 objets, se traduisant par un taux impressionnant de 94,42%.

Le tableau ci-dessous résume les résultats obtenus :

Méthode	Taux de Reconnaissance
Différence de frame	**86,04%**
Soustraction d'Arrière-plan	**89,30%**
Faster-RCNN	**91,62%**
Notre Méthode Proposée	**94,42%**

Tableau 4 Comparaison des taux de reconnaissance selon différentes méthodes

Ces chiffres parlent d'eux-mêmes. Ils attestent clairement que l'algorithme que nous avons développé se distingue non seulement par sa précision, mais qu'il surpasse également d'autres méthodes classiques en termes de taux de reconnaissance. Cette performance illustre notre engagement à repousser les limites de la technologie et à offrir une solution de détection des plus performantes.

Notre métrique d'évaluation est conçue pour évaluer l'efficacité de notre algorithme dans la détection d'objets mobiles. Pour une évaluation complète, nous avons utilisé des vidéos capturées sous différents angles, à savoir frontal et latéral, et à différents moments de la journée.

La Figure 26 offre une représentation visuelle, montrant un grand nombre d'images issues de différentes vidéos, capturées sous différents angles et à différents moments.

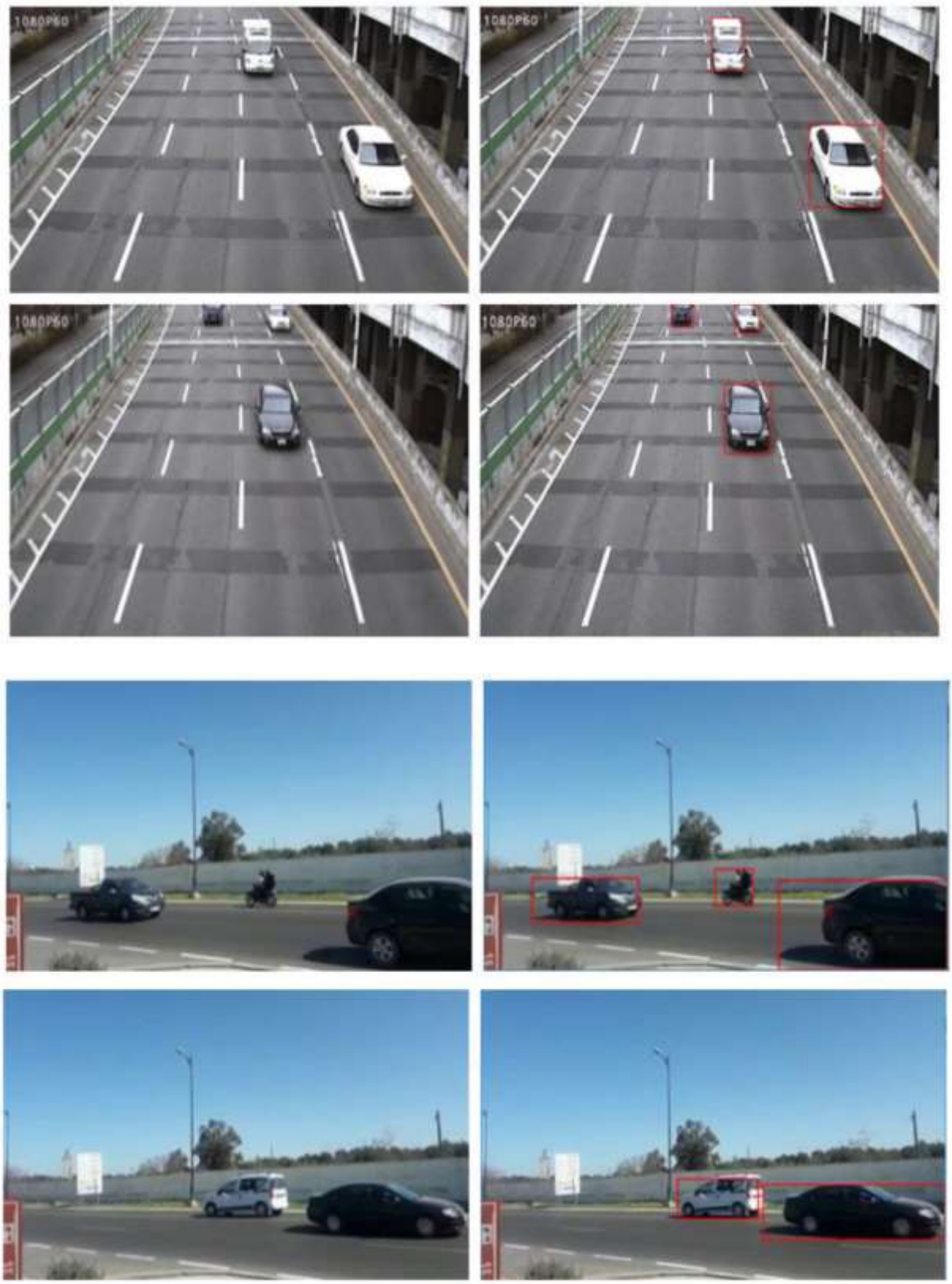




Figure 26 Résultats de détection de notre algorithme sur deux vidéos prises à différents moments

Les résultats empiriques démontrent sans équivoque la supériorité de précision de notre algorithme proposé, soulignant son taux de reconnaissance plus élevé par rapport à d'autres méthodes classiques.

11. Conclusion

Le chapitre actuel présente et discute notre approche proposée pour la détection des objets en mouvement. Nous avons commencé par explorer la thématique de la détection des objets en mouvement dans les séquences vidéo, un pilier de la vision par ordinateur avec des implications dans la sécurité et le trafic urbains.

Nous avons proposé une approche efficace qui utilise une segmentation d'image avancée et une classification SVM, offrant une amélioration significative par rapport aux méthodes traditionnelles. Les résultats que nous avons obtenus confirment la robustesse et l'efficacité de notre méthode, tant dans la détection que dans la localisation des objets en mouvement, surpassant les algorithmes existants. Après avoir établi les fondements de la détection d'objets en mouvement, nous progressons vers la mise en place d'un système embarqué pour le comptage de véhicules en temps réel. Cette étape représente une avancée clé dans nos

recherches, appliquant nos trouvailles à des situations réelles de trafic urbain et enrichissant notre compréhension des dynamiques du trafic.

Chapitre 3

Implémentation d'un Réseau Neuronal sur une carte embarquée

Chapitre 3 : Implémentation d'un Réseau Neuronal sur une carte embarquée

1. Introduction

Au cœur de notre recherche visant à atténuer les problèmes de congestion routière à Bouskoura, une ville en expansion près de Casablanca, nous avons mis l'accent sur le déploiement de technologies de pointe axées sur le matériel pour le comptage des véhicules. Notre approche s'est appuyée sur l'utilisation des cartes électroniques avancées, notamment l'Allwinner A133 pour la collecte de données et le Nvidia Jetson Nano pour le déploiement de solutions de détection et de comptage de véhicules. Ces choix de matériel ont été déterminants pour la mise en œuvre efficace des algorithmes de Deep Learning.

En collaboration avec les autorités locales, nous avons utilisé une caméra intelligente intégrant YOLOv7-tiny pour la détection et la classification des véhicules, et l'algorithme Deep SORT pour un suivi précis. Le Jetson Nano, avec sa capacité à traiter en temps réel les flux vidéo, s'est révélé essentiel pour l'exécution en temps réel de ces algorithmes complexes. De même, l'Allwinner A133 a joué un rôle crucial dans la collecte efficace des données nécessaires à l'apprentissage et à l'amélioration continue du modèle.

La constitution d'un jeu de données conséquent et bien étiqueté a été un autre défi de notre projet, permettant d'assurer l'apprentissage et la performance optimale du modèle. Cette solution intégrée, alliant matériel et algorithmes avancés, a prouvé son efficacité et sa précision dans la gestion du trafic à Bouskoura, marquant ainsi une avancée significative dans l'amélioration des infrastructures de mobilité urbaine.

2. Matériel et méthodes

2.1. Site d'étude

À la demande de la commune, nous avons développé une solution visant à analyser précisément le flux de trafic au rond-point situé aux coordonnées 33.48448, -7.63563, un point névralgique entre Bouskoura et Casablanca. Cette localisation a été choisie stratégiquement en raison de son volume de trafic élevé, particulièrement aux heures de pointe. En effet, ce rond-point est un axe majeur pour les résidents de Bouskoura travaillant à Casablanca, entraînant ainsi une congestion régulière et significative. L'objectif de notre étude est de fournir à la commune des données

détaillées sur les voies les plus denses, permettant ainsi d'envisager des améliorations d'infrastructure ciblées pour optimiser la fluidité du trafic.

Pour ce faire, nous avons opté pour l'utilisation d'une carte Jetson Nano, un choix dicté par plusieurs avantages clés. Cette carte embarquée offre une autonomie remarquable, essentielle pour une analyse continue et fiable du trafic sans dépendance excessive vis-à-vis des ressources centrales. De plus, sa capacité à exécuter des algorithmes avancés de Deep Learning directement sur le périphérique ('Edge computing') nous permet de traiter les données en temps réel, un atout considérable pour la réactivité et l'efficacité du système. Cette approche locale du traitement des données réduit également la charge sur la bande passante du réseau, tout en garantissant un haut niveau de confidentialité et de sécurité des données, conformément aux exigences de la commune. Ainsi, la carte Jetson Nano se présente comme le meilleur choix pour une solution de gestion du trafic à la fois efficace, autonome et respectueuse des enjeux de confidentialité.



Figure 27 Localisation du site d'étude sur Maps

2.2. Matériel

2.2.1. Le nano-ordinateur « Jatson Nano »

Le Jetson Nano de NVIDIA (figure 28), mis en lumière dans cet essai, est un nano-ordinateur sélectionné pour sa capacité à réaliser de l'inférence en temps réel sur site. Cette particularité permet d'éviter le transfert et le traitement différé des données à distance. Au cœur du Jetson

Nano se trouvent les GPU, initialement conçus pour le traitement graphique et d'images, qui brillent par leur architecture hautement parallèle et leurs multiples cœurs, utiles pour des calculs simultanés. NVIDIA a valorisé ces GPU au-delà de leur usage traditionnel, notamment avec la plateforme CUDA pour le traitement parallèle. Le Jetson Nano, fonctionnant sous un Ubuntu complet, se compare à un ordinateur sur une carte unique comme le Raspberry Pi, doté d'un GPU Maxwell de 128 cœurs, d'un CPU ARM A57 à 1.43 GHz, de 4 Go de RAM et utilisant une carte MicroSD pour le stockage, le rendant ainsi idéal pour des applications en apprentissage machine et d'autres domaines innovants [90].

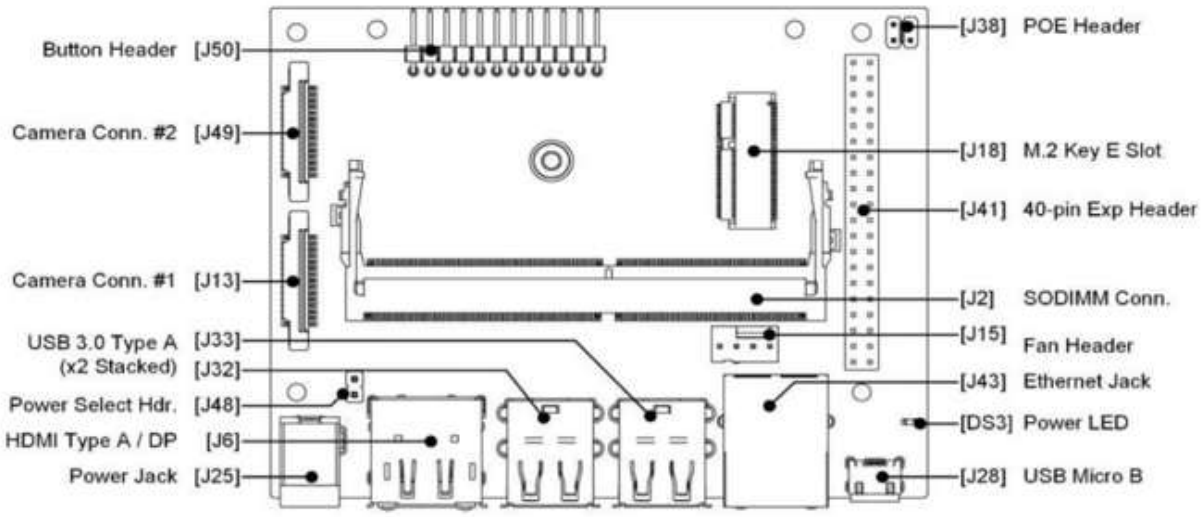


Figure 28 Jetson Nano pour l'implémentation GPU

La figure 29 présente un schéma détaillé de la carte porteuse du Jetson Nano, un produit de NVIDIA. Au centre, on trouve le SoC (System on Chip) qui constitue le cœur du Jetson Nano. Autour du SoC, diverses interfaces et connecteurs sont représentés, y compris :

- Des ports USB, avec des spécifications différentes comme USB 2.0 Micro-B, USB 3.0 et USB SS (SuperSpeed) Type A qui permettent la connexion de divers périphériques.
- Deux connecteurs pour caméra, qui permettent de connecter des caméras pour des applications de traitement d'image.
- Une sortie HDMI de type A et un DisplayPort pour la connexion à des écrans.
- Un connecteur pour carte SD, servant de stockage pour le système.
- Des broches GPIO (General Purpose Input/Output) pour les connexions d'entrée

et de sortie générales, utiles pour les projets IoT.

- Des connecteurs d'extension et des niveaux de décalage (Level Shifters) qui permettent d'ajuster les niveaux de tension pour différents modules.
- Un port Ethernet RJ45 pour la connectivité réseau.
- Une prise d'alimentation CC pour l'alimentation électrique de la carte.
- Des options de connectivité sans fil avec des emplacements pour des modules Wi-Fi et Bluetooth (M.2, Key E Socket).
- Un connecteur de ventilateur, indiquant la possibilité de refroidir activement le système pour maintenir les performances thermiques.

Le schéma illustre la flexibilité et les nombreuses capacités d'interface du Jetson Nano, le rendant adapté pour les applications de calcul embarqué et de deep learning, notamment dans les domaines de la robotique et de la vision par ordinateur.

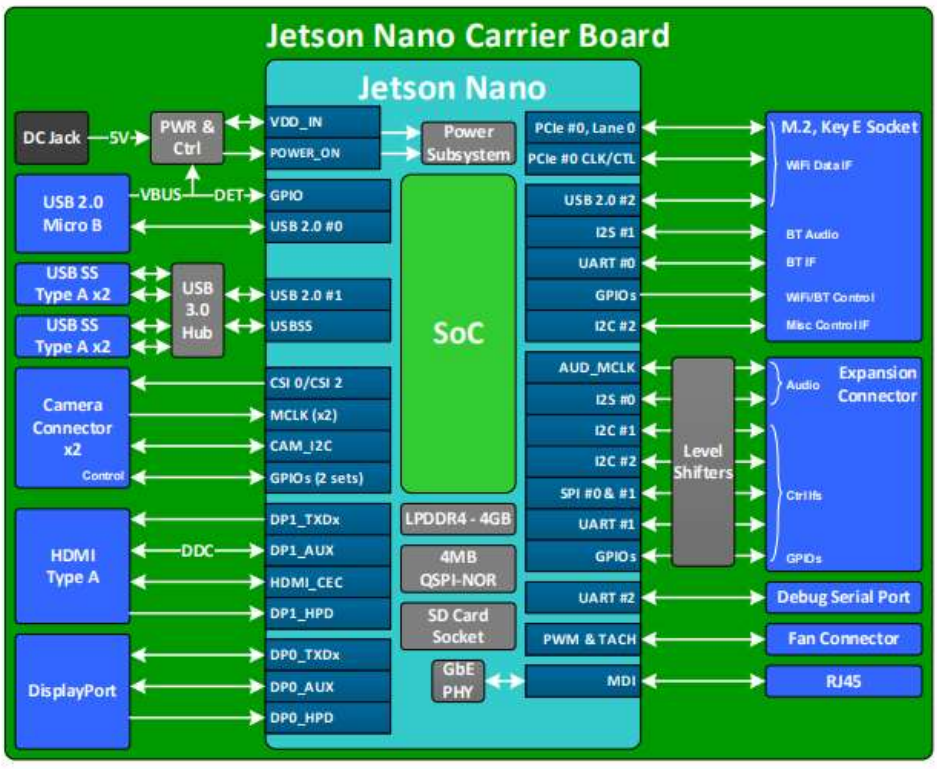


Figure 29 Schéma de Bloc du Jetson Nano

2.2.2. Le nano-ordinateur « Allwinner A133 »

Le Allwinner A133, un processeur intermédiaire. Fabriqué via un processus technologique de 12 nm, ce SoC se compose de quatre cœurs Cortex-A53 organisés en un seul cluster, le plus rapide d'entre eux atteignant une fréquence de 1,6 GHz et gérant les instructions 64 bits. La gestion des tâches graphiques est confiée au GPU PowerVR GE8300.

Avec un CPU qui peut atteindre une fréquence de 1.5 GHz, le A133 surpasse le CPU A35 en termes de performance, ce qui assure une réactivité optimale pour l'utilisation quotidienne des applications. Il est également conçu autour d'une architecture à basse consommation CoolFlex et intègre une gestion de consommation d'énergie dynamique, ce qui lui permet de maintenir une efficacité énergétique dans la plupart des scénarios d'utilisation. De plus, le GPU PowerVR GE8300 intégré répond aisément aux besoins de calcul graphique des applications contemporaines et supporte les dernières API de graphiques telles que OpenGL ES 3.2, Vulkan 1.1 et OpenCL 1.2, en conformité avec les dernières normes de certification Android GMS.

Pour ce qui est de la capacité opérationnelle, le A133 est capable de décoder des vidéos de formats répandus tels que H.265, VP9 et H.264, avec une performance maximale pouvant atteindre 4K à 30 ips. Il est également capable d'encoder des vidéos H.264 jusqu'à 1080p à 60 ips et de gérer des sorties d'affichage allant jusqu'à FHD 1920 x 1200 à 60 ips. La technologie d'amélioration d'affichage SmartColor 2.0 d'Allwinner garantit une expérience visuelle supérieure. En matière audio, le processeur prend en charge des interfaces analogiques comprenant 2 ADC et 2 DAC, ainsi que des écouteurs stéréo, tandis que les interfaces numériques incluent I2S, DMIC, OWA, répondant ainsi aux exigences des solutions de reconnaissance audio actuelles [91].

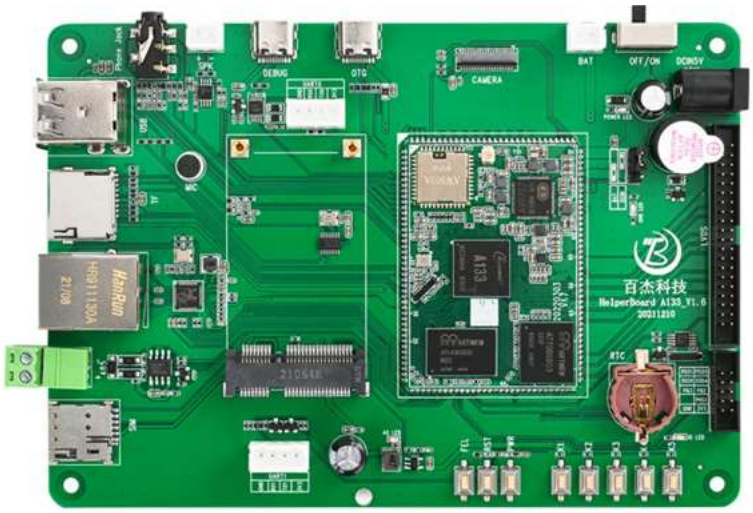


Figure 30 Allwinner A133

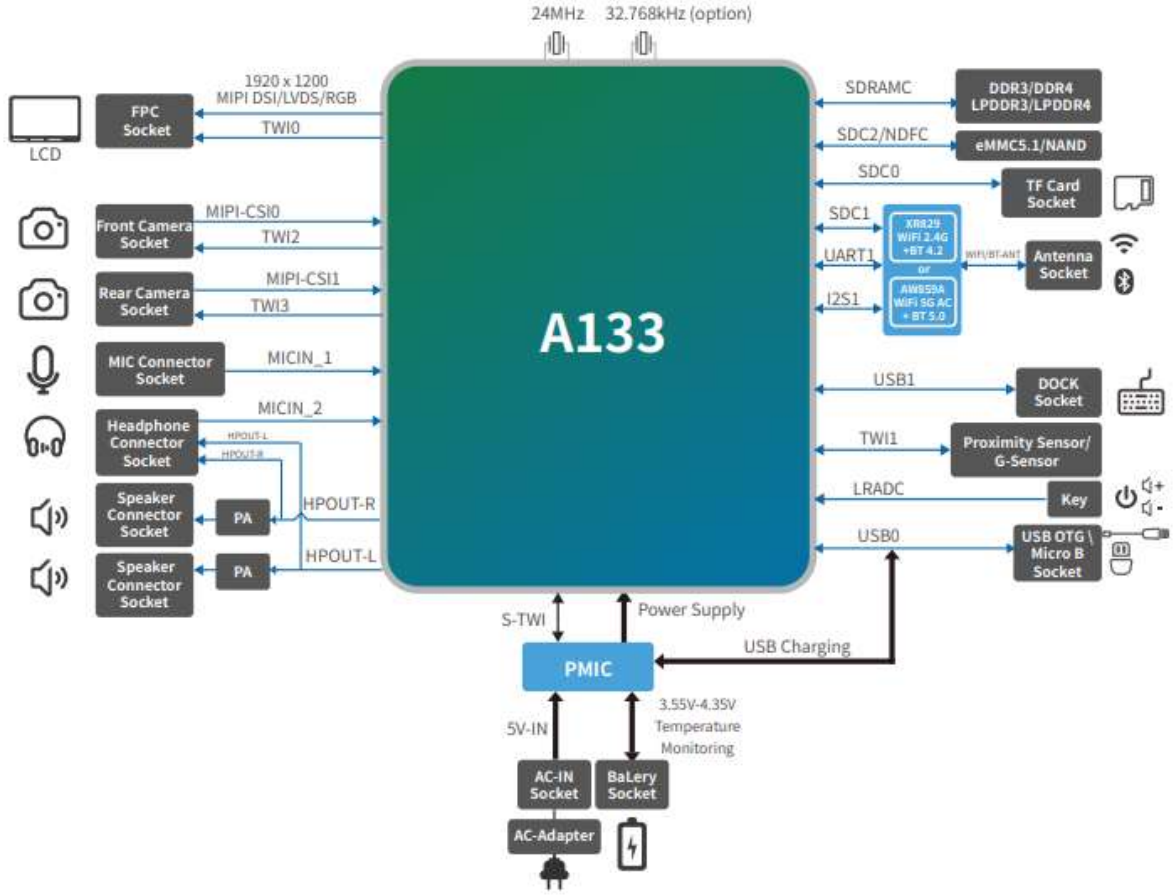


Figure 31 Diagramme d'application de la carte Allwinner A133

4.1.1.1. Spécifications du Allwinner A133

Composante	Spécification
CPU	Quad-core ARM Cortex™-A53@1.5GHz, 32 KB L1 I-cache + 32 KB L1 D-cache per core, 512 KB L2 cache, Low-power CoolFlex™ power management architecture
GPU	IMG PowerVR GE8300, Supports OpenGL ES3.2, Vulkan1.1, OpenCL1.2
Memory	DDR3/DDR3L/DDR4/LPDDR3/LPDDR4, 32-bit width, maximum memory capacity up to 4 GB, eMMC 5.1, 8-bit parallel NAND Flash, SPI Nand flash
Video	H.265 video decoder 4K@30fps, H.264 video decoder 4K@30fps, VP9 video decoder 720p@30fps, H.264 video encoder 1080p@60fps, MJPEG/JPEG Baseline encoder 4K@15fps
Display	Interface MIPI-DSI, interface LVDS double lien, interface RGB, La résolution maximale pour l'affichage est de 1920 x 1200, Traitement postérieur Allwinner SmartColor2.0 pour une excellente expérience d'affichage.
Camera	4 lanes MIPI-CSI@1.0Gbps, 13M@10fps or 8M@30fps ISP, supporting 3A/2D de-noise/defect pixel correction
Audio	2 DAC and 2 ADC, LINE-OUT/MIC-IN/Stereo headphone, 4 I2S, 8 channel DMIC, OWA OUT
Connectivity	2 USB2.0(USB Host x 1, USB OTG x 1), SDIO 3.0, SPIx3, UARTx6, TWIx6, PWM(5-ch), EMAC, GPADC, LRADC, CIR RX&TX
PMIC	Customized AXP707
WIFI/BT	Allwinner XR829 802.11 b/g/n + BT4.2, or AW859A 802.11 a/b/g/n/ac + BT 5.0
OS	Android 10.0 and above
Package	LFBGA 346balls, 12mm x 12mm body size, 0.5mm ball pitch, 0.3mm ball size

Tableau 5 Spécifications du Allwinner A133

Pour notre projet, nous avons sélectionné la carte Allwinner A133 en raison de ses capacités exceptionnelles qui répondent précisément à nos besoins en matière d'enregistrement et de transmission de données. Cette carte se distingue par son processeur puissant quad-core ARM Cortex™-A53 à 1.5 GHz, idéal pour gérer les tâches d'acquisition de données en continu sans compromettre les performances ou la consommation d'énergie.

Une des caractéristiques les plus pertinentes pour notre choix est la flexibilité de la mémoire

qu'offre la carte Allwinner A133. En plus de sa capacité interne, nous avons étendu ses possibilités avec l'ajout d'une carte mémoire externe. Cette extension nous permet de stocker des volumes importants de données, essentielles pour la constitution de notre dataset.

La capacité de la carte à prendre en charge divers types de mémoire flash, comme eMMC et NAND, associée à notre ajout d'une carte mémoire, assure une collecte de données sans interruption et une sauvegarde sécurisée. En outre, la connectivité USB et SDIO intégrée facilite le transfert efficace des données collectées vers d'autres systèmes pour le traitement et l'analyse ultérieurs.

En somme, la carte Allwinner A133 est la pierre angulaire de notre système de collecte de données, renforcée par une carte mémoire supplémentaire pour garantir une capacité de stockage étendue et fiable. Cette combinaison constitue une solution robuste et adaptée pour notre projet, permettant un enregistrement et un envoi de données efficaces et sécurisés.

2.3. Méthodes

Dans le monde en constante évolution de la gestion du trafic, disposer d'un système efficace et robuste est devenu une nécessité pour garantir la fluidité du trafic et la sécurité des usagers de la route. La complexité de la circulation dans les zones urbaines nécessite une solution technologiquement avancée, capable d'analyser, de comprendre et de réagir aux différents scénarios de trafic en temps réel. Dans cette optique, nous avons élaboré un système qui embrasse les dernières innovations en matière d'AI et de traitement d'image.

Ce système, axé principalement sur la surveillance et l'analyse du trafic routier, se distingue par sa capacité à traiter de multiples tâches allant de la collecte des données en direct à la transmission des informations pour des analyses plus poussées. Il intègre divers composants, chacun ayant une fonction spécifique, mais tous travaillant de concert pour obtenir un aperçu complet de la dynamique du trafic.

Collecte des données : La première étape de ce système consiste à acquérir les données. Une caméra intelligente est stratégiquement placée à chaque intersection. Elle est conçue pour capturer des séquences vidéo de haute qualité, permettant ainsi une analyse précise du trafic routier. Cette acquisition en temps réel est cruciale pour la fiabilité de tout le processus.

Détection et classification des objets : Après avoir capturé les images, il est essentiel d'identifier les objets, en particulier les véhicules, présents dans ces images. Pour cela, le système fait appel à l'algorithme YOLOv7-Tiny, reconnu pour sa rapidité et sa précision. Il ne

se contente pas de détecter les véhicules ; il les classifie également selon leur type (voiture, camion, moto, etc.), fournissant ainsi des données plus granulaires sur le type de trafic.

Suivi et identification des objets : Une fois les véhicules détectés et classifiés, le défi est de les suivre à travers les différentes images, en particulier dans un carrefour animé. L'algorithme Deep sort est spécialement conçu pour cela. Il garantit que chaque véhicule est suivi sans interruption, même s'il est temporairement masqué ou lorsqu'il se croise avec d'autres véhicules.

Comptage : Cette étape est centrale dans la gestion du trafic. Elle utilise une combinaison innovante de techniques de traitement d'image et d'algorithmes d'apprentissage profond. L'objectif est de compter avec précision le nombre de véhicules passant par chaque voie. Ces données quantitatives sont essentielles pour comprendre la densité du trafic à différentes heures de la journée.

Transmission des données : Enfin, après avoir traité les données localement, il est impératif de les centraliser pour une analyse plus approfondie. Le système utilise le protocole MQTT

(Transport de Télémesure par File d'Attente de Messages), optimisé pour la transmission rapide et fiable des données, garantissant ainsi que toutes les informations collectées parviennent à un serveur central. Ce serveur peut ensuite les analyser, éventuellement en conjonction avec d'autres données, pour prendre des décisions éclairées sur la gestion du trafic.

En somme, ce système offre une solution intégrée pour la surveillance et la gestion du trafic, depuis la capture des images jusqu'à l'analyse et la prise de décision basées sur les données collectées.



Figure 32 Architecture proposée pour la détection du flux de trafic. La détection, le suivi et le comptage des véhicules sont directement effectués sur le Nvidia Jetson Nano

Dans les sections suivantes, nous allons plonger dans les détails de ce système. Chaque composant, de la collecte des données à la transmission, sera examiné en profondeur, expliquant non seulement son fonctionnement, mais aussi son importance dans le cadre général de la gestion du trafic.

2.3.1. Collecte de données

La mise en place d'un système de détection de véhicules efficace dans un rond-point nécessite une collecte de données rigoureuse. Toutefois, cette étape s'avère complexe en raison de divers obstacles rencontrés sur le terrain.

Particularités de la surveillance d'un rond-point

La configuration de la caméra, notamment son angle et sa hauteur, joue un rôle crucial dans la perception des scènes capturées dans un rond-point. Dans notre situation, comme illustré dans la Figure 33, la caméra offre une vue d'ensemble du rond-point. Cette position élevée permet de couvrir l'ensemble des voies. Toutefois, elle présente également des défis :



Figure 33 Vue de la caméra

Déformation de l'apparence des véhicules : La taille et la forme des véhicules peuvent être altérées en fonction de leur position par rapport à la caméra. C'est particulièrement le cas pour les véhicules situés loin de la caméra ou ceux qui sont partiellement obstrués par d'autres véhicules ou objets sur la route.

Obstructions entre véhicules : Les routes, notamment les rond points, peuvent souvent être encombrées, ce qui conduit à des obstructions visuelles. Les véhicules se chevauchant ou se cachant les uns les autres peuvent entraîner des erreurs de détection.

Diversité des scènes : La présence simultanée de différents types de véhicules (voitures, motos, bus, camions) nécessite un modèle de détection capable de gérer cette variété.

Aléas météorologiques : La qualité de l'image peut être compromise par des conditions météorologiques défavorables. Par exemple, une luminosité trop faible ou des intempéries comme la pluie et le brouillard peuvent nuire à la capacité de détecter précisément les véhicules.

Face à ces défis, la nécessité de constituer une base de données adaptée est primordiale pour construire un modèle robuste. Ce chapitre détaille donc les étapes suivies pour la collecte des données dans ce rond-point, ainsi que les méthodes adoptées pour surmonter ces défis et garantir une détection précise et efficace des véhicules.

2.3.1.1. Acquisition des données

Avant d'entamer la construction de notre modèle de détection sur mesure, nous avons évalué

divers modèles déjà disponibles conçus à partir d'ensembles de données comme COCO. Toutefois, ces modèles n'ont pas su répondre aux exigences de précision que nous avions fixées. Cette défaillance était principalement due à une limitation majeure : nous n'utilisions qu'une seule caméra pour surveiller toutes les voies de l'intersection. Bien que cette caméra puisse aisément détecter les gros véhicules qui se trouvent à proximité, elle éprouve des difficultés avec les véhicules éloignés sur les autres voies, qui apparaissent donc plus petits à l'écran.

Pour pallier cette lacune, nous avons opté pour la création d'un ensemble de données spécifique, élaboré à partir de la même caméra fixée à l'intersection. Adopter cette stratégie nous a assuré que la détection était parfaitement adaptée aux spécificités de cette caméra. Par conséquent, la précision dans le comptage des véhicules, qui est l'essence même de notre application, a été améliorée.

Pour élaborer cet ensemble de données, nous avons mis en place un dispositif embarqué fonctionnant sous un système Linux, alimenté par la carte Allwinner A133. Nous avons configuré cette carte pour enregistrer des vidéos en continu tout au long de la journée depuis la caméra IP positionnée à l'intersection. Ces vidéos étaient sauvegardées sur une carte mémoire avant d'être transférées sur un serveur via FTP. Nous avons utilisé un script Python pour automatiser ce transfert et pour effacer les vidéos une fois qu'elles étaient correctement envoyées.

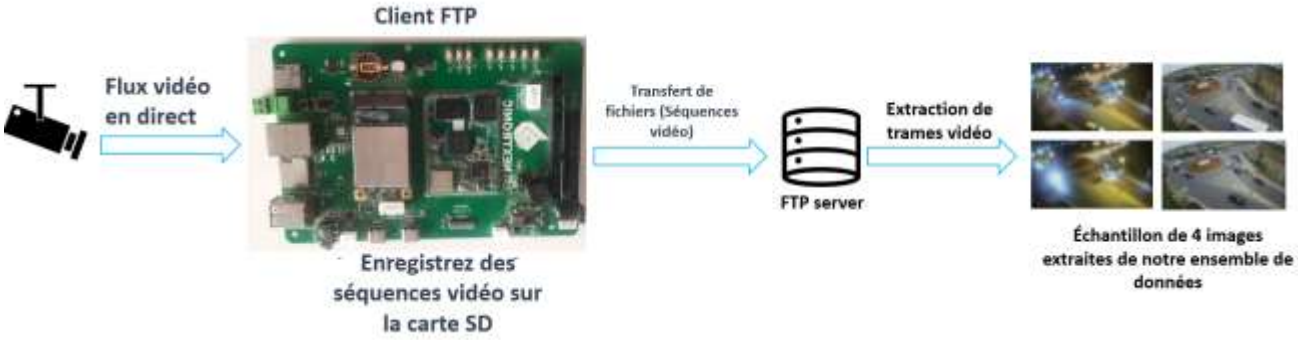


Figure 34 L'architecture du système utilisé pour la collecte de l'ensemble de données

Enfin, nous avons procédé à une sélection rigoureuse des vidéos collectées. Notre objectif était d'extraire des images offrant une grande variété, illustrant différentes catégories de véhicules (voitures, camions, bus, motos), capturées à divers moments de la journée et sous différentes conditions climatiques. Cette démarche nous a permis d'inclure dans notre base toutes les

variations potentielles rencontrées dans le monde réel.



Figure 35 Exemple de différents ensembles de données

2.3.1.2. Annotation des données



Figure 36 Annotation des véhicules sur une image de rond-point

L'annotation des images est une phase fondamentale dans la préparation des données destinées à l'entraînement du modèle YOLO. C'est au cours de cette étape que notre modèle est formé pour reconnaître et localiser spécifiquement les objets présents dans les images. Grâce à ce processus, chaque objet est délimité par une boîte englobante et associé à une catégorie appropriée (telles que voiture, camion, bus, moto).

Nous avons utilisé l'outil labelImg, un logiciel open source reconnu pour sa simplicité et son efficacité. Voici une vue d'ensemble de notre processus :

- Utilisation de labelImg : Cet outil nous a permis d'annoter manuellement chaque image de notre dataset. Lors de l'annotation, chaque objet était entouré d'une boîte englobante, et nous attribuions simultanément une étiquette correspondante. Les

informations associées étaient sauvegardées dans un format adapté à YOLO.

- Assurance qualité : Après la phase d'annotation, une vérification approfondie a été effectuée pour s'assurer de la justesse des annotations.

En mettant l'accent sur l'exactitude et la précision lors de l'annotation, nous avons jeté les bases d'un entraînement solide pour le modèle YOLO. L'importance de cette étape ne peut être sous-estimée, car elle influence directement la qualité des performances du modèle ultérieur.

La précision de notre modèle dépend grandement de la manière dont nos images sont étiquetées. Lors de la préparation de notre modèle de détection de véhicules, adapté aux conditions réelles du trafic, nous avons établi des directives précises pour l'annotation :

(1) Dans le processus de détection d'objets, tout ce qui est en dehors de la zone délimitée (ou boîte d'annotation) est considéré comme faisant partie de l'arrière-plan et non comme un objet à détecter. Par conséquent, il est essentiel d'étiqueter tous les véhicules visibles sur l'image. Cela comprend même les portions de véhicules qui pourraient être partiellement cachées.

(2) Dans les situations où les véhicules se superposent ou sont partiellement cachés par l'environnement, il est important d'étiqueter un véhicule tant que plus de la moitié de celui-ci est visible. Cependant, comme indiqué à la Figure 37, si un véhicule est masqué à un point où la majeure partie de ses caractéristiques distinctives est perdue, il est inutile de l'étiqueter car il ne fournirait pas d'informations utiles pour la détection.



Figure 37 Véhicule avec caractéristique perdue

2.3.1.3. Division des données

Notre ensemble de données se composait de 13 000 images, comme illustré à la Figure 38. Nous

avons divisé cet ensemble en trois catégories : 7 000 images pour la formation, 4 000 pour la validation et 2 000 pour les tests. Lors de cette division, nous avons veillé à maintenir un équilibre entre le nombre d'exemples dans chaque catégorie pour assurer une répartition homogène des données.

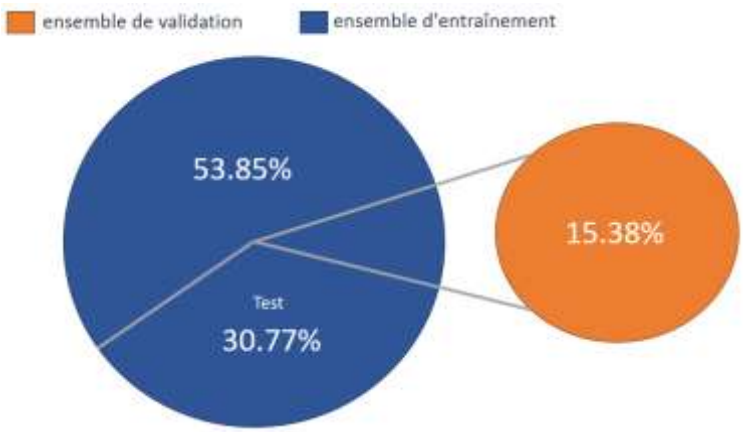


Figure 38 Répartition des données

2.3.2. Méthodes de Détection des Véhicules

2.3.2.1. Choix de YOLOv7 pour la Détection de Véhicules

L'ambition de cette phase est de déterminer la catégorie et la localisation précise du véhicule dans une image. Depuis une décennie, les techniques de détection d'objets dans les images ont principalement bifurqué en deux grandes directions. Avant 2013, les méthodes étaient essentiellement centrées sur l'utilisation de caractéristiques manuelles classiques. Cependant, après 2013, l'ère de l'apprentissage profond a débuté, transformant la détection d'objets en une discipline plus avancée et performante. Dans le domaine de l'apprentissage profond, deux techniques prédominent : l'une fondée sur les propositions de régions, incarnée par RCNN et ses variantes (comme Fast-RCNN, Faster-RCNN), et l'autre axée sur la régression, illustrée par YOLO et ses déclinaisons (tels que YOLOV3, SSD). Tandis qu-e la première technique excelle en termes de précision, la seconde est reconnue pour sa vitesse.

Pour les besoins de cette recherche, notamment la détection et la classification de véhicules, il était essentiel de choisir un algorithme performant capable de détecter en temps réel avec une grande fiabilité et pouvant être déployé sur la carte Nvidia Jetson Nano. Après avoir évalué plusieurs critères, nous avons opté pour YOLOv7 en raison de ses performances exceptionnelles, de sa renommée dans la détection d'objets vers la fin de 2022, et de sa validation réussie dans de nombreuses applications réelles [92-94]. Il est crucial de noter que le

choix du modèle doit être en harmonie avec les bibliothèques employées pour la conversion de PyTorch à TensorRT.

Chaque variante de YOLOv7 surclasse les détecteurs d'objets antérieurs en matière de rapidité et de justesse, avec des performances s'étendant de 5 FPS à 160 FPS. La figure 39 ci-après illustre bien la précision moyenne (AP) des modèles YOLOv7 en comparaison avec d'autres [95].

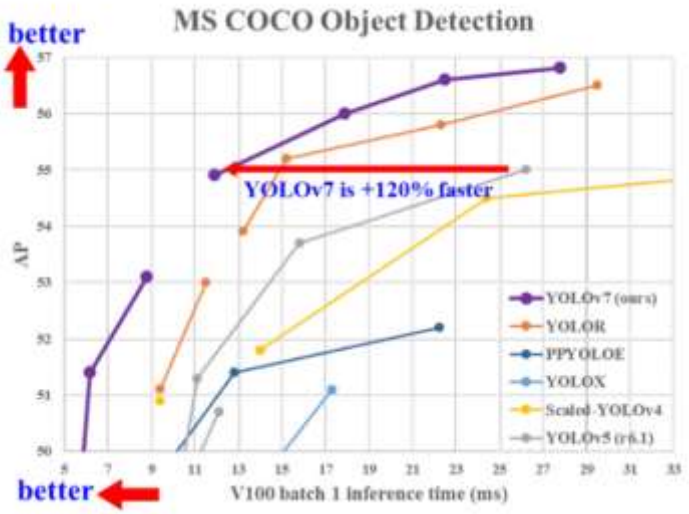


Figure 39 Résultats de YOLOv7

La figure 39 démontre clairement qu'avec l'introduction de YOLOv7, aucun autre modèle ne rivalise en termes de rapidité et de précision.

Note : les résultats évoqués ci-après sont tirés de l'article sur YOLOv7. Les expérimentations d'inférence ont été menées sur un GPU Tesla V100. Les scores de précision moyenne (AP) ont été calculés soit sur la validation COCO, soit sur l'ensemble de tests.

YOLOv7 représente une percée majeure dans le secteur de la détection d'objets en temps réel, ayant été révélé sur ArXiv en juillet 2022. Il perpétue la tradition d'innovation en termes de vitesse et de fiabilité, affichant des performances qui peuvent varier entre 5 FPS et 160 FPS, en fonction du matériel sur lequel il est exécuté. Comme son prédécesseur YOLOv4, YOLOv7 a été entraîné exclusivement sur l'ensembles de données MS COCO. Cependant, il se distingue par des modifications architecturales et des extensions conçues pour renforcer la fiabilité, sans compromettre la rapidité d'inférence, même si cela peut se traduire par une prolongation du temps nécessaire à l'entraînement. L'architecture intégrale de YOLOv7 est détaillée dans la figure 40 [96].

• L'introduction du Réseau d'Agrégation de Couches Efficaces Étendu (E-ELAN). ELAN [97] optimise l'apprentissage et la convergence des modèles profonds en garantissant un trajet de gradient optimisé. YOLOv7 présente l'E-ELAN, adapté aux structures comportant un enchaînement complexe de blocs computationnels. Il se distingue par la fusion et le mixage de caractéristiques de divers groupes, tout en préservant le trajet de gradient d'origine.

• L'ajustement proportionnel des modèles basés sur la concaténation, afin de garantir la performance optimale de la structure.

Parmi les extensions intégrées à YOLOv7, on compte :

• L'adoption de la Convolution Reparamétrisée Planifiée. Bien que YOLOv7 s'inspire de YOLOv6, il s'appuie également sur les convolutions reparamétrisées (RepConv) [98]. Cependant, une particularité a été observée concernant la connexion d'identité dans RepConv, qui impactait négativement le résiduel dans ResNet [99] et la concaténation dans DenseNet [100]. D'où l'introduction de RepConvN, dépourvue de cette connexion.

• La mise en œuvre d'étiquettes grossières pour une tête auxiliaire et d'étiquettes fines pour la tête principale, optimisant ainsi la formation.

• L'intégration de la normalisation par lots lors de l'inférence.

• L'incorporation de connaissances dérivées de YOLOR [101].

• L'utilisation de la moyenne mobile exponentielle pour le modèle d'inférence final.

2.3.2.2. YOLOv7-tiny : Un Choix Stratégique

YOLOv7-tiny est une version essentielle et optimisée de YOLOv7, spécialement conçue pour le calcul GPU en périphérie (Edge). Sa spécificité réside dans le suffixe "tiny", représentatif de son optimisation pour des tâches d'IA en périphérie et d'apprentissage profond, le rendant idéal pour une utilisation sur des dispositifs mobiles ou des serveurs et appareils en périphérie distribués. Cette version intègre la méthode de mise à l'échelle en cascade et renforce le réseau d'agrégation efficace sur de longues distances (ELAN), garantissant ainsi une précision de détection améliorée, avec moins de paramètres et une vitesse de détection accrue. Ceci est particulièrement avantageux pour des applications spécifiques. Sa structure se divise en quatre principales étapes : l'entrée, l'extraction des traits, la fusion des traits et la sortie, comme illustré sur la Figure 40. [102] Par ailleurs, tandis que d'autres versions du modèle utilisent SiLU (Unité Linéaire Pondérée par un Sigmoïde) comme fonction d'activation, YOLOv7-tiny opte pour la

fonction d'activation ReLU (Unité Linéaire Rectifiée) qui fuit. Cette particularité renforce son importance pour les applications de vision par ordinateur en périphérie.

Section d'Extraction de Traits : Cette section est constituée de blocs de convolution CBL (Convolution, Batch normalization, Leaky ReLU), d'une version optimisée de la couche ELAN-A, et d'une couche de convolution MPConv. Bien que la couche ELAN-A renforce la rapidité de l'extraction, elle pourrait diminuer sa capacité d'extraction, car elle omet deux ensembles de blocs d'extraction du Yolov7 original.

Section de Fusion de Traits : Yolov7-Tiny emprunte l'architecture PAFPN (Réseau de Pyramide de Caractéristiques avec Agrégation de Chemins) de la famille Yolov5 pour sa fusion des traits. Cette architecture fusionne les données sémantiques du FPN avec les données de localisation du PANet (Réseau d'Agrégation de Chemins), permettant un apprentissage à plusieurs niveaux. Néanmoins, la façon dont les tensors sont combinés dans cette section pourrait ne pas être idéale pour la fusion des données des couches voisines. De plus, il est possible que cette section n'accorde pas assez d'importance aux données cibles, provoquant une perte d'information.

Section de Sortie : La sortie de notre modèle emploie une tête de détection IDetect, similaire à celle du modèle YoloR. Elle met en œuvre une technique de représentation implicite pour perfectionner les prédictions basées sur les données fusionnées. Toutefois, la tête de détection de Yolov7-Tiny recourt à une convolution classique, ce qui pourrait causer des détections inadéquates.

Module de Prédiction : Avec notre jeu de données, certaines images montrent plusieurs cibles similaires très proches. Lors de l'utilisation de la suppression non maximale (NMS) traditionnelle, ces cibles peuvent être écartées du fait de leur faible score de confiance, menant à des omissions dans les détections.

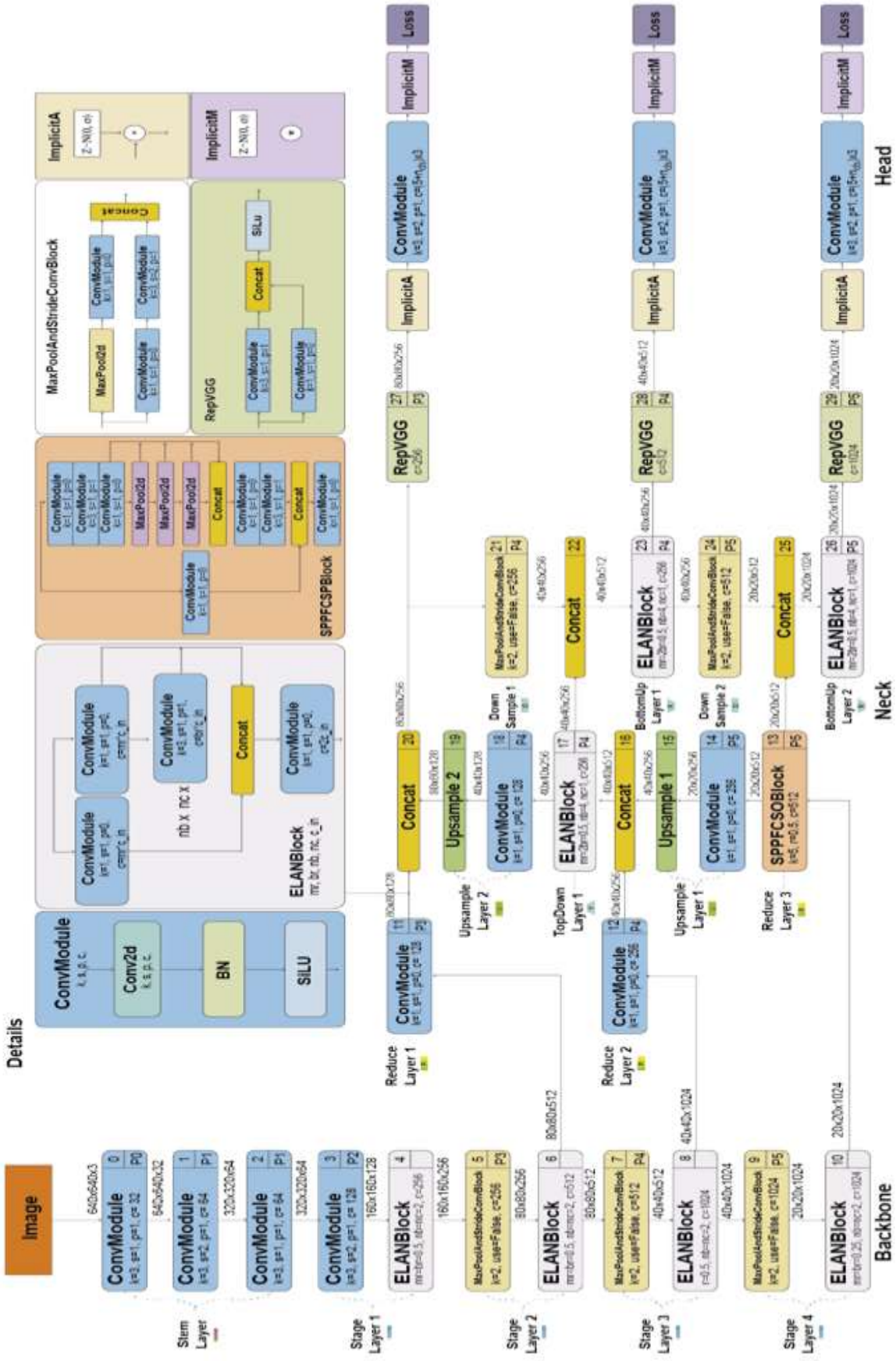


Image
Backbone
Neck
Head
Details
ConvModule
Conv2d
BN
SiLU
ELANBlock
SPPFCSPBlock
MaxPoolAndStrideConvBlock
RepVGG
ImplicitA
ImplicitM
Concat
MaxPool2d
Upsample
SPPFCSOBlock
Loss
Stem Layer
Stage Layer 1
Stage Layer 2
Stage Layer 3
Stage Layer 4
Reduce Layer 1
Reduce Layer 2
Reduce Layer 3
Upsample Layer 1
Upsample Layer 2
TopDown Layer 1
BottomUp Layer 1
BottomUp Layer 2
Down Sample 1
Down Sample 2

Figure 40 YOLOv7 Architecture. Les changements dans cette architecture incluent les blocs ELAN qui combinent les caractéristiques de différents groupes en les mélangeant et en fusionnant la cardinalité pour améliorer l'apprentissage du modèle, et un RepVGG modifié sans connexion identité. Diagramme basé sur [102]

2.3.2.3. YOLOv7 et Jetson Nano

YOLOv7 est un outil efficace pour la détection d'objets, spécialement en combinaison avec le Jetson Nano, une machine compacte et économe en énergie, idéale pour le calcul en bordure. La rapidité de détection en temps réel de YOLOv7 le rend essentiel pour de nombreuses applications nécessitant une identification précise d'éléments dans des images ou des vidéos. Le Jetson Nano, par sa faible consommation et sa taille réduite, est parfait pour diverses applications de calcul en bordure comme la surveillance vidéo, les véhicules autonomes, le contrôle du trafic et les appareils IoT avancés. En associant YOLOv7 au Jetson Nano, on peut exploiter sa détection rapide et fiable, créant ainsi des systèmes de calcul en bordure robustes et performants.

A- Paramétrage du Jetson Nano pour YOLOv7-tiny

Dans cette partie, nous aborderons l'implémentation de YOLOv7-tiny sur le Jetson Nano, une démarche significative pour les systèmes intégrés destinés à l'AI.

Cadre d'Intégration : Le Jetson Nano se distingue comme une plateforme embarquée économe en énergie, capable de faire tourner des algorithmes de Deep Learning directement sur le dispositif. L'adoption de YOLOv7-tiny est stratégique, grâce à sa version optimisée pour les contextes à capacités de calcul restreintes. Cette partie offre un aperçu théorique et pratique de l'importance de combiner YOLOv7-tiny avec le Jetson Nano.

Enjeux et Réponses lors de la Configuration : Configurer des algorithmes sur le Jetson Nano présente des défis. Il est essentiel de dépeindre la complexité d'implémenter le Deep Learning sur des dispositifs de calcul en bordure. Il faudra traiter des difficultés telles que l'installation des bibliothèques, la gestion des dépendances, et l'obtention des versions adéquates de bibliothèques comme OpenCV et Matplotlib. Lors de l'installation, ces difficultés doivent être prises en compte pour garantir une mise en place efficace de YOLOv7-tiny sur le Jetson Nano.

Optimisation des Ressources et Performances : Utilisation de l'environnement virtuel pour minimiser les conflits de dépendance, et la symbolisation de liens pour des bibliothèques globales, comme OpenCV, pour maximiser l'efficacité de l'espace de stockage. Les choix pratiques et théoriques effectués pour maximiser les performances du modèle tout en

minimisant l'utilisation des ressources devraient être mis en avant, y compris les raisons derrière le choix de versions spécifiques des bibliothèques comme PyTorch et TorchVision.

Améliorations via TensorRT : Devant ces obstacles, l'étape suivante a naturellement été la recherche de méthodes d'optimisation. C'est ici que TensorRT entre en jeu, une bibliothèque dédiée à l'optimisation permettant de transformer un modèle PyTorch en un modèle d'inférence TensorRT. Si cette démarche s'est révélée prometteuse, son application a demandé une rigueur particulière, en particulier pour la gestion des poids du modèle et l'ajustement des paramètres propres au modèle.

L'intégration du Jetson Nano avec le modèle YOLOv7-tiny a ouvert de nouvelles avenues pour la détection d'objets en temps réel dans des contextes à faible consommation de ressources. Cependant, il est crucial de considérer les défis liés à l'installation et à l'optimisation de ce tandem technologique pour assurer des performances optimales. Pour notre projet, nous avons élaboré un modèle sur mesure pour la détection et la classification des véhicules.

Ce modèle s'appuie sur l'apprentissage par transfert, en utilisant le fichier de poids pré-entraîné de YOLOv7-tiny combiné à notre jeu de données annoté. L'implémentation de YOLOv7-tiny sur le Jetson Nano a renforcé notre conviction que cet algorithme se distingue par son efficacité et son efficience pour la détection et la classification des véhicules en temps réel.

B- Déploiement et Conversion du Modèle sur la Jetson Nano

La Jetson Nano, équipée d'un GPU performant et de 4GB de mémoire, se présente comme la plateforme par excellence pour exécuter des modèles d'AI en direct. Dans le contexte de notre projet, nous avons privilégié cette carte pour la mise en service de notre modèle de détection.

Notre démarche a débuté par la transition des poids de notre modèle du format pt vers le format ONNX (Échange de Réseaux Neuronaux Ouverts). À la suite de cette mutation, nous avons conditionné la Jetson Nano pour l'accueil et l'exécution du modèle. À cette fin, la version JetPack 4.6.3, adaptée à la Jetson Nano, a été notre choix. Une fois celle-ci implantée sur une carte SD, nous avons orchestré l'installation du système d'exploitation ainsi que des outils essentiels, en capitalisant sur la vélocité du GPU de la Nano.

Avec la préparation adéquate de la Jetson Nano, le fichier ONNX a été rapatrié de l'ordinateur source à la Nano. Pour des motifs de commodité, ce fichier a été déposé sur le bureau de la Nano.

Enfin, le défi ultime résidait dans la mutation du modèle ONNX vers un format taillé pour la Jetson Nano, le format TRT (TensorRT). C'est dans ce contexte que l'instrument trtexec a été mobilisé. Inclus d'office dans la Jetson Nano, il est spécialement conçu pour transformer les modèles ONNX en TRT, optimisant ainsi l'usage du GPU. Grâce à cet outil, la conversion de notre modèle s'est déroulée sans encombre, le préparant pour une analyse en temps réel sur la Nano.

En résumé, par le biais d'un processus bien structuré, d'outils appropriés et du potentiel du GPU de la Jetson Nano, notre modèle a été brillamment déployé et affiné pour fonctionner impeccablement sur cette interface [103].

2.3.2.4. Établissement des Objectifs de Performance

Lors de la conceptualisation d'un réseau neuronal pour une tache définie, la phase initiale indispensable est la détermination de nos critères. Deux éléments centraux dans ces critères sont le taux d'erreur admissible et le degré de précision souhaité. Ces éléments guident toutes les étapes de l'élaboration, depuis la formation jusqu'à l'optimisation finale du modèle.

Nous sommes face à une tâche ardue : observer et analyser le trafic dans un rond-point à trois voies, chaque voie permettant un flux bidirectionnel. Afin d'éprouver la robustesse de notre modèle, nous l'avons mis à l'épreuve du terrain en comparant son taux d'erreur à une évaluation humaine dans un cadre similaire.

Nos objectifs sont les suivants :

- Aspirer à une détection de haute précision, tout en considérant les contraintes matérielles, la nécessité d'une réponse instantanée et l'unicité de notre base de données, soigneusement élaborée.

- S'assurer que YOLOv7-tiny garantit une vitesse d'inférence adaptée à une restitution en temps réel, visant un minimum de 24 FPS.

- Monitorer attentivement les requêtes adressées à la Jetson Nano pendant l'inférence, notamment les Opérations en Virgule Flottante par Seconde (FLOPs), afin d'ajuster notre modèle en fonction des spécifications techniques.

A- Protocole d'Évaluation de la Détection

L'évaluation de la performance des systèmes de détection est un aspect crucial pour mesurer leur efficacité et fiabilité. Les paramètres standardisés permettent de comparer objectivement

diverses approches et configurations. Ce protocole d'évaluation définit un cadre structuré pour quantifier la capacité d'un système de détection à identifier correctement et précisément les événements ou objets d'intérêt.

Critères d'évaluation

Vrai Positif (VP) : Il mesure la capacité d'un système à détecter correctement les objets ou événements qui sont présents.

Faux Positif (FP) : Le FP évalue le nombre de fois où un système signale à tort la présence d'un objet ou d'un événement. Un faible FP est essentiel pour minimiser les alertes inutiles et les erreurs de classification.

Vrai Négatif (VN) : Le VN quantifie le nombre de correctes identifications de non-présence d'objets ou d'événements. C'est un indicateur clé lorsque l'absence de détection est aussi significative que la détection elle-même.

Faux Négatif (FN) : Le FN mesure la fréquence à laquelle le système manque de détecter un objet ou un événement. Un taux élevé de faux négatifs peut indiquer une sensibilité insuffisante du système de détection.

Précision : La précision est la proportion de vrais positifs parmi tous les cas identifiés positivement par le système. Elle reflète la justesse des alertes positives du système.

Rappel (Sensibilité) : Le rappel est la proportion de vrais positifs détectés par rapport au nombre total de cas positifs réels. C'est une mesure de la capacité d'un système à détecter tous les cas pertinents.

Score F1 : Le score F1 est la moyenne harmonique de la précision et du rappel. Il fournit un seul indicateur de performance en équilibrant l'importance de la précision et du rappel, surtout lorsque ces deux mesures sont en tension.

Courbe ROC (Receiver Operating Characteristic) : La courbe ROC illustre la performance diagnostique d'un système de détection en tant que son seuil de discrimination est varié, en traçant le taux de vrais positifs contre le taux de faux positifs.

$$Précision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.1}$$

$$Rappel = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

$$ScoreF1 = \frac{2 * Précision * \text{rappel}}{Précision + \text{rappel}} \tag{4.3}$$

$$Précision = 1 - \frac{Nombre\ compté + Nombre\ réel}{Nombre\ réel} \tag{4.4}$$

2.3.3. Suivi des véhicules

Dans le domaine des applications de suivi d'objets, un défi significatif se pose lorsqu'il s'agit d'identifier de manière cohérente le même objet à travers plusieurs images dans une séquence vidéo. Deep SORT propose une solution à ce problème en attribuant une identification unique (ID) à chaque objet détecté et en utilisant une fusion de données d'apparence et de mouvement pour surveiller les mouvements de l'objet dans le temps. L'objectif central de notre projet est le comptage des véhicules, ce qui nécessite un suivi précis de tous les véhicules, une tâche cruciale dans cette application spécifique.

Le processus de suivi d'objets comprend trois étapes clés : la détection de l'objet, sa reconnaissance dans une image, et le suivi continu tout au long de la séquence jusqu'à sa sortie de la scène. Pour ce faire, nous avons choisi d'implémenter l'algorithme Deep SORT, conçu pour un suivi en temps réel et en ligne avec une métrique d'association sophistiquée. Deep SORT est une méthode de suivi multi-objets qui exploite les caractéristiques spatiales et temporelles des cibles pour assurer un suivi et une identification cohérents de tous les objets en mouvement à travers les séquences [104] (comme illustré dans la Figure 41). En attribuant un ID unique à chaque objet détecté, Deep SORT garantit l'identification et le suivi cohérents du même véhicule à travers plusieurs images, permettant ainsi un comptage précis des véhicules.



Figure 41 Identification et suivi des véhicules par Deep SORT

L'algorithme Deep SORT comporte plusieurs étapes, comme illustré dans la Figure 42 : tout d'abord, la détection des véhicules est effectuée à l'aide de YOLOv7-tiny. Ensuite, la phase de suivi multi-objets utilise le filtre de Kalman pour prédire les trajectoires en fonction de la position et de la vitesse de l'objet, tandis que l'algorithme hongrois évalue les corrélations [105]. Ce qui distingue Deep SORT, c'est son utilisation de deux couches convolutionnelles, suivies de six blocs résiduels et d'un descripteur d'apparence [105]. Cette combinaison renforce la robustesse contre les occlusions qui pourraient obscurcir les objets, réduisant ainsi considérablement les changements d'identité. Cette caractéristique en fait un excellent choix pour les applications de surveillance en temps réel.

De plus, Deep SORT a démontré son efficacité dans la gestion de scénarios complexes avec une occlusion importante, un défi courant dans le suivi des véhicules dans les ronds-points. L'algorithme est également capable de suivre des véhicules avec des vitesses et des trajectoires variables, ce qui en fait un choix idéal pour les applications du monde réel. De plus, la capacité de Deep SORT à maintenir de manière cohérente l'identité de chaque véhicule lors de sa navigation dans le rond-point améliore la précision du système de comptage des véhicules. En résumé, la fusion des méthodes de suivi traditionnelles avec les techniques d'apprentissage profond, la capacité à gérer des scénarios complexes et à maintenir l'identité des objets font de Deep SORT un choix approprié pour le suivi des véhicules dans les ronds-points, contribuant ainsi de manière significative à la haute précision de la solution de comptage des véhicules proposée.

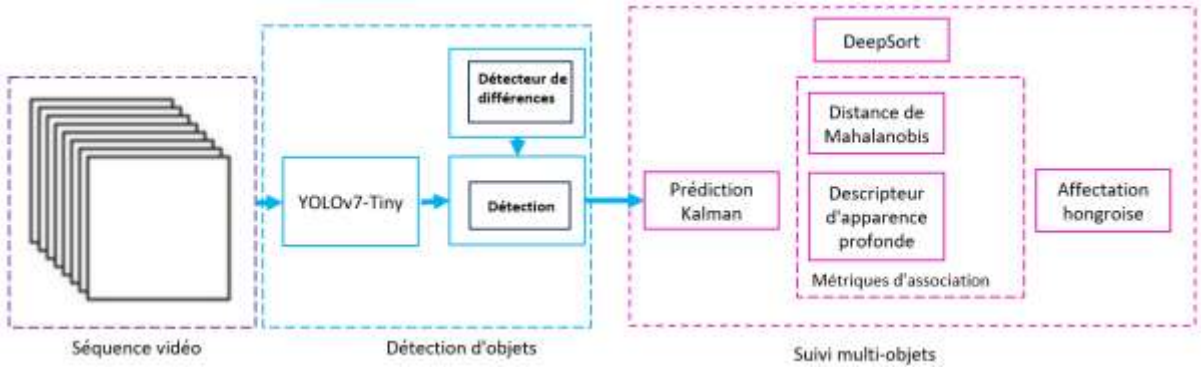


Figure 42 Architecture de l'algorithme Deep SORT

2.3.4. Évaluation du Trafic aux Carrefours

Afin de compter les véhicules circulant sur les diverses voies d'un carrefour, des lignes virtuelles spécifiques sont tracées, couvrant les six voies du rond-point. La figure 43 dépeint graphiquement la technique mise en œuvre. Il est impératif que chaque ligne s'étende d'un bord à l'autre de chaque voie dans une direction donnée. Lors de la détection et du suivi des véhicules à travers différentes images, le dispositif estime le croisement entre la ligne établie et le segment formé de deux centroïdes : celui de la position actuelle du véhicule et celui de sa position précédente (Figure 44). Quand ce segment franchit la ligne virtuelle, le dispositif augmente progressivement le décompte des véhicules et catégorise chaque véhicule détecté.

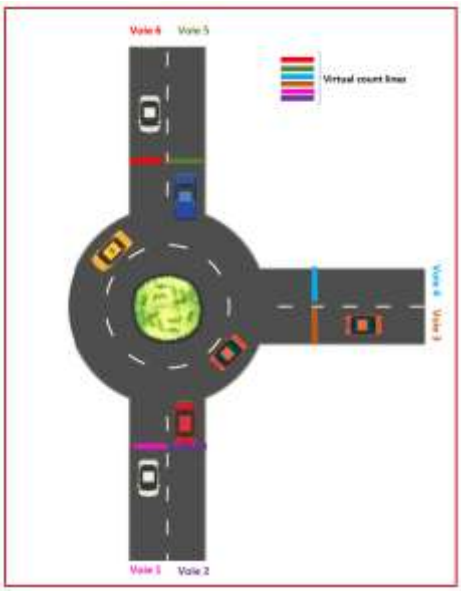


Figure 43 Schéma illustratif des lignes virtuelles utilisées pour compter les véhicules

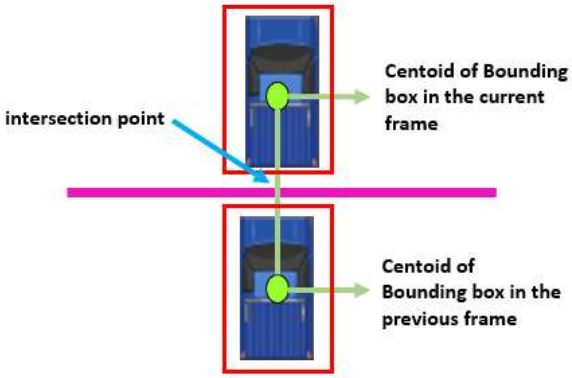


Figure 44 Mécanisme utilisé pour vérifier si une boîte englobante a été intersectée par la ligne de comptage.

Pour déterminer le nombre de véhicules changeant de voie, nous avons consigné les identifiants des véhicules empruntant chaque voie. Lorsqu'un véhicule ayant un identifiant spécifique emprunte deux voies en même temps, nous avons progressivement compté le nombre de véhicules effectuant cette action. À titre d'illustration, si un véhicule portant le numéro 1 traverse d'abord la voie 3, puis la voie 4, nous avons progressivement dénombré les véhicules allant de la voie 3 à la voie 4. La Figure 45 montre des exemples de trajectoires qu'un véhicule de la voie 2 pourrait emprunter. Selon cette illustration, un véhicule de la voie 2 pourrait rejoindre la voie 5, la voie 3 ou la voie 1, empruntant ainsi les itinéraires V2 → V5, V2 → V3 et V2 → V1. Ce même raisonnement s'applique aux autres voies de l'intersection.



Figure 45 Schéma illustrant la trajectoire possible des véhicules provenant de la Voie 2

3. Résultats

Dans cette section, nous exposons l'évaluation des performances du système déployé, l'efficacité du modèle de détection de véhicules formé avec YOLOv7-tiny, l'efficience de la méthode de comptage employée et présentons les scores de précision du comptage.

3.1. Résultats de l'entraînement

Notre modèle dédié à la détection et à la classification des véhicules a été formé à partir de 7000 images pour l'entraînement, 4000 pour la validation et 2000 pour les tests. Sa performance a été évaluée selon plusieurs critères : une mAP@.5 à 91%, un rappel de 84,1% et une précision de 87,2%.

Pour l'entraînement, nous avons utilisé l'architecture du modèle YOLOv7-tiny comme base. Cette formation a duré 300 époques et a nécessité 8,715 heures sur un ordinateur équipé d'un GPU Nvidia RTX 3060ti. La Figure 46 détaille les différentes pertes du processus. Elle inclut la perte de régression des boîtes, illustrant la manière dont ces boîtes englobent les véhicules,

ainsi que leur aptitude à identifier le centre de chaque type de véhicule. Elle contient également la perte de classification, qui met en lumière la capacité à distinguer parmi les différents objets détectés, et la perte d'objectivité, qui estime la probabilité qu'un véhicule détecté figure bien dans le ROI.

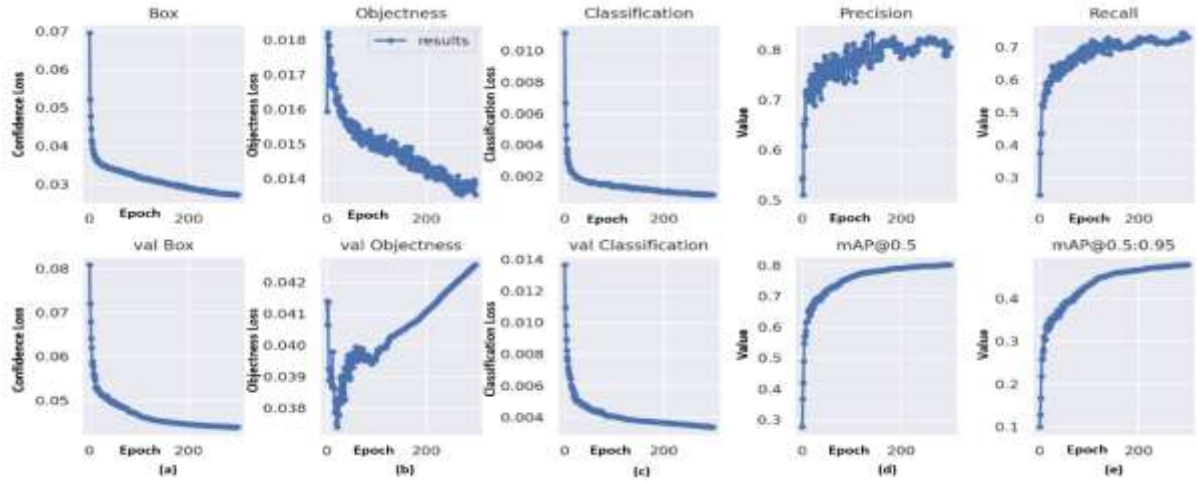


Figure 46 Résultats de l'entraînement du modèle pré-entraîné YOLOv7-tiny pour la détection d'objets. (a) Perte de Coordonnées ; (b) Perte d'Objet ; (c) Perte de Classification ; (d) Précision ; et (e) Rappel

Les résultats montrent qu'avec l'augmentation du nombre d'époques, la précision, le rappel et la mAP du modèle YOLOv7-tiny pré-entraîné pour la détection d'objets s'améliorent, démontrant l'efficacité de notre entrainement. En parallèle, les erreurs liées à l'objectivité, la classification et la boîte ont diminué.

La courbe de précision-rappel illustre le compromis entre précision et rappel pour un classificateur. Dans le cadre de notre modèle formé pour la détection et la classification des véhicules, cette courbe, représentée à la Figure 47, repose sur les données de précision et de rappel recueillies pour chaque classe (comme la voiture, le camion, le bus et la moto). D'après nos résultats, la précision moyenne (mAP) pour toutes les classes, avec un seuil de 0,5, est de 0,801. Cela démontre la haute précision de notre modèle pour détecter et classifier les véhicules.

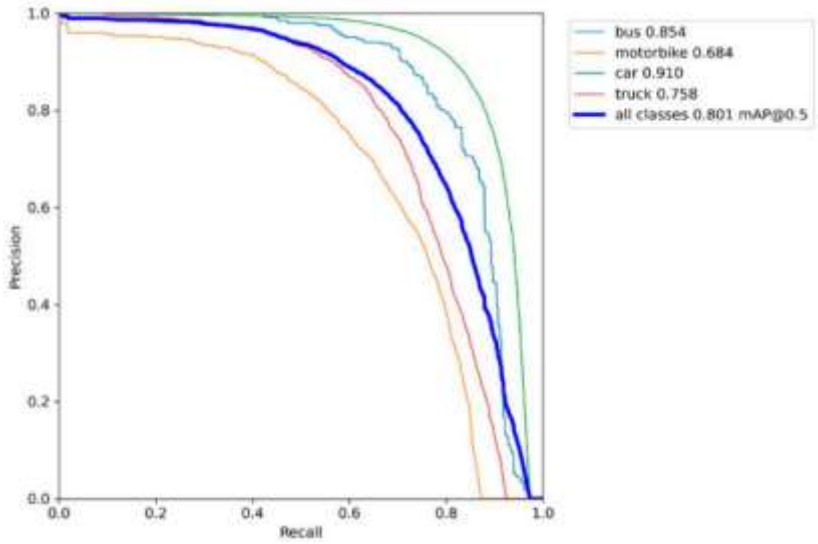


Figure 47 Courbe de précision-rappel

3.2. Résultats de comptage

Pour évaluer notre application qui compte les véhicules à un carrefour dans le monde réel, nous avons employé le Nvidia Jetson Nano comme système informatique périphérique. Le dispositif expérimental a été déployé à l'intersection entre Bouskoura et Casablanca. Nous avons utilisé le Nvidia Jetson Nano pour effectuer toutes les étapes nécessaires pour réaliser l'analyse, la détection, le suivi et le comptage. En dernière étape, les résultats du comptage ont été transmis via MQTT pour aider à la prise de décision concernant l'agrandissement du carrefour afin de réduire les embouteillages.

Les résultats de détection, de suivi et de comptage étaient affichés sur un écran connecté via HDMI. Nous présentons un exemple des résultats obtenus avec une vitesse d'exécution moyenne de 16 FPS dans la Figure 48. De plus, nous avons pris deux photos, une de jour et l'autre de nuit, pour mettre en évidence la performance de notre algorithme selon différentes conditions d'éclairage. La première photo a été prise le 24 octobre 2022 à 06 :44 :26 dans des conditions de faible luminosité, tandis que la seconde a été capturée à 13 :44 :00 le même jour. Ces résultats attestent de l'efficacité de notre solution pour détecter et suivre des véhicules dans diverses conditions lumineuses.

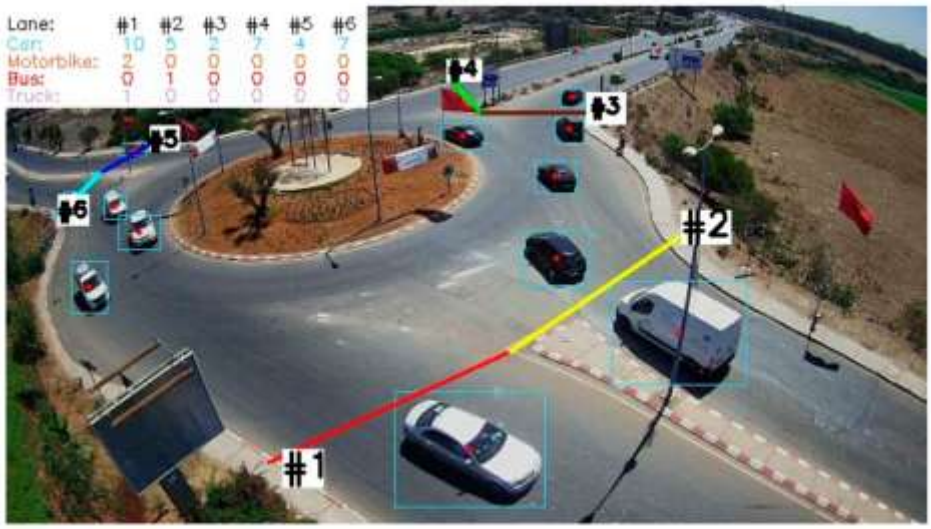


Figure 48 Exemple issu d'une simulation

3.3. Évaluation du Système

Pour évaluer l'efficacité de notre système de comptage, nous utilisons quatre critères : la précision, le rappel, la mesure F et la justesse. Ces critères ont été calculés en utilisant les paramètres suivants : vrai positif (VP) ; faux négatif (FN) ; et faux positif (FP).

Notre système a montré des résultats prometteurs dans la plupart des vidéos testées. Le (Tableau 4) affiche les résultats de comptage d'une vidéo précédemment testée, avec chacune des six voies à l'intersection. Cependant, nous avons observé des cas de faux négatifs dans les voies 4, 5 et 6, qui pourraient être dus à une détection non entièrement fiable, possiblement en raison de la grande distance entre la caméra et ces voies.

Lane	Vehicle	real number	counted number	TP	FN	FP	Precision	Recall	F-measure	Accuracy
1	Car	96	97	96	0	1	0.989	1	0.994	0.989
	Truck	12	12	12	0	0	1	1	1	1
	Motorbike	14	12	12	2	0	1	0.857	0.922	0.857
	Bus	10	10	10	0	0	1	1	1	1
2	Car	36	37	36	0	1	0.972	1	0.985	0.972
	Truck	10	9	9	1	0	1	0.9	0.947	0.9
	Motorbike	5	3	3	2	0	1	0.6	0.75	0.6
	Bus	12	12	12	0	0	1	1	1	1
3	Car	24	23	22	2	1	0.956	0.916	0.936	0.958
	Truck	6	5	5	1	0	1	0.833	0.907	0.833
	Motorbike	1	1	1	0	0	1	1	1	1
	Bus	3	3	3	0	0	1	1	1	1
4	Car	49	46	46	3	0	1	0.938	0.968	0.938
	Truck	10	8	8	2	0	1	0.8	0.888	0.8
	Motorbike	3	1	1	2	0	1	0.333	0.50	0.333
	Bus	2	1	1	1	0	1	0.5	0.666	0.5
5	Car	37	33	30	7	3	0.909	0.810	0.856	0.891
	Truck	9	7	7	2	0	1	0.777	0.875	0.777
	Motorbike	0	0	0	0	0	0	0	0	0
	Bus	1	2	1	0	1	0.5	1	0.666	0.5
6	Car	50	47	45	5	2	0.957	0.9	0.927	0.94
	Truck	13	12	12	1	0	1	0.923	0.959	0.923
	Motorbike	3	2	2	1	0	1	0.666	0.799	0.666
	Bus	3	3	2	0	1	0.666	1	0.799	1

Tableau 6 Résultats de comptage pour les six voies de l'intersection

3.4. Visualisation des Données

À partir des résultats envoyés par notre caméra intelligente installée à l'intersection, nous avons élaboré les graphiques ci-dessous. La Figure 49 présente les résultats de comptage pour la période du 5 décembre 2022 au 13 décembre 2022. Elle indique également le nombre de voitures, camions, bus et motos traversant chaque voie de l'intersection. En analysant ces chiffres, il est facile de déterminer quelles voies sont les plus encombrées, ainsi que les jours de la semaine connaissant le plus de trafic. De plus, nous avons spécifiquement choisi les jours représentés dans les graphiques car ils correspondent à une période pluvieuse, et notre modèle s'est avéré fiable pour détecter les véhicules dans diverses conditions météorologiques.



Figure 49 Résultats du comptage pour la période du 5 décembre 2022 au 13 décembre 2022

La Figure 50 présente les données transmises par notre système sous forme de graphique circulaire. Ces données illustrent le pourcentage de chaque trajet emprunté par les véhicules le 8 décembre 2022.

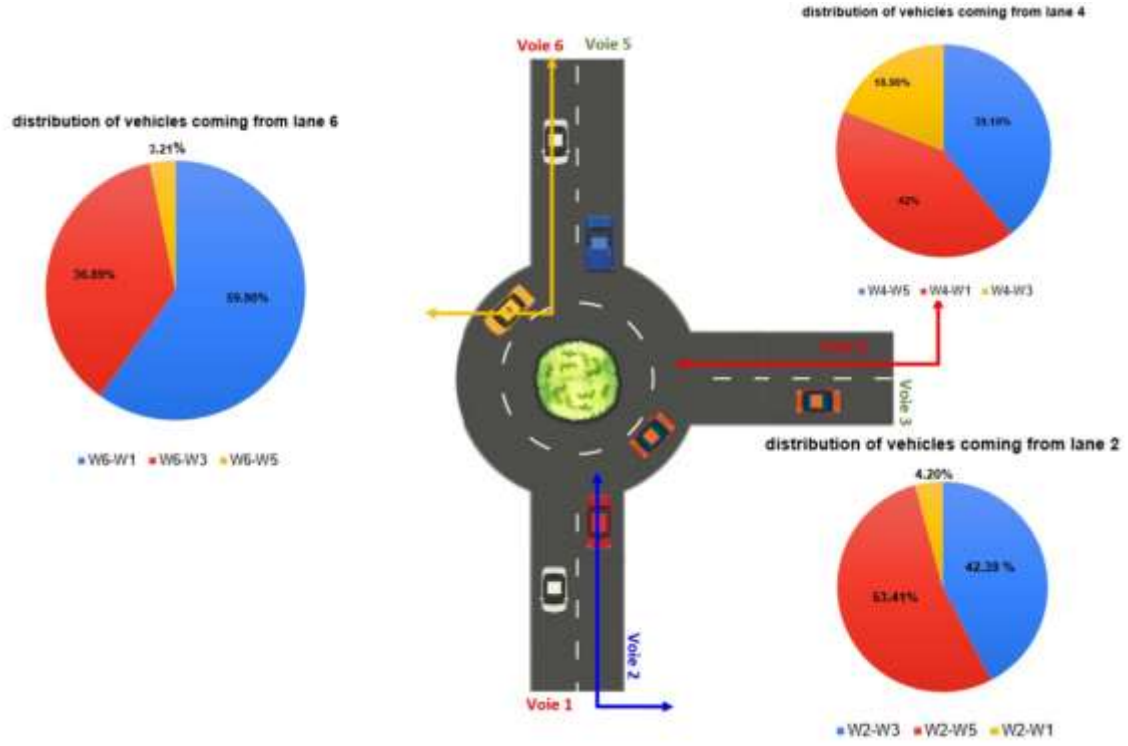


Figure 50 Diagrammes circulaires des itinéraires empruntés par les véhicules, 08/12/2022.

4. Discussion

La gestion du trafic est essentielle pour suivre l'expansion de villes telles que Bouskoura, et notre système a prouvé sa fiabilité lors de la phase expérimentale. Les données recueillies constituent un outil précieux permettant à la municipalité de prendre des décisions éclairées concernant les modifications de l'infrastructure et de mieux cerner l'impact de l'ouverture d'un nouveau centre commercial en ville. Plus précisément, notre système a aidé la municipalité à déterminer le nombre de véhicules, classés par type, passant par le rond-point. Il contribue également à évaluer le pourcentage de véhicules pour chaque voie, ce qui s'est révélé déterminant pour décider quelles voies agrandir et comment gérer la circulation afin de soutenir la croissance économique de la ville. En somme, notre système de collecte de données est un outil inestimable, lui permettant de prendre des décisions judicieuses en matière de gestion du trafic et d'adaptation de l'infrastructure pour répondre aux besoins de sa population croissante.

5. Conclusion

Ce chapitre a mis en lumière l'implémentation matérielle d'un système pour la détection et la classification des véhicules à Bouskoura, en soulignant particulièrement le rôle crucial des cartes électroniques dans l'exécution des algorithmes de Deep Learning. Notre système s'appuie sur la carte Nvidia Jetson Nano, essentielle pour faire fonctionner efficacement le modèle

YOLOv7-tiny et l'algorithme Deep SORT. Cette combinaison a prouvé sa valeur en offrant une grande précision et en résistant aux fluctuations des conditions météorologiques, bien que certains défis subsistent en termes de détection dans des conditions adverses et de la qualité du flux vidéo.

Pour relever ces défis, nous envisageons d'approfondir nos méthodes d'apprentissage et d'explorer l'usage de configurations multi-caméras. L'efficacité du Jetson Nano a été déterminante dans la validation de notre approche, contribuant significativement à la collecte de données précises pour la planification urbaine et le développement économique de Bouskoura. Les résultats obtenus témoignent de l'efficacité de notre système dans l'atténuation des problèmes de congestion, apportant une amélioration notable à la qualité de vie urbaine.

En conclusion, ce chapitre illustre non seulement l'efficacité de notre système de détection de véhicules, mais met aussi en avant l'importance capitale de l'infrastructure matérielle, notamment les cartes électroniques comme le Jetson Nano, dans la réalisation pratique des solutions de Deep Learning pour la gestion du trafic. Cette approche matériel-centrée souligne l'interaction complémentaire entre la technologie avancée et ses applications concrètes dans l'amélioration des infrastructures urbaines.

Conclusion générale et perspectives

Cette thèse apporte des contributions significatives dans le domaine de la surveillance du trafic routier, en combinant des avancées tant sur le plan algorithmique que matériel. Notre première réalisation majeure est le développement d'un algorithme de détection d'objets en mouvement, qui a dépassé les méthodes traditionnelles en termes de précision. Cet algorithme a été minutieusement testé sur des vidéos capturées sur les routes de Bouskoura, prouvant son efficacité et sa résilience face aux variations environnementales comme les changements d'éclairage et les ombres. Cette avancée résulte d'une étude exhaustive des méthodes existantes, nous permettant de créer une solution sur mesure, hautement précise.

Notre recherche a également conduit à une seconde contribution notable : le déploiement d'un algorithme de deep learning avancé pour le comptage des véhicules et l'analyse du trafic. Utilisant le modèle YOLOv7-tiny et l'algorithme Deep SORT sur la carte Nvidia Jetson Nano, ce système a été spécifiquement conçu pour évaluer le trafic à Bouskoura. Il ne se limite pas à compter les véhicules, mais calcule également le pourcentage d'utilisation de chaque voie et identifie les itinéraires les plus fréquentés par classe de véhicules.

Les résultats de cette implémentation ont été remarquablement positifs. Le système a démontré une grande précision et fiabilité dans diverses conditions climatiques tout au long de l'année, ainsi qu'en journée et en nocturne. Cette performance confirme non seulement l'efficacité de notre approche, mais valide également le choix du Jetson Nano comme plateforme matérielle. Sa robustesse et son adaptabilité ont été cruciales pour la durabilité de l'application, qui a prouvé son potentiel pour des déploiements prolongés et a eu un impact significatif sur la gestion du trafic et le développement urbain à Bouskoura.

Chaque recherche ouvre la voie à de nouvelles questions et perspectives. Dans le futur, nous envisageons de développer des cartes électroniques dédiées, comme une carte spécialement conçue pour la classification des objets, en incluant sa propre conception électronique. Cette innovation permettrait une spécialisation encore plus grande et une efficacité accrue dans le traitement des données de trafic. En outre, nous souhaitons explorer des méthodes d'apprentissage automatique plus avancées et intégrer des configurations multi-caméras pour

renforcer la détection dans des conditions extrêmes et optimiser la collecte des données.

En résumé, cette thèse a établi un équilibre réussi entre les dernières avancées technologiques et les besoins pratiques sur le terrain. Les travaux futurs devraient se concentrer sur l'amélioration de l'architecture matérielle et le raffinement des algorithmes avec des techniques d'AI de pointe. En envisageant l'extension du système à d'autres régions ou applications liées à la mobilité urbaine et à la sécurité publique, nous pourrons continuer à contribuer à l'amélioration de la qualité de vie dans les zones urbaines en expansion.

La réalisation de ces objectifs nécessitera une collaboration soutenue avec les autorités locales et une vigilance constante sur les progrès technologiques dans les domaines de la vision par ordinateur et de l'apprentissage automatique, visant toujours à assurer un développement urbain durable et centré sur les besoins de la communauté.

Liste de publications

1. Articles

- Manaa, I. E., Benjelloun, F., Sabri, M. A., Yahyaouy, A., & Aarab, A. (2023). Autonomous approach for moving object detection and classification in road applications. International Journal of Computer Aided Engineering and Technology, 18(1-3), 211-223.
- Elmanaa, I., Sabri, M. A., Abouch, Y., & Aarab, A. (2023). Efficient Roundabout Supervision: Real-Time Vehicle Detection and Tracking on Nvidia Jetson Nano. Applied Sciences, 13(13), 7416.

2. Conférences

- El Manaa, I., Benjelloun, F., Sabri, M. A., Yahyaouy, A., & Aarab, A. (2022, January). Road traffic anomaly detection: a survey. In International Conference on Digital Technologies and Applications (pp. 772-781). Cham : Springer International Publishing..
- El Manaa, I., Sabri, M. A., & Aarab, A. (2019, December). Classification and segmentation approach for detecting moving object in road applications. In 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS) (pp. 1-5). IEEE.
- Benjelloun, F., El Manaa, I., Sabri, M. A., Yahyaouy, A., & Aarab, A. (2020, June). The comparison between two methods of object detection: Fast Yolo model and Delaunay Triangulation. In 2020 International Conference on Intelligent Systems and Computer Vision (ISCV) (pp. 1-6). IEEE.

Bibliographie

[1] Branci, S. (1999). Solutions techniques pour la vidéosurveillance du trafic routier. [Rapport de recherche] Centre d'études sur les réseaux, les transports, l'urbanisme et les constructions publiques (CERTU), 176 p., figures. Disponible à HAL : ⟨hal-02163571⟩.

[2] Ahsaine, S. (2018). Politiques publiques en matière de la sécurité routière au Maroc. Revue des Etudes et Recherche en Logistique et Développement, 3, pp. 79-102. ISSN 2458-5890.

[3] Gouaillier, D. (2009). La vidéosurveillance intelligente : promesses et défis. Rapport technique, TechnoPole Defense and Security, CRIM.

[4] Bugeau, A. (2007). Détection et suivi d'objets en mouvement dans des scènes complexes, application à la surveillance des conducteurs (Thèse de doctorat). Université Rennes 1.

[5] Medjahed, F. (2012). Détection et suivi d'objets en mouvement dans une séquence d'images (Thèse de doctorat). Université des Sciences et de la Technologie d'Oran (USTO).

[6] Ye, D. H., Li, J., Chen, Q., Wachs, J., & Bouman, C. (2018). Deep learning for moving object detection and tracking from a single camera in unmanned aerial vehicles (UAVs). Electronic Imaging, 2018(10), pp. 466-1.

[7] 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.

[8] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[9] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., & Wechsler, H. (2000). Tracking groups of people. Computer Vision and Image Understanding, 80(1), pp. 42–56.

[10] Stauffer, C., & Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), pp. 747–757.

[11] Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), pp. 657–662.

[12] El Kah, S., Aqel, S., Abdelouahed, S., & Aarab, A. (2019). Region-based background modeling and subtraction using textural features. In The 1st International Conference on Embedded Systems and Artificial Intelligence (ESAI'19), Fez, Morocco, May 2–3.

[13] Kavitha, K., & Tejaswini, A. (2012). Background detection and subtraction for image sequences in video. International Journal of Computer Science and Information

Technologies, 3(5), pp. 5223–5226.

[14] Lu, N., Wang, J., Wu, Q. H., & Yang, L. (2008). An improved motion detection method for real-time surveillance. International Journal of Computer Science, 1(6), pp. 1–10.

[15] Bouwmans, T., Silva, C., Marghes, C., Zitouni, M. S., Bhaskar, H., & Frelicot, C. (2018). On the role and the importance of features for background modeling and foreground detection. Computer Science Review, 28, pp. 26–91. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2018.01.004.

[16] Devi, K. S., Malmurugan, N., & Manikandan, M. (2013). Object motion detection in video frames using background frame matching. International Journal of Computer Trends and Technology, June, 4(6), pp. 1928–1931.

[17] Aqel, S., Sabri, M. A., & Aarab, A. (2015). Background modelling algorithm based on transitions intensities. International Review on Computers and Software (IRECOS), 10(4), pp. 387–392.

[18] Aqel, S., Hmimid, A., Sabri, M. A., & Aarab, A. (2017). Road traffic: vehicle detection and classification. In 2017 Intelligent Systems and Computer Vision (ISCV), pp. 1–7.

[19] Perreault, H., Bilodeau, G.-A., Saunier, N., & Gravel, P. (2019). Road user detection in videos. Computer Vision and Pattern Recognition (cs.CV), March 28.

[20] Giraldo, J. H., Le, H. T., & Bouwmans, T. (2020). Deep learning-based background subtraction: a systematic survey. In Handbook of Pattern Recognition and Computer Vision (6th ed., pp. 51–73), April.

[21] Zgheib, T., Borges, H., Feldman, V., Guntz, T., Di Loreto, C., Desmaison, O., & Corduant, F. (2022, June). Détection d'objets en temps réel : Entraînement de réseaux de neurones convolutifs sur images réelles et synthétiques. In Conférence Nationale en Intelligence Artificielle 2022 (CNIA 2022).

[22] Idoui, R., Himeur, A., & Bouatmane, S. E. (2020). Reconnaissance d'images par les réseaux de neurones convolutionnels (CNN) (Thèse de doctorat). Université de Jijel.

[23] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014, June). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 580–587), Columbus, OH, USA.

[24] He, K., Zhang, X., Ren, S., & Sun, J. (2014, September). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Proceedings of the 13th European Conference on Computer Vision (ECCV) (pp. 346–361), Zurich, Switzerland.

[25] Girshick, R. (2015, December). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448), Santiago, Chile.

[26] Cires¸an, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. Neural Networks, 32, pp. 333–338.

[27] Zhang, L., Lin, L., Liang, X., & He, K. (2016). Is Faster R-CNN Doing Well for Pedestrian Detection? In European Conference on Computer Vision (pp. 443–457).

Springer.

[28] Yang, W., Li, Z., Wang, C., & Li, J. (2020). A Multi-Task Faster R-CNN Method for 3D Vehicle Detection Based on a Single Image. Applied Soft Computing, 95, Article ID 106533.

[29] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask r-cnn," in 2017 IEEE International Conference on Computer Vision, 2017.

[30] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961-2969).

[31] Wang, X., Hua, X., Xiao, F., Li, Y., Hu, X., & Sun, P. (2018). Multi-object detection in traffic scenes based on improved SSD. Electronics, 7, p. 302.

[32] Liu, W., Anguelov, D., Erhan, D., et al. (2016). SSD: Single Shot MultiBox Detector. In Proceedings of the ECCV 2016, pp. 21–37. Springer, Amsterdam, Netherlands.

[33] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. Journal of Chemical Engineering Data, 27, pp. 306–308, doi:10.1021/je00029a022.

[34] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA.

[35] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Available online: https://arxiv.org/abs/2004.10934.

[36] G. Jocher, K. Nishimura, T. Mineeva, and R. Vilariño, YOLOv5, 2020.

[37] Menghan, Z., Zitian, L., & Yuncheng, S. (2020, June). Optimization and comparative analysis of YOLOV3 target detection method based on lightweight network structure. In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 20-24. IEEE.

[38] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. (2019) Pytorch: An imperative style, highper formance deep learning library. In Advances in Neural Information Processing Systems *32* (pp. 8024–8035), Curran Associates, Inc., Red Hook, NY, USA.

[39] Wang, C.; Bochkovskiy, A.; Liao, H.M. (2017) YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv 2022, arXiv:2207.02696

[40] Zhang, T., Xu, C., & Yang, M.-H. (2017). Multitask Correlation Particle Filter for Robust Object Tracking. In IEEE Conference on Computer Vision and Pattern Recognition.

[41] Shantaiya, S., Verma, K., & Mehta, K. (2015). Multiple Object Tracking Using Kalman Filter and Optical Flow. European Journal of Advances in Engineering and Technology, 2(2), pp. 34-39.

[42] Chantara, W., Mun, J.-H., Shin, D.-W., & Ho, Y.-S. (2015). Object Tracking Using Adaptive Template Matching. IEIE Transactions on Smart Processing and Computing, 4(1), February 2015.

[43] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, (2019). Tracking without bells and whistles," arXiv.

[44] Bochinski, E., Sikora, T. (2017, April). High-Speed Tracking-by-Detection Without Using Image Information. In Proceedings of the International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS, Lecce, Italy.

[45] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and Realtime tracking with a deep association metric. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 3645-3649). IEEE.

[46] Sindagi, V. A., & Patel, V. M. (2018). A survey of recent advances in CNN-based single image crowd counting and density estimation. Pattern Recognition Letters, 107, pp. 3–16.

[47] Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. Advances in Neural Information Processing Systems, pp. 1324–1332.

[48] Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 589–597

[49] Onoro-Rubio D, López-Sastre RJ (2016) Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, Springer, pp 615–629

[50] Amato, G., Bolettieri, P., Moroni, D., Carrara, F., Ciampi, L., Pieri, G., ... & Vairo, C. (2018, December). A wireless smart camera network for parking monitoring. In 2018 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.

[51] Ciampi, L., Amato, G., Falchi, F., Gennaro, C., & Rabitti, F. (2018, June). Counting Vehicles with Cameras. In SEBD.

[52] Barandiaran, J., Murguia, B., & Boto, F. (2008). Real-time people counting using multiple lines. pages 159–162.

[53] Embleton, K. V., Gibson, C. E., & Heaney, S. I. (2003). Automated counting of phytoplankton by pattern recognition: a comparison with a manual counting method. Journal of Plankton Research, 25(6), pp. 669–681.

[54] Antonini, G., & Thiran, J. P. (2006). Counting pedestrians in video sequences using trajectory clustering. IEEE Transactions on Circuits and Systems for Video Technology, 16(8), pp. 1008–1020.

[55] Song, H., Wang, X., Hua, C., Wang, W., Guan, Q., & Zhang, Z. (2018). Vehicle trajectory clustering based on 3D information via a coarse-to-fine strategy. Soft Computing, 22(5), pp. 1433–1444.

[56] Zeng, C., & Ma, H. (2010). Robust head-shoulder detection by PCA-based multilevel HOG-LBP detector for people counting. pages 2069–2072.

[57] Yang, H., & Qu, S. (2017). Real-time vehicle detection and counting in complex

traffic scenes using background subtraction model with low-rank decomposition. IET Intelligent Transport Systems, 12(1), pp. 75–85.

[58] Sun, M., Wang, Y., Li, T., Lv, J., & Wu, J. (2017). Vehicle counting in crowded scenes with multi-channel and multi-task convolutional neural networks. Journal of Visual Communication and Image Representation, 49, pp. 412–419.

[59] Bouaich, S., Mahraz, M. A., Riffi, J., & Tairi, H. (2018). Vehicle counting system in real-time. In 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1–4.

[60] Dai, Z., Song, H., Wang, X., Fang, Y., Yun, X., Zhang, Z., & Li, H. (2019). Video-based vehicle counting framework. IEEE Access, 7, pp. 64460–64470.

[61] Liu, F., Zeng, Z., & Jiang, R. (2017). A video-based real-time adaptive vehicle-counting system for urban roads. Plos One, 12(11): e0186098.

[62] Zhang, S., Wu, G., Costeira, J.P., & Moura, J.M.F. (2017). FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, pp. 3687–3696, doi:10.1109/ICCV.2017.396.

[63] Bui, K.H.N., Yi, H., Jung, H., & Cho, J. (2020). Video-Based Traffic Flow Analysis for Turning Volume Estimation at Signalized Intersections. In Proceedings of the 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS), Phuket, Thailand, pp. 152–162, doi:10.1007/978-3-030-42058-1_13.

[64] Xiang, X., Zhai, M., Lv, N., & El-Saddik, A. (2018). Vehicle Counting Based on Vehicle Detection and Tracking from Aerial Videos. Sensors, 18, 2560, doi:10.3390/s18082560.

[65] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G.R. (2011). ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, pp. 2564–2571, doi:10.1109/ICCV.2011.6126544.

[66] Rosas-Arias, L., Portillo-Portillo, J., Hernandez-Suarez, A., Olivares-Mercado, J., Sanchez-Perez, G., Toscano-Medina, K., Perez-Meana, H., Sandoval Orozco, A. L., & Garcia Villalba, L. J. (2019). Vehicle counting in video sequences: An incremental subspace learning approach. Sensors, 19, p. 2848, June 2019.

[67] Lücking, M., Rivera, E., Kohout, L., Zimmermann, C., Polad, D., & Stork, W. (2020). A video-based vehicle counting system using an embedded device in realistic traffic conditions. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221094.

[68] Pham, L. H., Phan, H. N., Chung, N. M., Vu, T., & Ha, S. V. (2020). A robust multiclass vehicle detection and classification algorithm for traffic surveillance system. In 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1–6.

[69] Popat, P., Sheth, P., & Jain, S. (2019). Animal/object identification using deep learning on raspberry pi. In Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 1 (pp. 319-327). Springer Singapore.

[70] Valladares, S.; Toscano, M.; Tufiño, R.; Morillo, P.; Vallejo-Huanga, D. Performance Evaluation of the Nvidia Jetson Nano Through a Real-Time Machine Learning Application. In Intelligent Human Systems Integration; Russo, D., Ahram, T., Karwowski, W., Di Bucchianico, G., Taiar, R., Eds.; Springer: Cham, Switzerland, 2021; Volume 1322.

[71] Khachatryan, T. B. (2022). Comparison of Jetson Xavier NX and Coral Dev Board for solving object detection task. In Science and Technology Research 2022 (pp. 88-92).

[72] Mansour Mohamed, A.S.M.M.; Rashid, M.M. Video-Based Vehicle Counting and Analysis using YOLOv5 and DeepSORT with Deployment on Jetson Nano. Asian J. Electr. Electron. Eng. 2022, 2, 11–20. Available online: https://journals.alambiblio.com/ojs/ index.php/ajoeee/article/view/34 (accessed on 1 March 2023).

[73] Dinh, D.-L.; Nguyen, H.-N.; Thai, H.-T.; Le, K.-H. Towards AI-Based Traffic Counting System with Edge Computing. J. Adv. Transp. 2021, 2021, 5551976. [CrossRef]

[74] Valladares, S.; Toscano, M.; Tufiño, R.; Morillo, P.; Vallejo-Huanga, D. Performance Evaluation of the Nvidia Jetson Nano Through a Real-Time Machine Learning Application. In Intelligent Human Systems Integration; Russo, D., Ahram, T., Karwowski, W., Di Bucchianico, G., Taiar, R., Eds.; Springer: Cham, Switzerland, 2021; Volume 1322.

[75] Han, Y.; Oruklu, E. Traffic sign recognition based on the NVIDIA Jetson TX1 embedded system using convolutional neural networks. In Proceedings of the 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 184–187. [CrossRef]

[76] Wang, Z., Bai, B., Xie, Y., Xing, T., Zhong, B., Zhou, Q., ... & Chai, H. (2020). Robust and fast vehicle turn-counts at intersections via an integrated solution from detection, tracking and trajectory modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 610-611).

[77] Piccardi, M. (2004, October). Background subtraction techniques: a review. In 2004 IEEE international conference on systems, man, and cybernetics (IEEE Cat. No. 04CH37583) (Vol. 4, pp. 3099-3104). IEEE.

[78] R. Jain, et H. H. Nagel, (1979). On the analysis of accumulative difference pictures from image sequences of real-world scenes. IEEE transactions on pattern analysis and machine intelligence,

[79] Huang, D. Y., & Wang, C. H. (2009). Optimal multi-level thresholding using a two-stage Otsu optimization approach. Pattern Recognition Letters, 30(3), 275-284.

[80] Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. International Journal, 1(6), 90-95.

[81] Dhankhar, P., & Sahu, N. (2013). A review and research of edge detection techniques for image segmentation. International Journal of Computer Science and Mobile Computing, 2(7), 86-92.

[82] Mandloi, G. (2014). A survey on feature extraction techniques for color images. International Journal of Computer Science and Information Technologies, 5(3), pp. 4615-4620.

[83] Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing (2nd ed.). Prentice Hall.

[84] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, (6), pp. 610-621.

[85] Leray, P., & François, O. (2004). Réseaux bayésiens pour la classification : Méthodologie et illustration dans le cadre du diagnostic médical. Rev. d'Intelligence Artif., 18(2), pp. 169-193.

[86] Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.

[87] Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P., & Kevitt, P. M. (2008, October). On points geometry for fast digital image segmentation. In The 8th International Conference on Information Technology and Telecommunication. Ireland: IEEE, pp. 54-61.

[88] Kim, Z. (2006, September). Realtime obstacle detection and tracking based on constrained Delaunay triangulation. In 2006 IEEE Intelligent Transportation Systems Conference (pp. 548-553). IEEE.

[89] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

[90] Nvidia Corporation. Jetson NANO Module. Available online: https://developer.nvidia.com/embedded/jetson-nano (accessed on 2 January 2023).

[91] CNX Software. Allwinner A133 tablet processor pairs with XR829 or AW859A WiFi & Bluetooth chip. CNX Software. https://www.cnx-software.com/2020/10/31/allwinner-a133-tablet-processor-pairs-with-xr829-or-aw859a-wifi-bluetooth-chip/

[92] Uday, P.; Shikha, B.S.; Uday, P.; Shubham, S. Using YOLO V7: Development of Complete VIDS Solution Based on Latest Re quirements to Provide Highway Traffic and Incident Real-Time Info to the ATMS Control Room Using Artificial Intelligence. SSRN 2022, 4313791. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4313791 (accessed on 20 December 2022).

[93] Jiang, K.; Xie, T.; Yan, R.; Wen, X.; Li, D.; Jiang, H.; Jiang, N.; Feng, L.; Duan, X.; Wang, J. An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation. Agriculture 2022, 12, 1659. https://doi.org/10.3390/agri culture12101659. 15.

[94] Wang, Y.; Wang, H.; Xin, Z. Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. IEEE Access 2022, 10, 133936–133944. https://doi.org/10.1109/access.2022.3230894.

[95] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-

of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7464-7475).

[96] Yang, F., Zhang, X., & Liu, B. (2022). Video object tracking based on YOLOv7 and DeepSORT. arXiv preprint arXiv:2207.12202.

[97] Wang, C.-Y., Liao, H.-Y. M., & Yeh, I.-H. (2022). Designing network design strategies through gradient path analysis. arXiv preprint arXiv:2211.04800.

[98] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13733–13742).

[99] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778).

[100] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700–4708).

[101] Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You only learn one representation: Unified network for multiple tasks. arXiv preprint arXiv:2105.04206.

[102] M. Contributors, "YOLOv7 by MMYOLO." https://github.com/open-mmlab/mmyolo/tree/main/ configs/yolov7, 2023. Accessed: May 13, 2023.

[103] Ananth Vivekanand, (2022, AUG 1). How to Deploy YOLOv7 to a Jetson Nano. https://blog.roboflow.com/how-to-deploy-yolov7-to-jetson-nano/

[104] Lamouik, I., Yahyaouy, A., & Sabri, M.A. (2023). Model Predictive Control for Full Autonomous Vehicle Overtaking. Transportation Research Record: Journal of the Transportation Research Board, 2677, pp. 1193–1207.

[105] Mandal, V., & Adu-Gyamfi, Y. (2020). Object Detection and Tracking Algorithms for Vehicle Counting: A Comparative Analysis. Journal of Big Data Analytics in Transportation, 2, pp. 251–261.