الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ YAHIA FARES MÉDÉA
DÉPARTEMENT DES MATHÉMATIQUES ET D'INFORMATIQUE



Mémoire de Projet de Fin d'Etudes
Présenté pour l'obtention du diplôme De master en informatique
Spécialité : Ingénierie des Systèmes et Technologie de Web

Détection et comptage des véhicules avec Deep Learning

Présenté par :
ABDOUCHE Anissa
ALLOUCHE Maroua

Proposé et dirigé par :
Dr HENTABLI Hamza

promotion 2021

ملخص ـ قدمنا نظام تحليل لحركة المرور من خلال فيديو يعتمد علىٰ تقنيات الرؤية الحاسوبية الهدف الاساسي من هذه المذكرة هو تطوير نموذج قادر على تقييم عدد المركبات في الزمن الحقيقي في الطرقات الحضرية بمساعدة تكنولوجيا الرؤية الحاسوبية . من خلال هده المذكرة نقترح خوارزمية لتحديد المركبات وحسابها والكشف عن سطح السيارات ولتحديد السيارات المكتشفة نقوم بوضع مربعات زرقاء حولها لدقة تحدبد وحساب اكبر من 90 في وقت فعلي
كلمات مفتاحية ـ تحليل حركة المرور بالفيديو ،كشف المركبات ، الخوارزمية الآلية ، الرؤية الحاسوبية التقنيات، شكل الكشف .

Abstract–We present a video traffic analysis system based on computer vision techniques. The main objective of this brief is to develop an adaptive model capable of assessing the number of vehicles in real time on urban roads using computer vision technologies. This memory proposes an automatic algorithm for vehicle detection and an adaptive model for vehicle counting based on the methods the form detection is equivalent to detecting the vehicle surface and to highlighting the detected vehicles, we draw blue boxes around them. The results show that the system is robust, with real-time counting accuracy greater than 99 % in most field scenarios.
Keywords : Traffic video analysis, vehicle detection, automatic algorithm, computer vision techniques, form detection

Résumé–Nous présentons un système d'analyse vidéo du trafic basé sur des techniques de vision par ordinateur. L'objectif principal de ce mémoire est de développer un modèle adaptatif capable d'évaluer le nombre de véhicules en temps réel sur les routes urbaines à l'aide de technologies de vision par ordinateur. Ce mémoire propose un algorithme pour la détection des véhicules et un modèle adaptatif pour le comptage des véhicules basés sur les méthodes la détection de forme est Équivalente à détecter la surface de véhicule et pour mettre en évidence les véhicules détectés, Nous dessinons des cases bleues autour d'elles . Les résultats montrent que le système est robuste, avec une précision de comptage en temps réel supérieure à 99 % dans la plupart des scénarios sur le terrain.
Mots-clés : Analyse vidéo du trafic, détection des véhicules, algorithme automatique, techniques de vision par ordinateur, détection de forme.

Remerciements

Tout d'abord, on remercie dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Un profond respect et un remerciement particulier à notre encadreur **Mr. Hentabli.Hamza** pour ses conseils et sa disponibilité, sa gentillesse et la patience qu'il a manifeste dans ce projet.

Nous adressons aussi nos remerciements à tous les enseignants et au corps administratif de la faculté des Sciences de Médéa.

Un profond remerciement pour les membres de jurées quiont accepté d'évaluer ce travail.

Dédicace

Je dédie avec grand plaisir, ce modeste travail à :
Tout d'abord je tiens à remercier Dieu Tout puissant ; Je puisse les honores ;
A mon père ;
A ma chère mère ;
A mes chères sœurs et frères ; A toute ma famille ;
A mon binôme dans ce travail Et tous mes amis proches ;
Et a Tous mes collègues promotion 2021.

ANISSA

Je dédie avec grand plaisir, ce modeste travail à :
Tout d'abord je tiens à remercier Dieu Tout puissant ; Je puisse les honores ;
A mon père ;
A ma chère mère ;
A mes chères sœurs et frères ;
A toute ma famille ;
A mon binôme dans ce travail Et tous mes amis proches ;
Et a Tous mes collègues promotion 2021.

Maroua

Table des Matières

Liste des Figures **8**

Liste des Abréviations **1**

Chapitre I: Introduction générale **15**
I.1 Contexte et problématique . . . 15
I.2 But et Objectifs . . . 16
I.3 Structure du mémoire . . . 16

Chapitre II: Revue litterature **18**
II.1 Introduction . . . 18
II.2 Apprentissage automatique pour la détection d'objets . . . 18
II.2.1 Problème de la reconnaissance d'objets . . . 18
II.2.2 Pourquoi compter les véhicules ? . . . 19
II.2.3 Sélection des caméras . . . 19
II.3 Les différents méthodes de comptage de véhicule . . . 19
II.3.1 Manuel comptant . . . 19
II.3.2 Dénombrement des vehicles . . . 20
II.3.3 Comptage automatisé . . . 21
II.3.3.1 Recherche sur le comptage automatisé . . . 21
II.3.4 Architecture du système UAV . . . 22
II.3.5 Détection de véhicule avec YOLOv3 . . . 23
II.3.5.1 La méthode YOLO . . . 24
II.3.5.2 YOLOv3 . . . 25
II.3.6 Méthode de comptage de véhicules utilisant le Mask - Rcnn . . . 25
II.3.7 DÉTECTION D'OBJETS BASÉE SUR CNN . . . 26
II.4 les réseaux neuronaux convolutifs (CNN) . . . 27
II.4.1 Que sont les réseaux neuronaux convolutifs ? . . . 27
II.4.1.1 Introduction . . . 27
II.4.2 input et output . . . 28
II.4.3 Structure du réseau neuronal convolutif . . . 28
II.4.3.1 Convolution . . . 29
II.4.3.2 Profondeur . . . 29
II.4.3.3 Stride . . . 30
II.4.3.4 Zero-Padding . . . 30
II.4.3.5 Calque ReLu . . . 30
II.4.4 Mise en commun maximale . . . 31

II.4.4.1 Aplatissement 32
II.4.4.2 Connexion complète 32
II.4.4.3 Rassembler tout cela formation à l'aide de la rétropropagation 33
II.5 Neural Convolutionnel Classique 33
II.5.1 Architecteurs de réseau 33
II.5.1.1 LeNet-5 33
II.5.1.2 AlexNet 33
II.5.1.3 VGGNet 34
II.6 Conclusion 35

Chapitre III: conceptions du système de détection et comptage 37
III.1 Introduction 37
III.2 Méthodologie 37
III.2.1 Dataset 37
III.3 Le recadrage (crop) les images 40
III.3.1 Introduction 40
III.3.2 Modèle de classification CNN : 40
III.3.2.1 Transfer Learning 42
III.3.3 Détection et comptages des véhicules 44
III.3.3.1 Différenciation des images 45
III.3.3.2 Trouver des contours 46
III.3.3.3 Dilatation d'image 47
III.3.3.4 Tracking 47
III.3.3.5 Comptage (Counting) 48
III.4 Conclusion 48

Chapitre IV: Résultat de détection et comptage 50
IV.1 Introduction 50
IV.2 Environnement du travail 50
IV.2.1 Environnement matériel ou le hardware 50
IV.2.2 Environnement immatériel ou le software 50
IV.3 les Fonctions utiles de la bibliothèque OpenCV 51
IV.3.1 Schéma général 51
IV.3.2 3.2 Détection d'objets en mouvement 51
IV.3.3 Pour le traitement de vidéo 52
IV.3.4 Pour le traitement d'image 52
IV.3.4.1 Transformer une image en niveau de gris 52
IV.3.4.2 Découpage des différents objets de l'image 52
IV.3.4.3 Segmentation : 53
IV.3.4.4 Minimisations des bruits : 54
IV.3.4.5 Détection de formes : 55
IV.4 Les resultat de la tecnique OMG2 56
IV.4.1 Arrière-planSubtractorMOG2 56
IV.4.2 Segmentation : 57
IV.4.3 Minimisations des bruits : 57

IV.4.4 Détection de formes : 58
IV.5 comptages 58
IV.6 conclusion 59

Conclusion Générale **61**
Perspectives 63
Bibliographie 65

Liste des Figures

FIGURE II.1 Composants du système de transport intelligent basé sur la vision. 22
FIGURE II.2 Cadre de la méthode proposée 23
FIGURE II.3 Flux global de la méthodeYOLO 24
FIGURE II.4 Architecture générale du système de comptage des véhicules 25
FIGURE II.5 la détection de véhicule par mask-RCNN 26
FIGURE II.6 Moyenne de précision (MAP) des cinq modèles de détection d'objets sur les cinq ensembles d'essai (un de chacune des quatre installations de caméra et un cinquième ensemble d'essai combiné [7]. 27
FIGURE II.7 Une vue générale de CNN. 28
FIGURE II.8 Cnn image d'entrée et sortie image 28
FIGURE II.9 Quatre étapes de Cnn. 29
FIGURE II.10 Deux fonctions d'activations (a) ReLU et (b) SoftMax 30
FIGURE II.11 La différence entre Average Pooling et Max Pooling. 31
FIGURE II.12 Deux couches convolutionnelles suivies d'une couche entièrement connectée. 32
FIGURE II.13 Architecture de LeNet-5 33
FIGURE II.14 Architecture AlexNet. 34
FIGURE II.15 Architecture VGGNet. 34

FIGURE III.1 Data set video M6 Motorway Traffic 38
FIGURE III.2 Dataset video route 38
FIGURE III.3 extraction et enregistrement d'image vidéo 39
FIGURE III.4 image de la vidéo extraite de data 40
FIGURE III.5 Model summary 41
FIGURE III.6 Model accuracy 42
FIGURE III.7 idée de transfert d'apprentissage 43
FIGURE III.8 Le diagramme représente le transfert profond basé sur l'extraction de la fonction 44
FIGURE III.9 la détection des voitures en mouvement dans une vidéo 45
FIGURE III.10image binaire. 46
FIGURE III.11trouver des contours sur l'image 47
FIGURE III.12la dilatation d'image. 47

FIGURE IV.1 schéma général du travail 51
FIGURE IV.2 image prise de la vidéo M6 Motorway Traffic 52
FIGURE IV.3 image prise de la vidéo alger.mp4 53
FIGURE IV.4 image prise de la vidéo route.mp4 53

FIGURE IV.5 segmentation image de vidéo M6 Motorway Traffic 54
FIGURE IV.6 segmentation image de vidéo route.mp4. 54
FIGURE IV.7 éliminé les pixels isolés vidéo M6 Motorway Traffic 55
FIGURE IV.8 éliminé les pixels isolés vidéo alger.mp4 55
FIGURE IV.9 éliminé les pixels isolés vidéo route.mp4 55
FIGURE IV.10 Détection de forme vidéo M6 Motorway Traffic 56
FIGURE IV.11 Détection de forme vidéo alger.mp4 56
FIGURE IV.12 Détection de forme vidéo route.mp4 56
FIGURE IV.13 Segmentation OMG vidéo route.mp4 57
FIGURE IV.14 Minimiser le bruit vidéo route.mp4 57
FIGURE IV.15 Détection de forme OMG vidéo route.mp4 58
FIGURE IV.16 Résultat du comptage de la video Alger.mp4 58
FIGURE IV.17 Résultat du comptage de la video route.mp4 59

Liste des Abréviations

CNN	Convolutional Neural Network
CCTV	closed-circuit television.
Faster R-CNN	Faster Region based Convolutional neural network .
SSD	Single Shot Detector.
YOLO	You Only Look Once.
UAV	Unmanned Aerial vehicle
YOLOv3	You Only Look Once v3
VGGnet	Visual Geometry Group Neural Networks
ML	Automatic Learning
DL	Deep Learning
CNNs	Convolutionals Neurals Networks
ANNs	artificial Neurals Networks
R-CNN	Réseaux de neurones convolutifs basés sur la région
Fast R-CNN	Fast Region based Convolutional neural network
CV	computer vision

Chapitre I :

Introduction générale

Chapitre I

Introduction générale

Avec le développement rapide de l'économie et de la technologie dans la société moderne, les automobiles sont devenues un moyen de transport indispensable dans les déplacements quotidiens des personnes.

Au cours des dernières décennies, l'industrie automobile a connu un développement spectaculaire. De nouvelles technologies émergent de temps en temps, ce qui rend nos vies plus commodes. Entre-temps, en raison du nombre croissant de véhicules, le problème du transport se pose. Les principaux défis pour le réseau de transport sont la congestion routière, les effets sur l'environnement, la consommation d'énergie, les dangers pour la sécurité, les coûts d'entretien élevés et l'occupation des terres.

I.1 Contexte et problématique

Il peut être utile de suivre les lignes directrices sur le dénombrement automatique pour réussir un sondage. Toutefois, les lignes directrices actuelles sur le comptage manuel ne sont pas générales, c'est-à-dire qu'elles ne s'appliquent qu'à un territoire et à une condition particuliers. Ainsi, le premier objectif de cette recherche est de formuler des lignes directrices générales pour les enquêtes de comptage automatique en général.

Il n'y a pas de terrain réel pour estimer les erreurs dans les dénombrements manuels sans effectuer des dénombrements répétés. Mais le fait d'effectuer des dénombrements répétés pour tous les sites augmente le coût et le temps d'un projet. Cette recherche propose quelques formules pour estimer les erreurs lorsque des dénombrements répétés pour un échantillon de sites sont considérés comme des dénombrements réels. Cette étude développe des algorithmes pour appliquer le détecteur d'objets pour effectuer des comptages automatiques à partir de vidéos préenregistrés.

Le comptage manuel est coûteux et long. De nombreuses technologies sont disponibles dans nos jours qui peut être bonne.

En plus de mettre à jour les réseaux routiers, les administrations locales mettent au point leurs propres systèmes de transport intelligents pour relever les nouveaux défis liés à la congestion routière.

L'une des composantes de base est le système de comptage des véhicules, qui est conçu pour recueillir des renseignements sur la circulation. Dans ce mémoire, nous nous concentrons sur le développement d'un système de détection et de comptage des véhicules. Les exigences de base du nouveau système sont les suivantes :

1. sa précision de détection et de comptage doit être supérieure à 90%
2. il doit être robuste dans diverses conditions météorologiques et de lumière, et il doit bien fonctionner la nuit.
3. il doit détecter et compter les véhicules à n'importe quel moment et sur n'importe quel site.
4. l'entretien du système ne devrait pas consommer des ressources considérables. [1]

I.2 But et Objectifs

L'objectif de ce travail est de proposer une méthode pour la détection et comptage des véhicules. Afin d'atteindre notre objectif principal, d'autres sous objectifs spécifiques ont étés spécifiés :

- Élaborer des lignes directrices sur les dénombrements automatiques aux différents sites à l'aide de séquences vidéo préenregistrées.
- Élaborer une méthode automatisée de comptage des arrivées et des départs de véhicules à des endroits particuliers.
- Comparer l'exactitude de la méthode automatisée avec les comptages manuels. [3]

I.3 Structure du mémoire

Ce mémoire s'organise en quatre chapitres :
Dans le premier chapitre, nous présenterons la détection de véhicule, leur problématique et leur objectifs. Puis, dans le deuxième chapitre, nous verrons les différentes méthodes et techniques utilisées pour mettre en œuvre un système de détection et de comptage des véhicules. Ensuite, dans le troisième chapitre nous allons présenter l'architecture du système de détection et de comptage des véhicules que nous avons proposé, ainsi que les méthodes et techniques utilisées. Dans le quatrième chapitre,l'implémentation des méthodes, les détails matériels, logiciels, paramètres de configuration et description des bases de données que nous avons utilisées. Aussi, nous présenterons en détails les résultats obtenus avec discutions. Enfin, nous conclurons notre ce mémoire, par une synthèse des résultats et mise en perspectives du travail réalisé au cours de ce projet.

Chapitre II :

Revue litterature

Chapitre II

Revue litterature

II.1 Introduction

La recension des écrits comprend deux sections : une section de comptage manuel et une section de comptage automatisé basée sur la programmation.

Dans ce chapitre, nous présentons ce qu'est un système de comptage automatique, pourquoi nous l'utilisons, les différents types de ce système de comptage et nous verrons également son architecture générale et les différentes étapes le constituent.

II.2 Apprentissage automatique pour la détection d'objets

En général, la détection d'objets est une tâche de vision par ordinateur qui consiste à identifier la présence, l'emplacement et le type d'un ou de plusieurs objets prévus dans une image de test donnée. C'est un problème difficile qui se compose de trois processus principaux, à savoir la reconnaissance des objets, la localisation et la classification. Au cours des dernières années, des techniques d'apprentissage profond ont été appliquées à de nombreux problèmes de détection de véhicules et montrent des résultats prometteurs, comme sur les ensembles de données de référence standard et sur les défis de la vision par ordinateur.

Chaque élément de la carte de fonctionnalités a plusieurs boîtes par défaut qui lui sont associées. Les ensembles de cartes de caractéristiques provenaient de différentes couches du réseau CNN. Un réseau CNN typique réduit progressivement la taille de la carte des caractéristiques et augmente la profondeur à mesure qu'il se déplace vers les couches plus profondes. Les couches profondes couvrent des champs réceptifs plus grands et construisent une représentation plus abstraite, tandis que les couches peu profondes ne couvrent que des champs réceptifs plus petits.[2]

II.2.1 Problème de la reconnaissance d'objets

La reconnaissance automatique d'objets dans les images est un des problèmes les plus difficiles en vision par ordinateur.

Le problème de la reconnaissance d'objets est de décider si un objet spécifique ou un objet appartenant à une catégorie d'objets prédéfinis existe dans une image. On peut poser le problème de la reconnaissance comme un appariement entre le modèle de l'objet cible et un ensemble de descripteurs qu'on extrait à partir de l'image de test. Cette généralisation, aussi simple qu'elle semble être, est capable d'expliquer la multitude d'approches qui se différencient par le choix des descripteurs qui représentent l'objet, le type et la complexité du modèle, et les méthodes utilisées pour l'apprentissage du modèle d'objet et l'appariement.

II.2.2 Pourquoi compter les véhicules ?

La vision par ordinateur est un domaine interdisciplinaire qui consiste à donner aux ordinateurs la capacité de «voir» ou de comprendre le contenu d'images numériques telles que des photos et des vidéos. Si la vision est une tâche insignifiante pour les humains et les animaux, elle est actuellement assez difficile pour les machines. Cependant, beaucoup de progrès ont été réalisés dans le domaine au cours des dernières décennies et de nouvelles techniques et technologies pour rendre le CV plus rapide et plus précis sont activement recherchées.

La vision par ordinateur (CV) faisait partie de liste de choses à apprendre depuis longtemps, donc décidé de profiter de mon projet final pour l'apprendre. En temps réel qui utilisait une bonne dose d'IA, mais savais que n'aurais pas pu le terminer à temps pour ma défense, alors pensé qu'un projet de CV était le moyen de le faire. Aller car utiliserais très probablement le Machine Learning (ML).

II.2.3 Sélection des caméras

Les caméras ont été sélectionnées en fonction de la résolution, de la durée de vie de la batterie et des conditions météorologiques. La résolution d'une caméra est un facteur important qui contrôle le succès du comptage automatisé. L'un des objectifs de cette recherche était d'utiliser des vidéos préenregistrées pour le comptage automatisé.

II.3 Les différents méthodes de comptage de véhicule

II.3.1 Manuel comptant

Dans les premiers jours avant la montée de l'apprentissage automatique, le processus de comptage des véhicules a été effectué manuellement. Il a été effectué par une personne debout au bord de la route ; à l'aide d'un dispositif électronique pour enregistrer les données à l'aide d'une feuille de pointage. Dans certains cas, la personne peut faire le comptage en observant les séquences vidéo capturées par les caméras de la ville ou la télévision en circuit fermé (CCTV) placée au-dessus de la route ou de l'autoroute. Selon une étude réalisée en [1], les performances de calcul manuel du véhicule sont

exactes à 99. Cette enquête est basée sur le calcul manuel de divers véhicules à partir d'un enregistrement vidéo de 5 minutes. Bien que la méthode manuelle offre une grande précision, elle nécessite une quantité importante de ressources humaines.

En outre, il a tendance à être sujette à l'erreur, en particulier sur la circulation intense et de multiples lignes routières. Par conséquent, les calculs manuels sont habituellement effectués avec seulement un petit échantillon de données, et les résultats sont extrapolés pour l'ensemble de l'année ou de la saison pour les prévisions à long terme. [3]

il existe principalement deux méthodes de comptage manuel dans l'ère actuelle : le comptage du trafic sur site et le comptage de la vidéo préenregistrée. Les chiffres de la circulation sur le site font référence au comptage de la circulation sur le site par des personnes formées. Le nombre de personnes nécessaires au comptage peut varier selon le nombre de voies, le volume de circulation, la vitesse des véhicules et les conditions météorologiques. Le comptage du trafic sur site fournit des données de comptage immédiates, car les personnes formées déclarent les comptages en temps réel. Le comptage manuel à partir de la vidéo préenregistrée fait référence au comptage du trafic à partir des séquences vidéo du site. Dans cette méthode, des caméras sont installées sur un site pour enregistrer le mouvement de la circulation.

Après l'enregistrement, les vidéos sont téléchargées à partir des caméras et stocké sur un serveur. Enfin, le comptage du trafic est effectué à partir des vidéos stockées par des personnes formées.

Le comptage manuel à partir d'une vidéo préenregistrée fonctionne mieux que le comptage sur site en termes de comptage répété, de sécurité et de conditions météorologiques défavorables. Le comptage manuel à partir de la vidéo préenregistrée peut être vérifié à plusieurs reprises pour trouver l'erreur dans les comptages, alors que le comptage sur site ne fournit pas cette possibilité.[3]

II.3.2 Dénombrement des vehicles

La méthode de comptage est basée sur les marques de la boîte de délimitation régionale du véhicule et sur la ligne de référence virtuelle. Cette technique suppose que le véhicule se déplace dans une direction. Pour le comptage, chaque véhicule détecté dans l'étape de détection est attribué avec une étiquette unique et suivi jusqu'à ce qu'il atteigne la ligne virtuelle. Dans ce travail, nous avons utilisé cinq étiquettes de classe différentes, à savoir vélo, voiture, moto, bus et camion. Et toutes ces étiquettes sont catégorisées comme objet de véhicule et seront utilisées dans le système de comptage.

Il existe différents modèles traditionnels pour l'application d'algorithmes informatiques pour compter le trafic. La procédure de base de tous les modèles est l'application de la technique de traitement d'image (Joseph 2018). La méthode traditionnelle de soustraction d'arrière-plan (Stauffer 2009) et la méthode séquentielle Monte Carlo (Li 2002) sont des méthodes courantes pour le traitement d'images. La première étape de ces modèles consiste à détecter des objets dans chaque cadre d'une vidéo. Différents modèles utilisent différentes techniques de détection d'objets ; par exemple, la méthode de soustraction

d'arrière-plan sépare les objets en mouvement de l'arrière-plan d'une image en appliquant la technique de changement de pixel. L'étape suivante consiste à suivre les objets détectés d'une image à l'autre à l'aide de différents trackers tels que le tracker Kalman box. La plupart de ces modèles traditionnels de détection d'objets sont lents, ne peuvent pas classer les véhicules et ont de faibles niveaux de précision. Cependant, récemment, quelques modèles de détection d'objets en réseau neuronal préformés ont été développés, qui sont plus efficaces que les méthodes traditionnelles pour le comptage des véhicules.

Actuellement, il existe trois modèles de détection d'objets modernes populaires disponibles : les modèles Faster R-CNN, SSD et YOLO. Ils sont préformés pour détecter quelques objets par défaut. Ce modèle peuvent être exploités sur la plateforme OpenCV (une bibliothèque pour la mise en œuvre d'algorithmes de vision informatique). N'importe quel langage de programmation peut être utilisé à l'interface désirée des modèles, mais Python est un langage de programmation le plus populaire utilisée dans l'analyse de vision par ordinateur.

II.3.3 Comptage automatisé

II.3.3.1 Recherche sur le comptage automatisé

L'une des méthodes classiques de détection d'objets consiste à appliquer la méthode de soustraction d'arrière-plan, qui consiste à séparer la partie mobile de l'image de l'ensemble du cadre analysé. La méthode de soustraction d'arrière-plan peut être appliquée de différentes manières en fonction de la modélisation de chaque pixel de l'image. Par exemple, Ridder et al. (1995) ont modélisé chaque pixel de l'arrière-plan en appliquant un filtre Kalman pour identifier quels pixels appartiennent à l'arrière-plan et lesquels ne le font pas, tandis que Wren (1997) modélisé l'arrière-plan en utilisant une seule valeur gaussienne pour estimer la probabilité qu'un pixel appartienne ou non à l'arrière-plan. [3]

Cependant, ces types initiaux de méthodes de soustraction d'arrière-plan ont de nombreuses limites, comme le fait de ne pas être robuste lorsque les séquences vidéo contiennent des ombres, une faible luminosité, des changements de lumière et des véhicules lents. [3]

En outre, la vitesse de traitement vidéo de la plupart de ces méthodes est trop lente pour le comptage pratique des véhicules. Pour surmonter ces problèmes, Stauffer et Grimson (1999) ont modélisé chaque pixel comme un mélange de valeurs gaussiennes, c'est-à-dire que chaque pixel est modélisé comme un mélange de valeurs. L'étape initiale de leur méthode est de diviser chaque image en plusieurs pixels. L'étape suivante est de modéliser chaque pixel de l'image en suivant quelques règles. Si la source d'un pixel est une seule surface d'éclairage, un seul Gaussien est utilisé pour représenter le pixel. Pratiquement, dans la vue frustum (c.-à-d., l'image capturée par la caméra) de un pixel particulier, plusieurs surfaces peuvent apparaître, et les conditions d'éclairage sur la surface peuvent changer. Dans ce cas, ils ont utilisé un mélange de Gaussiens dits adaptatifs pour approcher ce processus. Les gaussiens adaptatifs sont des valeurs qui peuvent changer avec les changements dans la condition d'un pixel. [3]

Le pixel d'un cadre vidéo est modélisé de cette manière. Dans l'image suivante d'une vidéo, les valeurs gaussiennes pour les pixels qui ne correspondent pas aux valeurs de pixels de l'image précédente sont regroupées à l'aide de composants connectés.

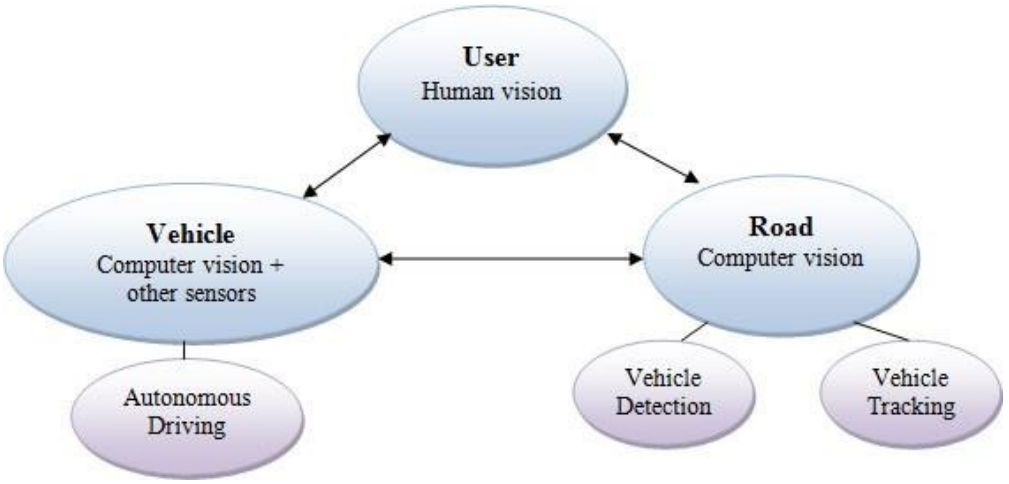


Figure II.1 – Composants du système de transport intelligent basé sur la vision.

II.3.4 Architecture du système UAV

Ce cadre est basé sur la plateforme de l'UAV. Dans la Figure II.1, le flux vidéo est capturé par une caméra montée sur l'UAV. Le détecteur peut traiter deux situations (arrière-plan statique et arrière-plan mobile). Nous distinguons si l'arrière-plan se déplace ou non selon le mode de mouvement de l'UAV. Pour l'arrière-plan statique, un algorithme basé sur des échantillons pour la soustraction de l'arrière-plan est employé dans notre cadre, qui peut détecter le véhicule en mouvement par la modélisation de l'arrière-plan et peut mettre à jour le modèle d'arrière-plan en continu.

En mettant à jour le modèle de fond, les paramètres du modèle sont plus appropriés pour décrire la scène en temps réel. Pour déplacer l'arrière-plan, les platesformes de caméra se déplacent avec un drone. Dans ce cas, l'enregistrement d'images est utilisé dans notre cadre pour transformer les coordonnées de la caméra des images adjacentes en un système de coordonnées de référence. Ainsi, le mouvement de la caméra peut être compensé dans les images adjacentes, afin que nous puissions détecter les véhicules à partir du cadre de référence. [4]

Les images capturées par UAV sont caractérisées par un arrière-plan complexe et une forme variable du véhicule, ce qui entraîne une discontinuité du détecteur et influe ainsi sur la précision du comptage des véhicules. Ainsi, pour résoudre ce problème, un outil de suivi de l'apprentissage en ligne est utilisé dans notre cadre, qui permet de mettre à jour les échantillons utilisés pour la formation. En outre, étant donné que le tracker traditionnel ne peut suivre qu'un seul objet, nous concevons un module de gestion multi-objet efficace en utilisant la technologie multithreading, qui peut assigner multitâche de suivi des objets à des blocs parallèles et peut analyser et valider l'état des véhicules suivis. Enfin, l'état des véhicules chenillés sert à compter le nombre de véhicules.[4]

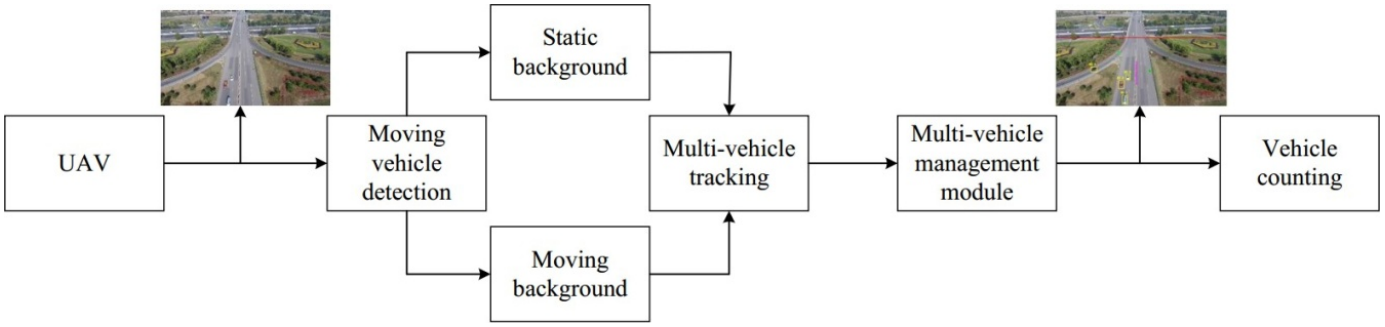


Figure II.2 – Cadre de la méthode proposée

II.3.5 Détection de véhicule avec YOLOv3

Cette section décrit les méthodes de détection d'objets utilisées dans cette étude. La mise en œuvre du cadre de détection des véhicules routiers a utilisé le réseau YOLOv3. L'algorithme YOLOv3 poursuit l'idée de base des deux premières générations d'algorithmes YOLO. Le réseau neuronal convolutionnel est utilisé pour extraire les caractéristiques de l'image d'entrée. Selon la taille de la carte de caractéristiques, telle que 13*13, l'image d'entrée est divisée en 13*13 grilles. Le centre de la zone d'étiquette d'objet est dans une unité de grille et l'unité de grille est responsable de la prédiction de l'objet. La structure de réseau adoptée par YOLOv3 est appelée Darknet-53. [5]

Cette structure adopte la pleine convolution et remplace la version précédente du réseau de neurones convolutifs à connexion directe par la structure résiduelle. La branche est utilisée pour connecter directement l'entrée à la couche profonde du réseau. L'apprentissage direct des résidus garantit l'intégrité des informations sur les caractéristiques de l'image, simplifie la complexité de l'apprentissage et améliore la précision de détection globale du réseau. [5]

Dans YOLOv3, chaque unité de grille aura trois boîtes englobantes d'échelles différentes pour un objet. La boîte candidate qui a la plus grande zone de chevauchement avec la boîte annotée sera le résultat final de la prédiction. De plus, le réseau YOLOv3 a trois échelles de sortie, et les trois branches d'échelle sont finalement fusionnées. Les entités peu profondes sont utilisées pour détecter les petits objets, et les entités profondes sont utilisées pour détecter les grands objets ; le réseau peut ainsi détecter des objets avec des changements d'échelle. La vitesse de détection est rapide et la précision de détection est élevée. Les scènes de trafic prises par la vidéo de surveillance des autoroutes ont une bonne adaptabilité au réseau YOLOv3.[5]

Le réseau affichera enfin les coordonnées, la confiance et la catégorie de l'objet Lorsque vous utilisez la détection YOLO, les images sont redimensionnées à la même taille, par exemple 416 x 416, lorsqu'elles sont envoyées au réseau. Puisque l'image est segmentée, la taille de la surface de la route distante se déforme et devient plus grande.

Par conséquent, plus de points caractéristiques d'un petit objet véhicule peuvent être acquis pour éviter la perte de certaines caractéristiques d'objet en raison du fait que

l'objet véhicule est trop petit. L'ensemble de données présenté dans la section «Jeu de données véhicule» est placé dans le réseau YOLOv3 pour la formation, et le modèle de détection d'objet véhicule est obtenu. Le modèle de détection d'objets de véhicule peut détecter trois types de véhicules : les voitures, les bus et les camions.[5]

Parce qu'il y a peu de motos sur l'autoroute, ils n'ont pas été inclus dans notre détection. La zone éloignée et la zone proximale de la chaussée sont envoyées au réseau pour détection. Les positions détectées de la boîte de véhicule des deux zones sont reconstituées à l'image d'origine, et la position d'objet correcte est obtenue dans l'image d'origine. L'utilisation du procédé de détection d'objet de véhicule pour obtenir la catégorie et l'emplacement du véhicule peut fournir les données nécessaires pour le suivi d'objet. Les informations ci-dessus sont suffisantes pour le comptage de véhicules, et le procédé de détection de véhicule ne détecte donc pas les caractéristiques spécifiques du véhicule ou l'état du véhicule.[5]

II.3.5.1 La méthode YOLO

You Only Look Once (YOLO) est un modèle de détection d'objets préformé récemment développé. C'est le modèle le plus efficace en termes de vitesse et de précision, et puisque la vitesse et la précision sont les facteurs les plus critiques, YOLO est le modèle le plus approprié pour compter les véhicules à partir de vidéos préenregistrées. De plus, les planificateurs des transports exigent le dénombrement directionnel des véhicules, ce qui est une exigence supplémentaire qui va au-delà de la simple reconnaissance des véhicules. Le comptage dans les intervalles de temps est également un facteur à considérer, car de nombreux projets nécessitent des données sur le trafic pour différents intervalles de temps, par exemple, à des intervalles de 5 ou 15 minutes. L'application de ces caractéristiques à la capacité de comptage de l'algorithme de détection d'objets YOLO .

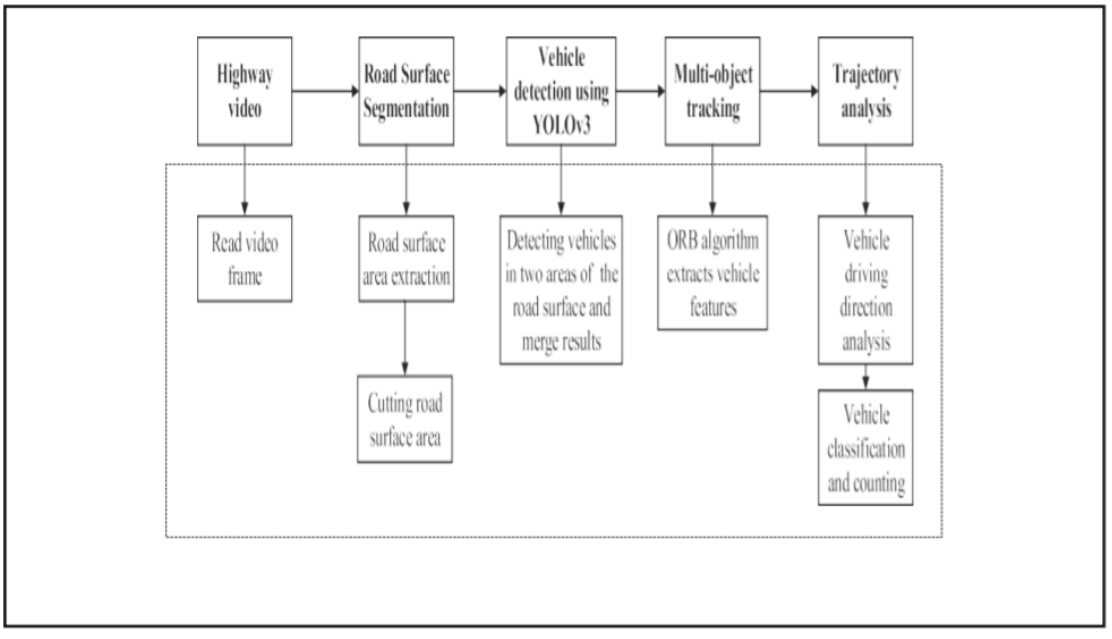


Figure II.3 – Flux global de la méthodeYOLO

II.3.5.2 YOLOv3

YOLOv3 est extrêmement rapide et précis. YOLOv3 est sur le même pied que la perte focale, mais environ 4x plus rapide. En outre, vous pouvez facilement faire un compromis entre vitesse et précision simplement en changeant la taille du mode.[6]

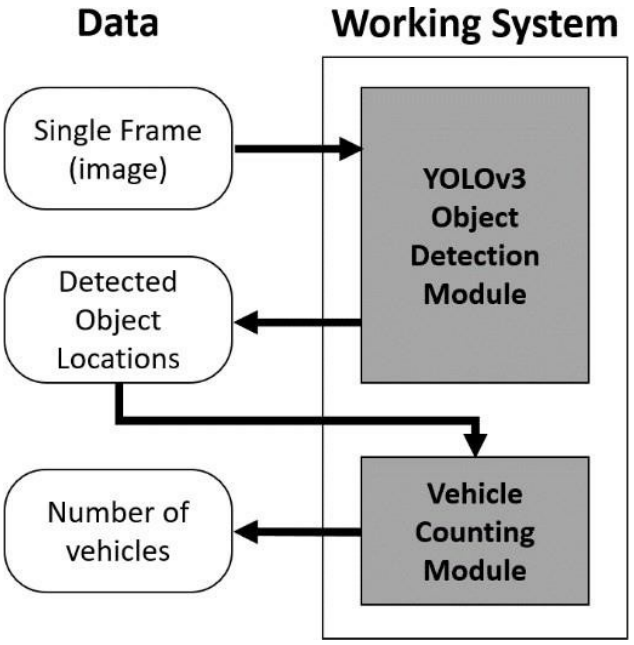


Figure II.4 – Architecture générale du système de comptage des véhicules

II.3.6 Méthode de comptage de véhicules utilisant le Mask - Rcnn

Dans cet article, une approche précise pour le comptage de véhicules dans des vidéos utilisant Mask R-CNN et KLT tracker est proposée. La détection de véhicule est effectuée pour chaque N trames à l'aide du modèle de segmentation d'instance Mask R-CNN. Ce modèle surpasse les autres modèles d'apprentissage en profondeur qui, en utilisant la détection de la boîte englobant, car il fournit un masque de segmentation pour chaque objet détecté, la surperformance apparaît clairement en cas d'occlusion. Une fois les objets détectés, leurs points d'angle sont extraits et suivis. Une méthode efficace est introduite pour attribuer des trajectoires ponctuelles à leurs véhicules détectés correspondants. L'algorithme de comptage proposé distingue précisément les véhicules neufs des véhicules comptés. Les expériences réalisées sur diverses vidéos difficiles montrent d'excellents résultats par rapport aux méthodes de comptage de pointe.[15]



Figure II.5 – la détection de véhicule par mask-RCNN

II.3.7 DÉTECTION D'OBJETS BASÉE SUR CNN

Pour la détection d'objets et la classification dans nos 7 classes d'annotations, nous ajustons le modèle YOLO CNN, qui a été connu pour donner une grande précision de détection d'objets à faible latence d'inférence. Dans YOLO, un seul réseau neuronal est appliqué à l'image entière. Le réseau divise l'image en zones de grille rectangulaires et prédit les cases limitatives et les probabilités pour chaque zone. Les détections sont menacées par une certaine valeur de probabilité pour ne voir que les détections à cote élevée (ce processus est connu sous le nom de suppression non maximale dans la communauté de la vision par ordinateur). [7]

Nous utilisons un modèle de YOLO pré-formé sur l'ensemble de données MS-COCO. Nous le peaufinons sur les ensembles de données PASCAL VOC 2007 et KITTI. Cette étape est suivie d'une mise au point sur nos propres ensembles de données annotées personnalisées (tableau 1, ligne 4). Nous ajustons 5 modèles YOLO différents, un modèle utilisant exclusivement les données d'entraînement d'une seule installation de caméra. Pour quatre installations, cela nous donne quatre modèles YOLO. Nous ajustons un cinquième modèle, en combinant les données de formation de toutes les installations. Ces modèles seront appelés YOLO1, YOLO2, YOLO3, YOLO4 et YOLO5 dans les sections suivantes. Comme chaque installation fournit des images vidéo de différentes résolutions, ces images sont redimensionnées à 416 X 416 avant d'être introduites dans le modèle pour le réglage et l'inférence.

Pour la mise au point, nous utilisons un poste de travail Dell Precision Tower 5810, équipé d'une carte graphique NVIDIA Quadro P5000. Pour chaque modèle (sauf YOLO5), les poids de la 10600e l'époque de réglage fin est utilisée pour évaluer la précision de l'inférence sur les données d'essai (tableau 1, ligne 5). Pour YOLO5, après un réglage fin de plus de 80200 itérations, les meilleures précisions sont obtenues à l'aide des poids de la 720006e itération. Chaque période de formation prend environ une seconde sur cette station de travail, et par conséquent le processus de réglage fin pour les quatre premiers modèles prend 3 heures et que pour YOLO5 prend 20 heures.[7]

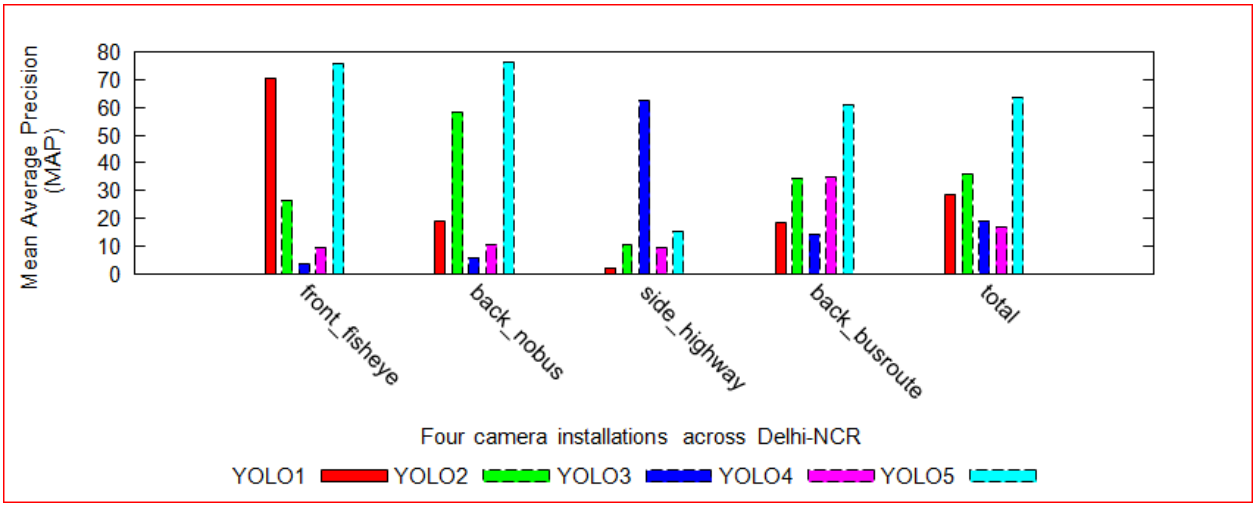


Figure II.6 – Moyenne de précision (MAP) des cinq modèles de détection d'objets sur les cinq ensembles d'essai (un de chacune des quatre installations de caméra et un cinquième ensemble d'essai combiné [7].

II.4 les réseaux neuronaux convolutifs (CNN)

L'utilisation de réseaux convolutionnels profonds (CNN) a connu un succès incroyable dans le domaine de la détection d'objets de véhicules. Les CNN ont une forte capacité à apprendre les faits saillants de l'image et peuvent effectuer de multiples tâches connexes, comme la classification et la régression des cases limitatives [7]. La méthode de la détection peut généralement être divisée en deux catégories. La méthode en deux étapes génère une boîte candidate de l'objet via différents algorithmes et classe ensuite l'objet par un réseau neuronal convolutionnel. La méthode à une étape ne génère pas de boîte candidate mais convertit directement le problème de positionnement de la boîte de délimitation de l'objet en un problème de régression à traiter.

II.4.1 Que sont les réseaux neuronaux convolutifs ?

II.4.1.1 Introduction

Réseaux neuronaux convolutionnels. Cela ressemble à une combinaison bizarre de biologie et de mathématiques avec un peu de CS, mais ces réseaux ont été quelques-unes des innovations les plus influentes dans le domaine de la vision par ordinateur. 2012 a été la première année où les réseaux neuronaux ont pris de l'ampleur, Alex Krizhevsky s'en étant servi pour remporter la compétition ImageNet de cette année-là (essentiellement, les Jeux olympiques annuels de la vision par ordinateur). Depuis lors, une multitude d'entreprises utilisent le deep learning au cœur de leurs services.

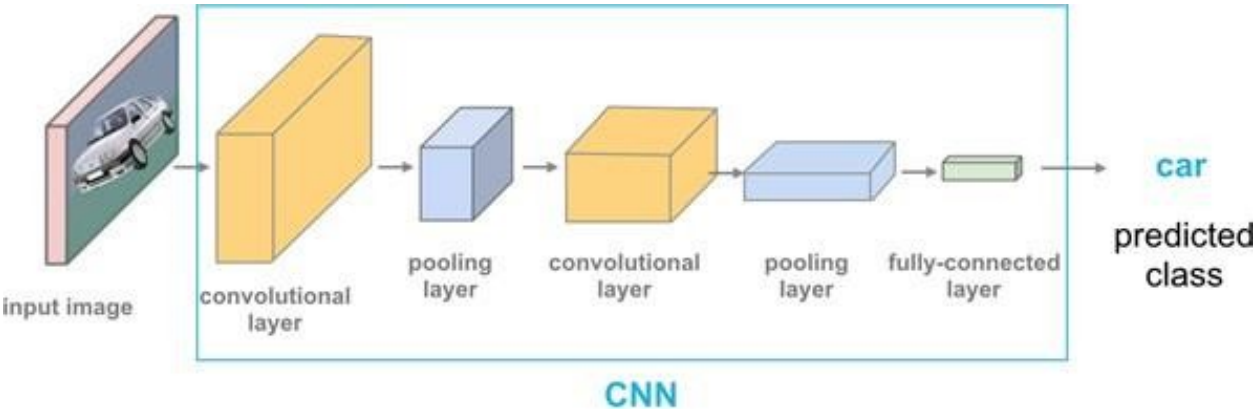


Figure II.7 – Une vue générale de CNN.

II.4.2 input et output

Lorsqu'un ordinateur voit une image (prend une image en entrée), il voit un tableau de valeurs de pixels. En fonction de la résolution et de la taille de l'image, il verra un tableau de nombres 32 x 32 x 3 (Le 3 se réfère aux valeurs RVB). Juste pour faire comprendre le point, disons que nous avons une image couleur sous forme de JPG et sa taille est 480 x 480. La matrice représentative sera 480 x 480 x 3. Chacun de ces nombres est donné une valeur de 0 à 255 qui décrit l'intensité des pixels à ce point.

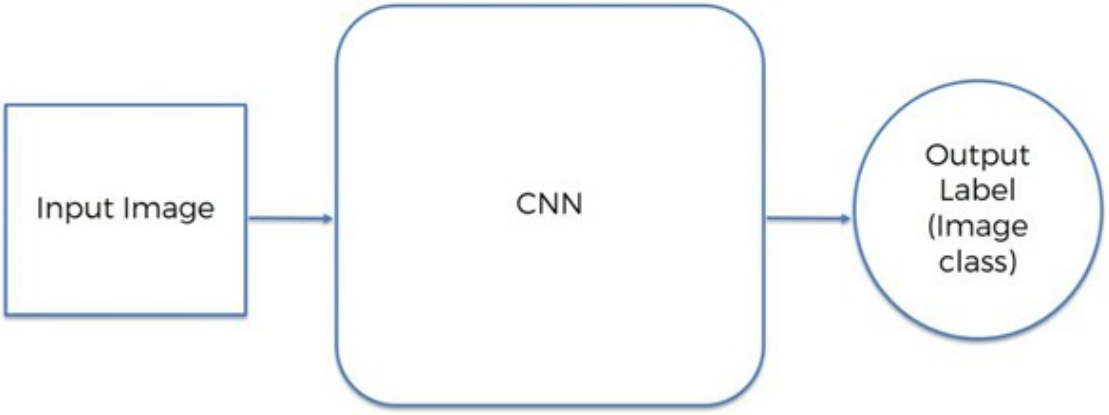


Figure II.8 – Cnn image d'entrée et sortie image

II.4.3 Structure du réseau neuronal convolutif

Un aperçu plus détaillé de ce que font les CNN serait de prendre l'image, de la passer à travers une série de couches convolutives, non linéaires, groupées (sous-échantillonnage), et entièrement connectées, et d'obtenir une sortie. Comme nous l'avons dit précédemment, lasortie peut être une classe unique ou une probabilité de classes qui décrit le mieux l'image. Maintenant, la partie dure comprend ce que chacune de ces couches fait. Alors entrons dans la partie la plus importante.

Le processus de construction d'un CNN comporte quatre grandes étapes

1. Convolution.
2. Mise en commun.
3. Aplatissement.
4. Connexion complète.



Figure II.9 – Quatre étapes de Cnn.

II.4.3.1 Convolution

ConvNets dérive son nom de l'« Opération de Convolution ». La Convolution dans le cas de ConvNet est d'extraire des fonctionnalités des images d'entrée. La convolution préserve les relations spatiales entre les pixels en apprenant les caractéristiques de l'image à l'aide du petit carré de données d'entrée. Comme nous l'avons vu plus haut, chaque image peut être considérée comme une matrice de la valeur des pixels. Considérez une image 5 x 5 dont les valeurs de pixel sont seulement 0 et 1 (notez que pour une image en niveaux de gris, les valeurs de pixel vont de 0 à 255, où les valeurs de pixel sont seulement 0 et 1.

Terminologie de CNN, la matrice 3 x 3 est appelée un « filtre » ou un « noyau » ou un « détecteur de caractéristiques » et la matrice formée en faisant glisser le filtre sur l'image et en calculant le produit de point est appelée la « fonction convoquée » ou la « carte d'activation » ou la « carte de caractéristiques ». Il est important de noter que les filtres agissent comme des détecteurs de fonction à partir de l'image d'entrée d'origine.

II.4.3.2 Profondeur

La profondeur correspond au nombre de filtres que nous utilisons pour l'opération de convolution. Dans le réseau illustré dans la figure ci-dessous, nous effectuons la convolution de l'image originale du bateau à l'aide de trois filtres distincts, produisant

ainsi trois cartes de caractéristiques différentes comme indiqué. Vous pouvez penser à ces trois cartes d'entités comme des matrices 2D empilées, de sorte que la « profondeur » de la carte d'entités serait de trois. Plus il y a de filtres, plus le résultat est précis.

II.4.3.3 Stride

La foulée est le nombre de pixels par lesquels nous glissons notre matrice de filtre sur la matrice d'entrée. Lorsque la foulée est de 1, nous déplaçons les filtres un pixel à la fois. Lorsque la foulée est de 2, puis les filtres sautent 2 pixels à la fois que nous les faisons glisser autour. Ayant une plus grande foulée va produire des cartes de caractéristiques plus petites.

II.4.3.4 Zero-Padding

Parfois, il est pratique de tamponner la matrice d'entrée avec des zéros autour de la frontière, afin que nous puissions appliquer le filtre aux éléments limitrophes de notre matrice d'images d'entrée. Une caractéristique intéressante de zéro rembourrage est qu'il nous permet de contrôler la taille des cartes de caractéristiques. Ajouter zéro-padding est également appelé convolution large, et ne pas utiliser zéro-padding serait une convolution étroite.

II.4.3.5 Calque ReLu

Une opération supplémentaire appelée ReLU a été utilisée après chaque opération de convolution. ReLU signifie l'unité linéaire rectifiée et est une opération non linéaire. Sa sortie est donnée par :

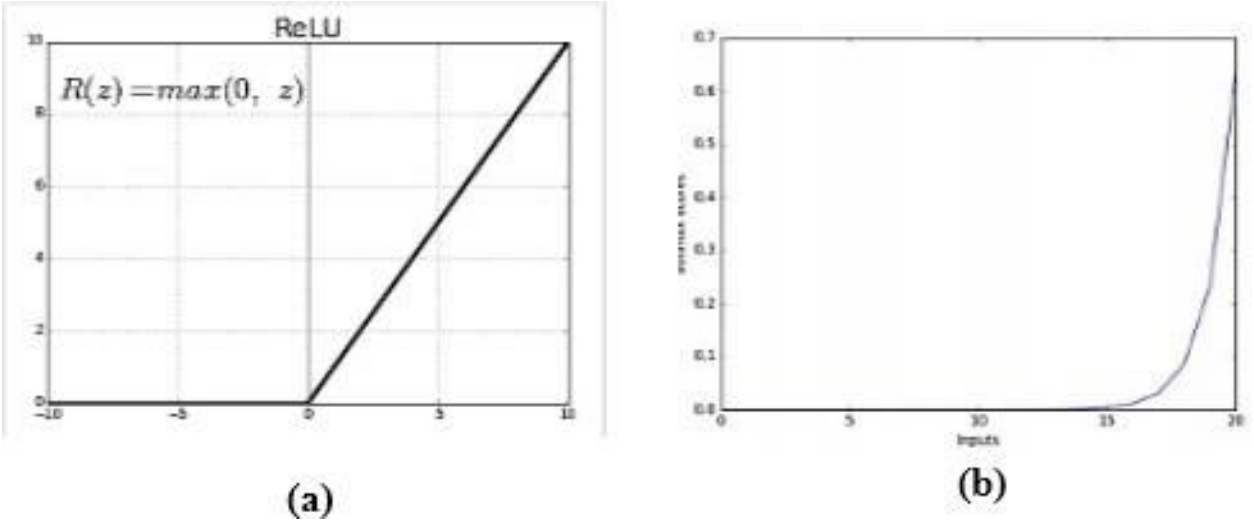


Figure II.10 – Deux fonctions d'activations (a) ReLU et (b) SoftMax

ReLU est une opération par élément (appliquée par pixel) et remplace toutes les valeurs négatives de pixel dans la carte des fonctionnalités par zéro. Le but de ReLU

est d'introduire la non-linéarité dans notre ConvNet puisque la plupart des données du monde réel que nous voudrions que notre ConvNet apprenne seraient non-linéaires (La convolution est une opération linéaire ,multiplication et addition de matrice dans le sens des éléments, de sorte que nous tenons compte de la non-linéarité en introduisant une fonction non linéaire comme ReLU). Comme le montre l'image ci-dessous, nous appliquons l'opération Relu et il remplace tous les nombres négatifs par 0.

II.4.4 Mise en commun maximale

La mise en commun spatiale (également appelée sous-échantillonnage ou souséchantillonnage) réduit la dimensionnalité de chaque carte de caractéristiques, mais conserve les informations les plus importantes. La mise en commun spatiale peut être de différents types : Max, Average, Sum, etc.

Dans le cas de Max Pooling, nous définissons un voisinage spatial (par exemple, une fenêtre 2x2) et prenons le plus grand élément de la carte des caractéristiques rectifiées dans cette fenêtre. Au lieu de prendre le plus grand élément, nous pourrions également prendre la moyenne (Average Pooling) ou la somme de tous les éléments dans cette fenêtre. En pratique, il a été démontré que Max Pooling fonctionne mieux.

Ci-dessous montre un exemple d'opération de Max Pooling sur une carte Rectified Feature (obtenue après opération de convolution + ReLU) en utilisant une fenêtre 2 x 2.

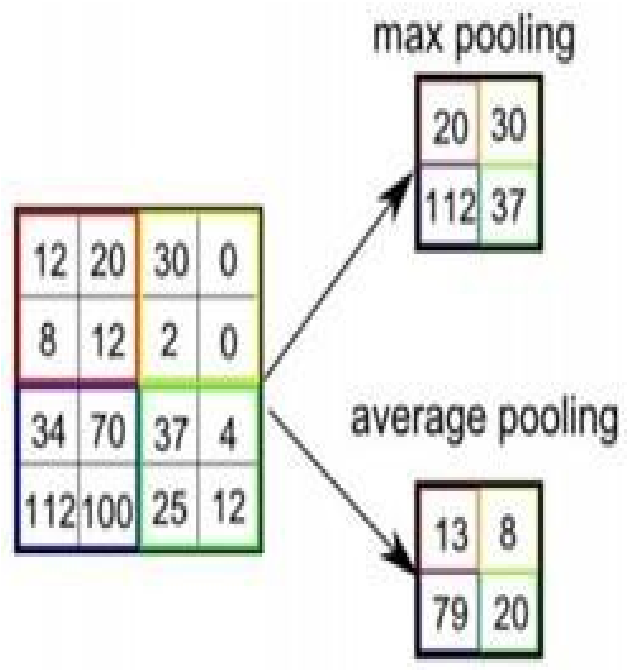


Figure II.11 – La différence entre Average Pooling et Max Pooling.

II.4.4.1 Aplatissement

L'aplatissement est le processus qui consiste à convertir tous les réseaux bidimensionnels résultants en un seul vecteur linéaire long et continu.

II.4.4.2 Connexion complète

La couche Fully Connected est un perceptron multi-couches traditionnel qui utilise une fonction d'activation softmax dans la couche de sortie (d'autres classificateurs comme SVM peuvent également être utilisés, mais colleront à softmax dans ce post). Le terme « entièrement connecté » signifie que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante.

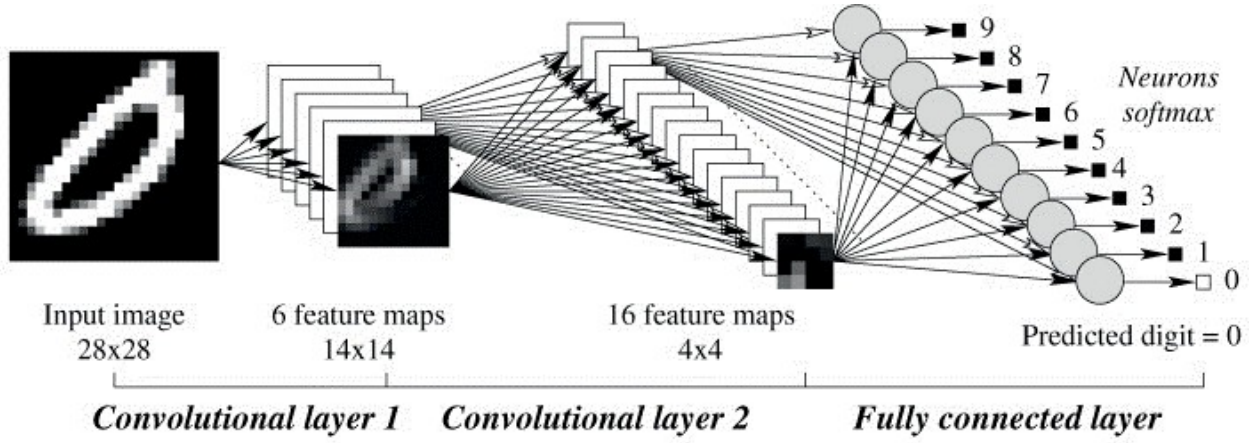


Figure II.12 – Deux couches convolutionnelles suivies d'une couche entièrement connectée.

La sortie des couches convolutionnelles et de mise en commun représente les caractéristiques de haut niveau de l'image d'entrée. L'objectif de la couche Fully Connected est d'utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données de formation. Par exemple, la tâche de classification des images que nous nous sommes fixés d'exécuter comporte quatre sorties possibles, comme le montre la figure ci-dessous (notez que la figure II.12 ne montre pas les connexions entre les nœuds dans la couche entièrement connectée).

Outre la classification, l'ajout d'une couche entièrement connectée est également (habituellement) un moyen bon marché d'apprendre des combinaisons non linéaires de ces caractéristiques. La plupart des caractéristiques des couches de convolution et de mise en commun peuvent être bonnes pour la tâche de classification, mais les combinaisons de ces caractéristiques pourraient être encore meilleures. La somme des probabilités de sortie de la couche entièrement connectée est 1. Ceci est assuré en utilisant le Softmax comme fonction d'activation dans la couche de sortie de la couche entièrement connectée. La fonction Softmax prend un vecteur de scores arbitraires à valeur réelle et l'arrondit à un vecteur de valeurs entre zéro et un qui se résume à un.

II.4.4.3 Rassembler tout cela formation à l'aide de la rétropropagation

Comme nous l'avons vu plus haut, les calques convolution + Pooling agissent comme extracteur de fonctionnalités à partir de l'image d'entrée tout en la couche connectée agit comme classificateur. [8]

II.5 Neural Convolutionnel Classique

II.5.1 Architecteurs de réseau

Nous avons décrit jusqu'à présent les différentes couches de CNN. Maintenant, nous présentons comment ces couches sont combinées pour former l'architecture du réseau. Les architectures CNN les plus simples empilent quelques couches convolutionnelles ensemble, elles suivent la couche de mise en commun, puis ce motif se répète, et on ajoute à la fin les couches entièrement connectées. Les architectures classiques de CNN sont LeNet-5 , AlexNet et VGGNet.

II.5.1.1 LeNet-5

Le LeNet-5 a été le premier CNN, proposé par Yann LeCun et son équipe chez Bell Labs en 1998, cette architecture est montrée à Figure II.10. Ce réseau a été consacré à la reconnaissance des chiffres, LeCun et.al. utilisé le système pour la détection de signature manuscrite dans les contrôles, et il a été déployé avec succès commercialement à cet effet. Il est composé seulement sur quelques couches et quelques filtres, en raison des limitations informatiques à ce moment-là. Comme le montre la Figure II.13, l'architecture a deux couches de convolution, deux couches de pool av- erage, deux couches entièrement connectées et une couche de sortie avec connexion gaussienne. LeNet-5 a 60000 paramètres. Comme fonction d'activation, la fonction d'activation de tanh est utilisée.[9]

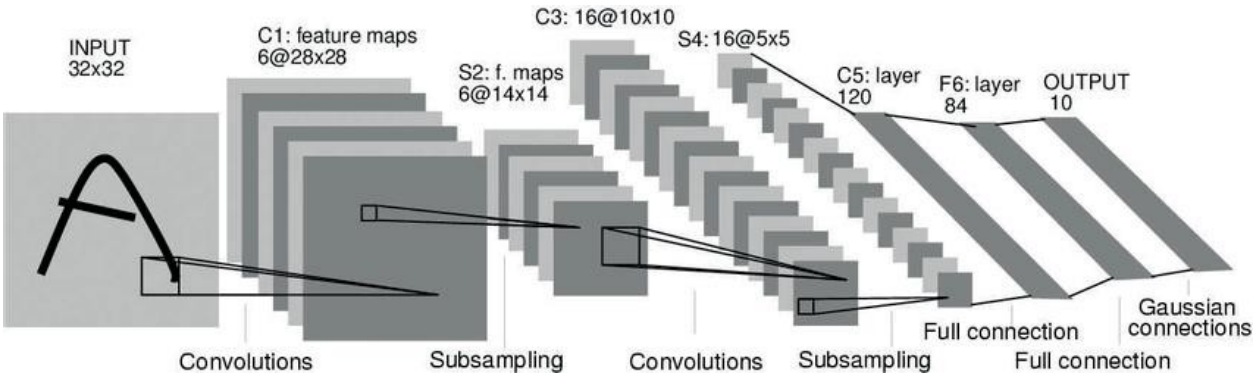


Figure II.13 – Architecture de LeNet-5

II.5.1.2 AlexNet

L'architecture d'AlexNet a été le premier travail à populariser les réseaux convolutionnels dans la vision informatique, il a été le gagnant du concours ImageNet

ILSVRC en 2012, il a eu un 15,4% top-5 taux d'erreur contre 26,2% pour le prochain réseau le plus bas. suit le modèle de l'architecture LeNet-5 , mais elle était plus profonde, plus grande et comportait des couches convolutionnelles empilées les unes sur les autres. La fonction d'activation du réservoir, utilisée dans LeNet-5 . [10]

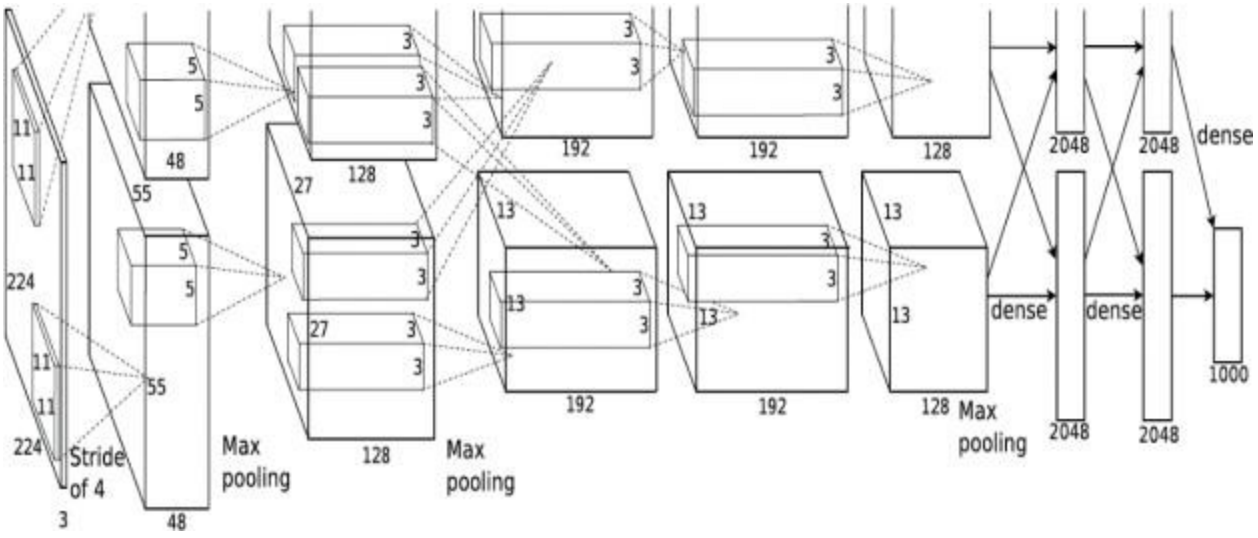


Figure II.14 – Architecture AlexNet.

II.5.1.3 VGGNet

Le réseau VGG est introduit en 2014 par Ka- ren Simonyan et Andrew Zisserman. A cette époque, il était considéré comme un réseau très profond. Sa principale contribution a été de montrer que la profondeur du réseau est un élément essentiel pour améliorer la reconnaissance ou la précision de la détection dans les NC.[11]

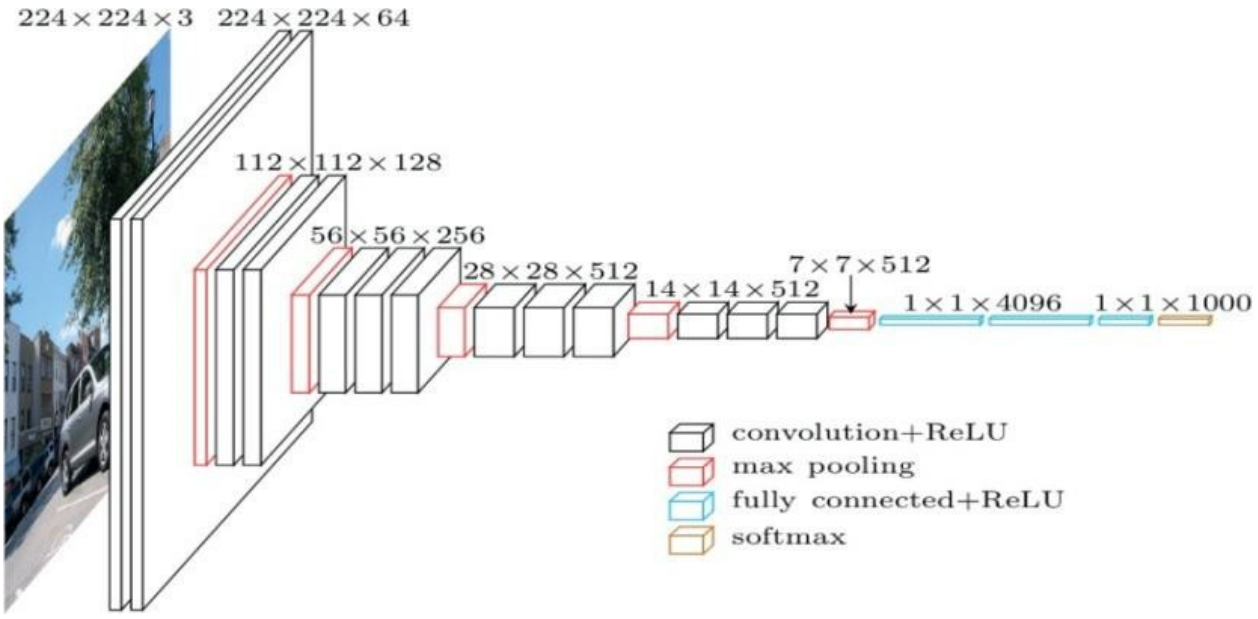


Figure II.15 – Architecture VGGNet.

II.6 Conclusion

Dans ce chapitre, nous avons présenté les méthodes de détection et de comptage des véhicules en général qui nécessite une chaine de traitement qui est constituée de plusieur étapes, nous avons décrit les différents principaux domaines d'application et les défis majeur de la reconnaissance d'object .

Chapitre III :

conceptions du système de détection et comptage

Chapitre III

conceptions du système de détection et comptage

III.1 Introduction

Dans ce chapitre, nous allons présenter notre méthode de détection et comptage a base d'apprentissage .

Le système de détection et comptage de véhicules est composé de trois éléments principaux : un détecteur, un tracker et un compteur. Le détecteur identifie les véhicules dans une trame vidéo donnée et renvoie une liste de cadres de délimitation autour des véhicules au tracker. Le tracker utilise les cadres de délimitation pour suivre les véhicules dans les images suivantes. Le détecteur est également utilisé pour mettre à jour les trackers périodiquement pour s'assurer qu'ils suivent toujours correctement les véhicules.

III.2 Méthodologie

La méthodologie de notre travail est présentée, comparée et déterminée dans ce chapitre. Diverses méthodes de détection et de suivi de véhicules basés sur la vision sont discutées dans littérature actuelle, les exemples de la réalisation de détection et de suivi de véhicules basés sur la vision sont prises pour référence et enfin, les méthodes à adopter dans la thèse sont décidées.

III.2.1 Dataset

On peut dire que les vidéos sont une collection d'un ensemble d'images disposées dans un ordre précis. Ces ensembles d'images sont également appelés cadres (frame).

Il existe plusieurs façons de compter les véhicules sur la route, des comptages manuels aux tubes pneumatiques en passant par les capteurs piézoélectriques. Pourquoi la vidéo a-t-elle été utilisée ? Pourquoi est-il préféré ? Les données des capteurs (séquences vidéo) peuvent être utilisées pour vérifier les résultats du système, ce qui facilite et accélère l'évaluation et l'amélioration du système. Les images peuvent également être utilisées à

d'autres fins, notamment la surveillance, la reconnaissance automatique du numéro de plaque, la détection du type de véhicule et la détection de la vitesse du véhicule pour n'en nommer que quelques-unes. Il est relativement moins cher à mettre en œuvre et à mettre à l'échelle en tant que système de comptage de véhicules permanent par rapport à d'autres systèmes. Il peut suivre et compter plusieurs véhicules se déplaçant dans différentes directions sur plusieurs voies.

On a utilisé deux dataset , une vidéo nommée M6 Motorway Traffic et une vidéo route



Figure III.1 – Data set video M6 Motorway Traffic

Data M6 motorway est caractérise par :

- taille 82.0 mo .
- de type mp4 .
- longueur de video 00 :34 :08 .
- largeur de trame de 1280 .
- hauteur de trame 720 .
- débit de donnes 617 Kbits/s .
- débit total 745 kbits/s .
- fréquence d'image 25 trame par /s .
- source de téléchargement le lien : `https://learnml.today/`.
- ce data est composer de 12870 image .



Figure III.2 – Dataset video route

Data vidéo route .mp4 est caractérise par :

- taille 7.48 Mo.
- de type mp4 .
- longueur de video 00 :00 :57 .
- largeur de trame de 1280 .
- hauteur de trame 720 .
- débit de donnes 972 Kbits/s .
- débit total 1098 kbits/s .
- fréquence d'image 30 trame par /s .
- ce data est composer de 1711 image .

on a extraire et enregistrer des images de notre fichier vidéo Pour construire une dataset de 12870 images avec la suivre des étapes suivantes :

1. ouvrir le fichier vidéo ou camera on utilise cv2.VideoCapture().
2. Lire frame par frame.
3. Sauvegarder l'image utilise : cv2.imwrite().
4. Release le video Capture et destroy all windows.

La méthodologie de cette recherche est abordée : Développement d'un nouvel algorithme informatique pour compter les véhicules à partir de vidéos préenregistrées.

Après on choisit 20 images pour notre data. Lorsque on dit que la fréquence d'une image est 25 frame on choisit les images à partir de 1ere jusqu'a 500 avec un saute de 25 image.

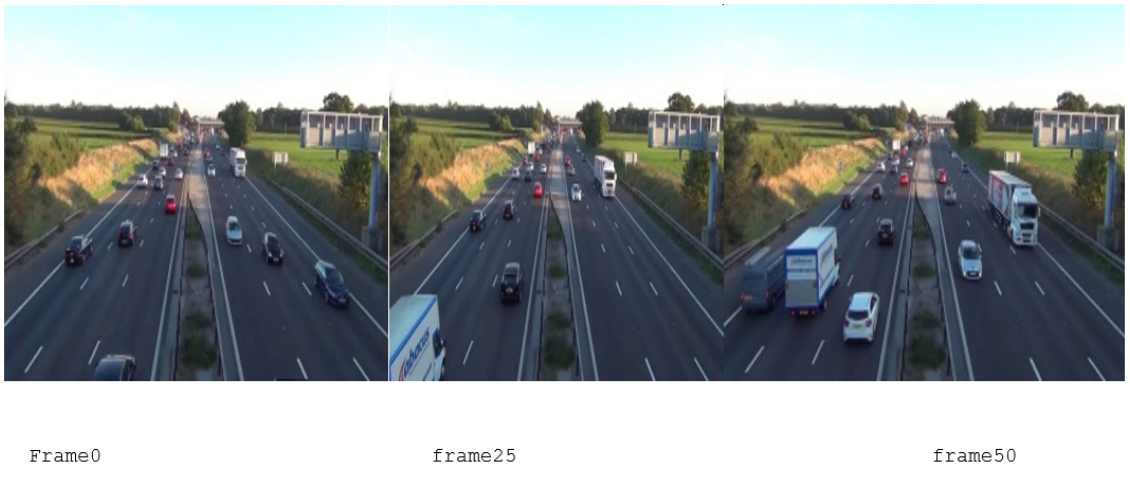


Figure III.3 – extraction et enregistrement d'image vidéo

III.3 Le recadrage (crop) les images

III.3.1 Introduction

Le recadrage (crop) d'une image afin d'obtenir une composition appropriée pour obtenir une meilleure qualité visuelle est notoirement difficile, Principalement en raison de trois faits :

1. déterminer l'objet principal/le thème d'une image donnée est une tâche non triviale, qui nécessite des connaissances approfondies du domaine et des compétences sophistiquées .

2. l'évaluation de l'esthétique de l'image recadrée est hautement subjective, de sorte que différents observateurs peuvent avoir des opinions différentes sur le sujet.

Dans la première partie, nous verrons comment représenter les images OpenCV sous forme de tableaux NumPy. Puisque chaque image est un tableau NumPy, nous pouvons tirer parti du découpage des tableaux NumPy pour recadrer'crop) une image.

Notre objectif dans cette étape de construire un data de 120 image par cop l'image selon les voie la route dans notre cas et lorsque on a deux routes dans vidéo alors on a trois image par route et (6) image de 64*64 multiple par nombre d'images choisi 20*6 = 120 image de taille 64*64 .



Figure III.4 – image de la vidéo extraite de data

III.3.2 Modèle de classification CNN :

La classification vidéo n'est pas si différente d'un problème de classification d'images. Pour une tâche de classification d'images, nous prenons des images, utilisons des extracteurs de caractéristiques (comme des réseaux de neurones convolutifs ou des CNN) pour extraire des caractéristiques des images, puis classons cette image en fonction de ces caractéristiques extraites. La classification des vidéos n'implique qu'une étape supplémentaire. Nous extrayons d'abord les images de la vidéo donnée. Nous pouvons

alors suivre les mêmes étapes que pour une tâche de classification d'images. C'est le moyen le plus simple de traiter les données vidéo. [12]

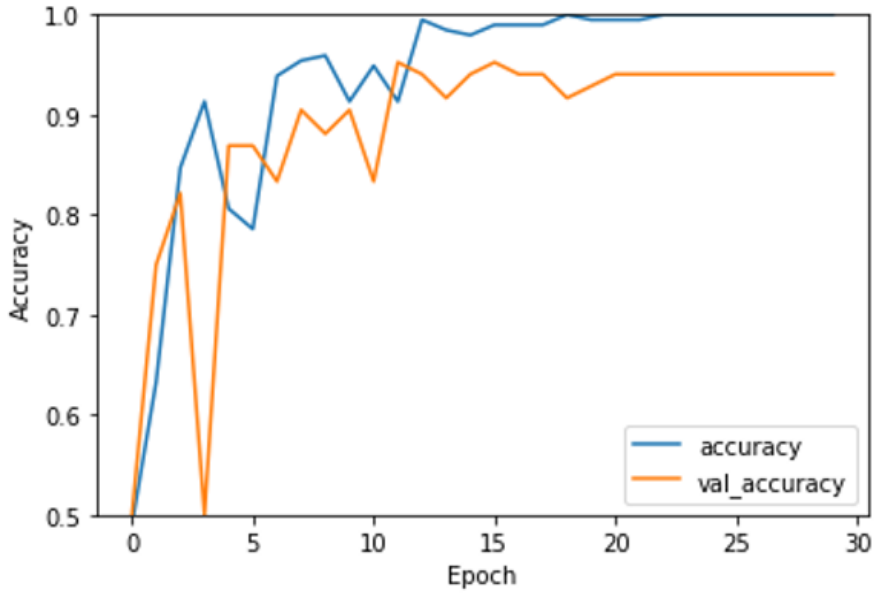
Pour cela, nous utiliserons des CNN pour extraire des caractéristiques des images des vidéos pour prédit si un image contient une voiture ou bien non on passe les 120 image précédentes dans notre simple model suivant :

- Nous avons construit cette architecture en utilisant une seule couche convolutive suivie d'une couche de polling .puis une couche flatten et une couche fully_connected suivie d'un classificateur softmax.

- Etant donné que les dimensions d'entrée et de sortie de la base de données sont respectivement 64*64 avec 2 classes .

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 64)        36928
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
_________________________________________________________________
dense (Dense)                (None, 64)                589888
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 646,858
Trainable params: 646,858
Non-trainable params: 0
```

Figure III.5 – Model summary




```
Epoch 30/30
196/196 [==============================] - 21s 105ms/sample - loss: 0.0013 - accuracy:
1.0000 - val_loss: 0.2832 - val_accuracy: 0.9405
```

Figure III.6 – Model accuracy

III.3.2.1 Transfer Learning

Le principe de base de l'apprentissage par transfert est simple : prendre un modèle formé sur un grand ensemble de données et transférer ses connaissances à un ensemble de données plus petit. Pour la reconnaissance d'objets avec un CNN, nous gelons les premières couches convolutionnelles du réseau et n'entraînons que les dernières couches qui font une prédiction. L'idée est que les couches convolutionnelles extraient des caractéristiques générales de bas niveau applicables à toutes les images - telles que les bords, les motifs, les gradients - et que les dernières couches identifient des caractéristiques spécifiques dans une image. [13]

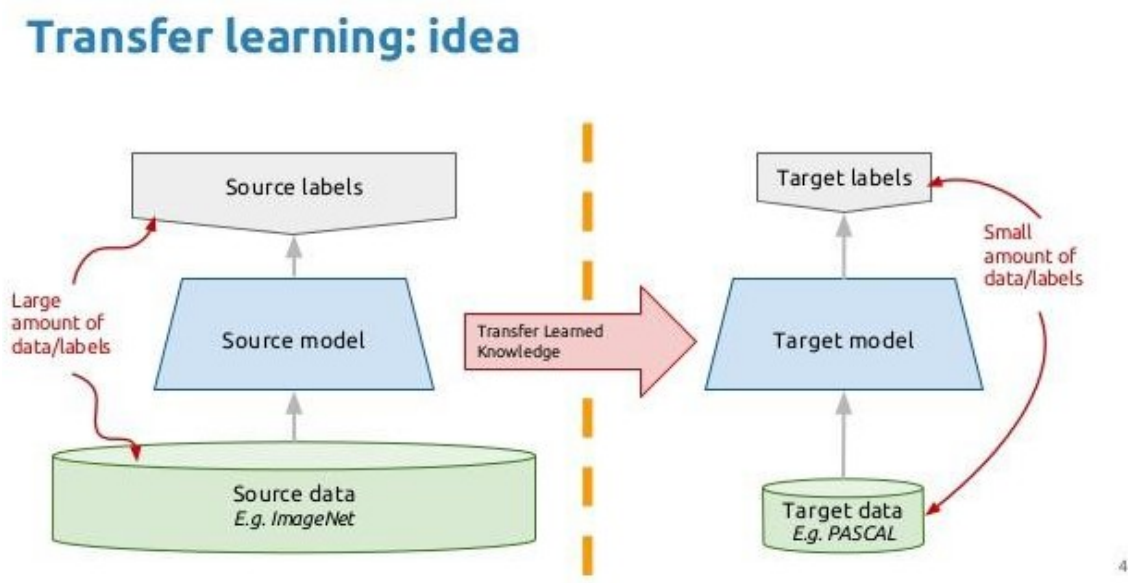


Figure III.7 – idée de transfert d'apprentissage

Voici le schéma général de l'apprentissage par transfert pour la reconnaissance d'objets :

1. Chargement d'un modèle CNN pré-entraîné sur un grand ensemble de données.

2. Geler les paramètres (poids) dans les couches convolutionnelles inférieures du modèle.

3. Ajouter un classificateur personnalisé avec plusieurs couches de paramètres entraînables au modèle.

4. Entraîner les couches du classificateur sur les données d'entraînement disponibles pour la tâche.

5. Ajustez les hyper paramètres et débloquez d'autres couches si nécessaire.

Cette approche a fait ses preuves dans un grand nombre de domaines. C'est un excellent outil à avoir dans son arsenal et c'est généralement la première approche à essayer lorsqu'on est confronté à un nouveau problème de reconnaissance d'images.

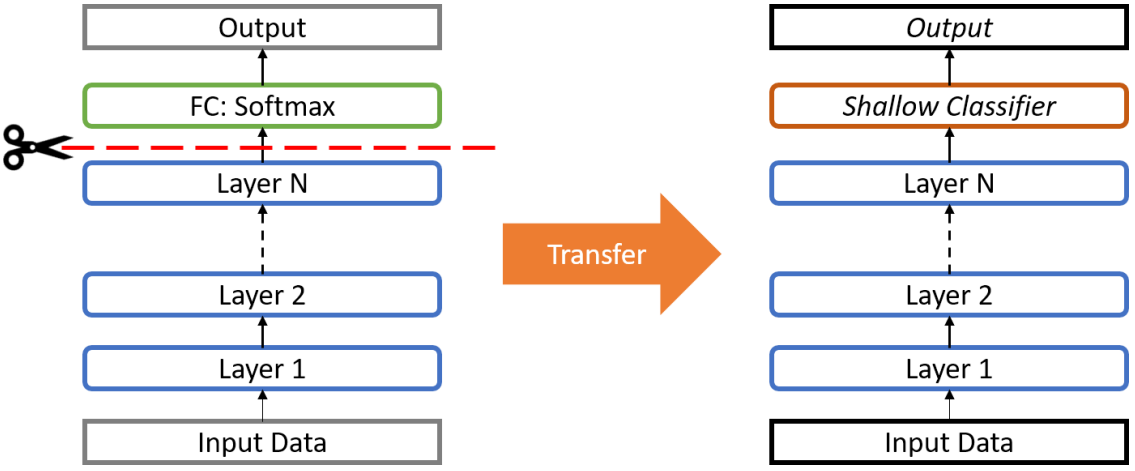


Figure III.8 – Le diagramme représente le transfert profond basé sur l'extraction de la fonction

III.3.3 Détection et comptages des véhicules

Voici notre plan du travail :

1. Comprendre l'idée principale des algorithmes de soustraction d'arrière-plan utilisés pour la détection des avant-plans.

2. Filtres d'image OpenCV.

3. Détection d'objets par les contours.

Les algorithmes de détection d'objets nous permettent d'effectuer des tâches très utiles et à forte valeur ajoutée, comme la surveillance, la gestion du trafic, la lutte contre la criminalité, etc.

Il existe un certain nombre de sous-tâches que nous pouvons effectuer dans la détection d'objets, comme le comptage du nombre d'objets, la détermination de la taille relative des objets ou la détermination de la distance relative entre les objets. Toutes ces soustâches sont importantes car elles contribuent à la résolution de certains des problèmes les plus difficiles du monde réel.

Notre objectif est de capturer les coordonnées de véhicule en mouvement et de mettre en évidence ce véhicule dans la vidéo. Considérons l'image d'une vidéo ci-dessous :

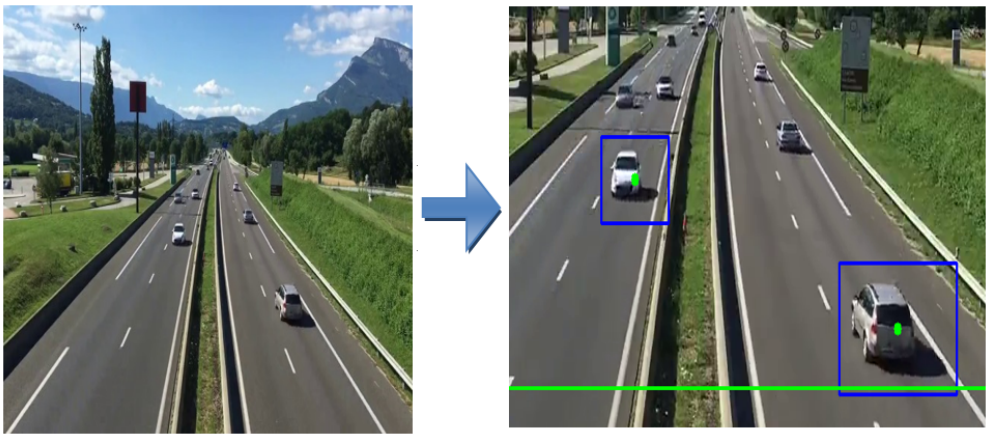


Figure III.9 – la détection des voitures en mouvement dans une vidéo

Nous souhaitons que notre modèle détecte les objets en mouvement dans une vidéo, comme le montre l'image ci-dessus. La voiture en mouvement est détectée et une boîte englobante est créée autour de la voiture.

Il existe plusieurs techniques pour résoudre ce problème. Nous pouvons entraîner un modèle d'apprentissage profond pour la détection d'objets ou nous pouvons choisir un modèle pré-entraîné et l'affiner sur nos données. Cependant, il s'agit d'approches d'apprentissage supervisé et elles nécessitent des données étiquetées pour entraîner le modèle de détection d'objets

nous allons nous concentrer sur la méthode non supervisée de détection d'objets dans les vidéos, c'est-à-dire la détection d'objets sans utiliser de données étiquetées. Nous allons utiliser la technique de différenciation des images. [14]

III.3.3.1 Différenciation des images

Une vidéo est un ensemble d'images empilées dans la bonne séquence. Ainsi, lorsque nous voyons un objet se déplacer dans une vidéo, cela signifie que l'objet se trouve à un endroit différent à chaque image consécutive

Si nous supposons qu'à part cet objet, rien d'autre n'a bougé dans une paire d'images consécutives, alors la différence de pixels entre la première et la deuxième image mettra en évidence les pixels de l'objet en mouvement. Nous aurons alors les pixels et les coordonnées de l'objet en mouvement. C'est en gros la façon dont la méthode de différenciation des images fonctionne.

- Appliquer la différenciation des images sur chaque paire d'images consécutives.
- Appliquer le seuillage d'image sur l'image de sortie de l'étape précédente.
- Effectuer une dilatation d'image sur l'image de sortie de l'étape précédente.
- Trouver les contours dans l'image de sortie de l'étape précédente.

- Réduire la liste des contours apparaissant dans la zone de détection.
- Sauvegarder les images avec les contours finaux.
- Seuil d'image(Image Thresholding).

Dans cette méthode, les valeurs des pixels d'une image en niveaux de gris se voient attribuer l'une des deux valeurs représentant les couleurs noire et blanche en fonction d'un seuil. Ainsi, si la valeur d'un pixel est supérieure à une valeur seuil, on lui attribue une valeur, sinon on lui attribue l'autre valeur.

Dans notre cas, nous allons appliquer le seuillage d'image sur l'image de sortie de la différenciation d'images de l'étape précédente :

Vous pouvez voir qu'une grande partie de la zone en surbrillance indésirable a disparu. Les bords en surbrillance du bloc-notes ne sont plus visibles. L'image résultante peut également être appelée image binaire car elle ne contient que deux couleurs. Dans l'étape suivante, nous allons voir comment capturer ces régions en surbrillance.

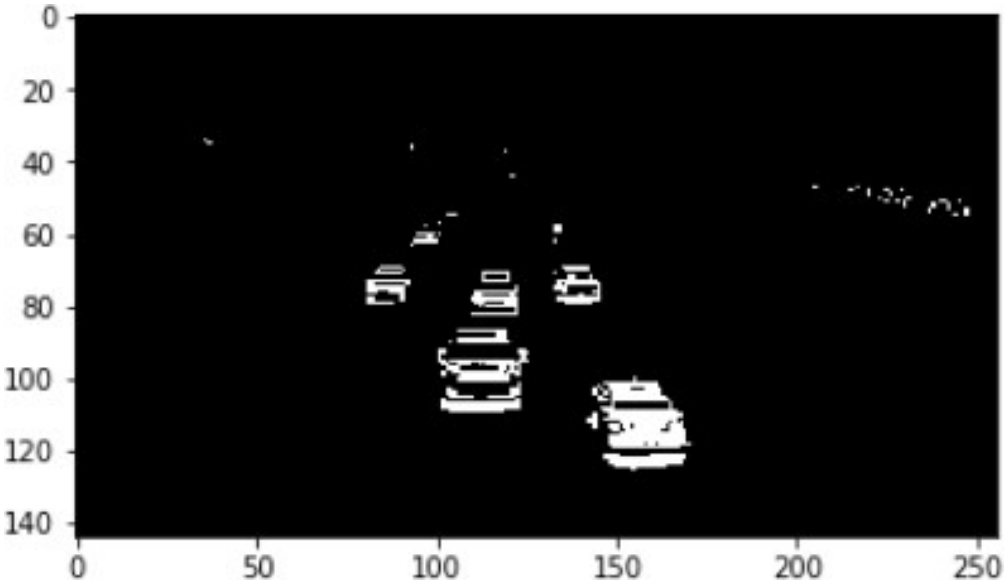


Figure III.10 – image binaire.

III.3.3.2 Trouver des contours

Les contours sont utilisés pour identifier la forme d'une zone de l'image ayant la même couleur ou la même intensité. Les contours sont comme des frontières autour des régions d'intérêt. Ainsi, si nous appliquons les contours sur l'image après l'étape de seuillage, nous obtiendrons le résultat suivant :

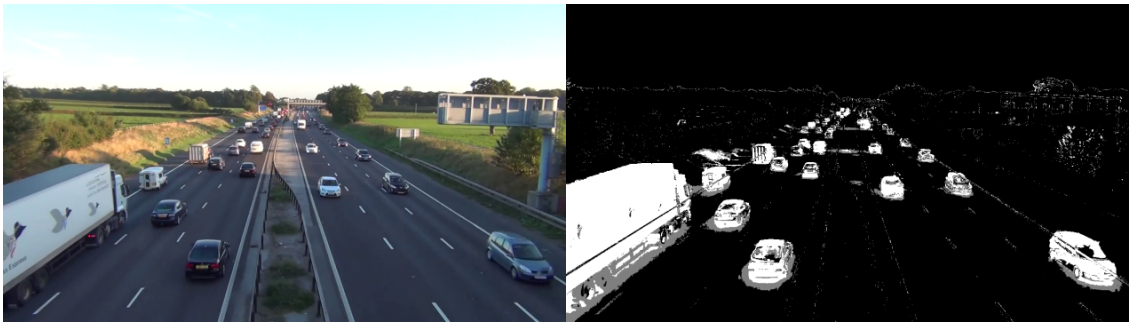


Figure III.11 – trouver des contours sur l'image

Les régions blanches ont été entourées de limites grisâtres qui ne sont rien d'autre que des contours. Nous pouvons facilement obtenir les coordonnées de ces contours. Cela signifie que nous pouvons obtenir l'emplacement des régions en surbrillance.

III.3.3.3 Dilatation d'image

Il s'agit d'une opération de convolution sur une image dans laquelle un noyau (une matrice) est passé sur l'image entière. Pour vous donner une idée, l'image de droite est la version dilatée de l'image de gauche :

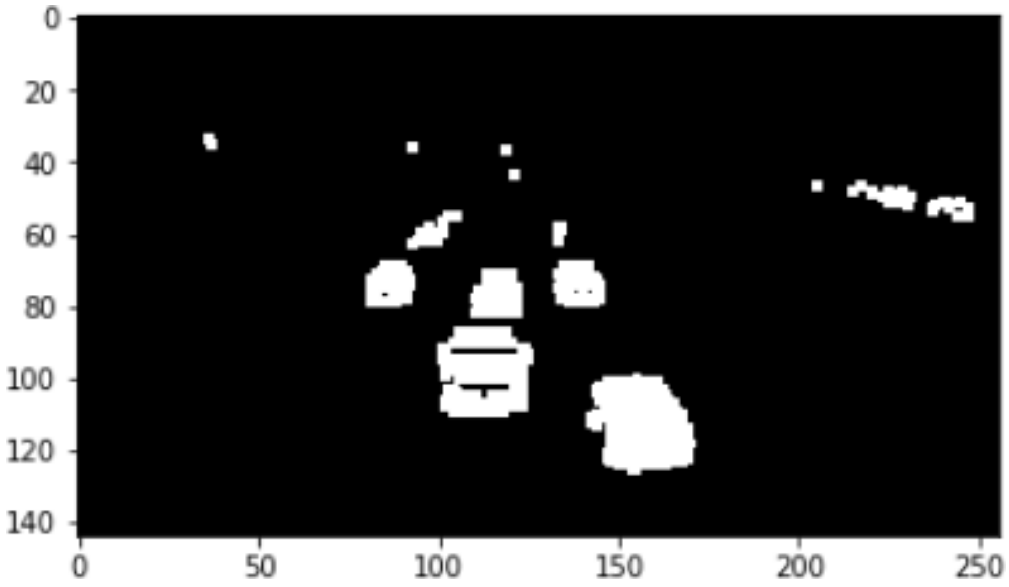


Figure III.12 – la dilatation d'image.

III.3.3.4 Tracking

Lorsque nous ne sommes pas dans la phase de "détection", nous sommes dans la phase de "suivi". Pour chacun de nos objets détectés, nous créons un suiveur d'objet pour suivre l'objet pendant qu'il se déplace dans l'image. Notre traqueur d'objets devrait être plus rapide et plus efficace que le détecteur d'objets. Nous continuons à suivre l'objet jusqu'à ce que nous ayons atteint la N-ième image, puis nous relançons notre détecteur d'objets. Le processus entier se répète alors. Un algorithme simple de suivi d'objet repose sur le

suivi des centroïdes des objets.

En règle générale, un algorithme de suivi d'objets fonctionne de pair avec un détecteur d'objets moins efficace. Le détecteur d'objets est chargé de localiser un objet. Le traqueur d'objets est chargé de garder la trace de chaque objet en attribuant et en maintenant des numéros d'identification (ID).

L'algorithme de suivi d'objets que nous mettons en œuvre est appelé suivi centroïde car il repose sur la distance euclidienne entre les centroïdes d'objets existants (c'est-à-dire les objets que le suiveur centroïde a déjà vus auparavant) et les centroïdes de nouveaux objets entre les images suivantes d'une vidéo. L'algorithme de suivi des centroïdes est un processus en plusieurs étapes. Les cinq étapes sont les suivantes :

- Étape 1 : Accepter les coordonnées des boîtes englobantes et calculer les centroïdes.
- Étape 2 : Calcul de la distance euclidienne entre les nouvelles boîtes englobantes et les objets existants.
- Étape 3 : mise à jour des coordonnées (x, y) des objets existants.
- Étape 4 : enregistrement des nouveaux objets.
- Étape 5 : suppression des anciens objets.

III.3.3.5 Comptage (Counting)

Le comptage était la partie la plus facile, Les véhicules sont comptés lorsqu'ils quittent le cadre ou traversent une ligne à un point de sortie du cadre. L'utilisation d'une ligne de comptage permet de compter plus facilement les véhicules se déplaçant dans une certaine direction.

III.4 Conclusion

Dans ce chapitre, nous avons présenter notre méthode et la technique de détection et de comptage de vehicule.

Chapitre IV :

Résultat de détection et comptage

Chapitre IV

Résultat de détection et comptage

IV.1 Introduction

La détection et le comptage de véhicules peuvent être utilisés pour analyser les profils de Trafic.

Notre projet montre comment utiliser l'analyse de détection pour détecter et compter les Véhicules dans une séquence vidéo, on suppose que la caméra est stationnaire et on va Appliquer les traitements de détection d'objets. Plutôt que de traiter immédiatement toute la vidéo, nous avons commencé par l'obtention d'une image vidéo initiale dans laquelle les Objets en mouvement sont segmentés.

IV.2 Environnement du travail

Dans cette section, nous présentons les environnements matériels et logiciels de notre travail

IV.2.1 Environnement matériel ou le hardware

Un ordinateur dell avec les caractéristiques suivantes :

- Processeur : Intel (R) Core (TM)2 Duo CPU T5750 @2.00 GHz 2.00GHz.
- RAM : 3.00 GO.
- DisqueDur : 500 GO.

IV.2.2 Environnement immatériel ou le software

- Système d'exploitation : Microsoft Windows 10- 64 bits.
- Logiciel PyCharm :Il s'agit d'un IDE (environnement de développement intégré) Spécifiquement utilisé pour le langage python.
- PyCharm est facile à utiliser. Il supporte OpenCV et toutes ses bibliothèques une fois les dépendances installées.
- Il supporte tous les frameworks

- La bibliothèque OPENCV version 2.3.1 : Il s'agit d'une bibliothèque open-source utilisée en vision par ordinateur,l'apprentissage automatique, le traitement d'images Il existe diverses applications et fonctionnalités d'opencv , ce qui le rend polyvalent.

- NumPy : Il s'agit d'une bibliothèque python. Elle est essentiellement utilisée pour effectuer opérations mathématiques et il est facile de travailler avec des tableaux.

  Elle est facile à utiliser et donne d'excellents résultats. Il peut effectuer de grandes opérations mathématiques.

IV.3 les Fonctions utiles de la bibliothèque OpenCV

Dans la suite de ce chapitre, les différentes fonctions et classes utilisées de la bibliothèque OpenCV sont détaillées et expliquées.

IV.3.1 Schéma général

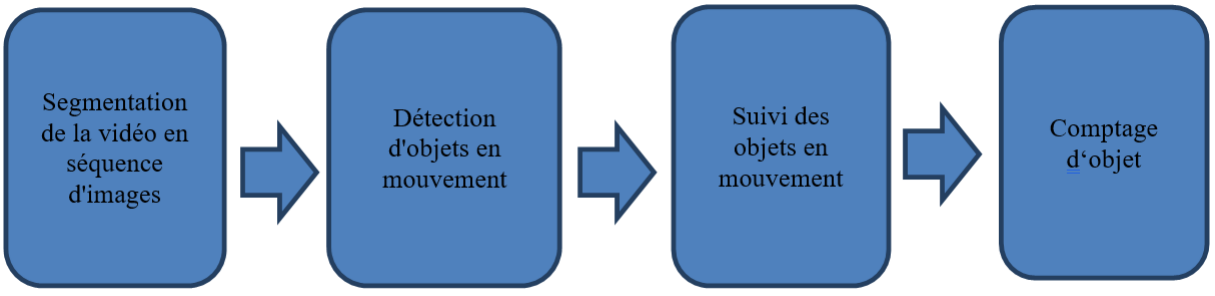


Figure IV.1 – schéma général du travail

IV.3.2 3.2 Détection d'objets en mouvement

Avant de commencer la détection d'objet il faut d'abord transformer la vidéo en images séparées appelés « frame » le travail va se faire sur ses dernière. On a utilisé deux méthodes de détection d'objets la première c'est par la différence d'images et la deuxième par la méthode du modèle de mélanges gaussiens. La détection des objets en mouvement utilise un algorithme de soustraction de l'arrière-plan.

IV.3.3 Pour le traitement de vidéo

Ouvrir et chargement de la vidéo dans la variable cap on utilise cap = cv2.VideoCapture (’aaaaaaa.mp4’)

Stocker l’image issue de la vidéo a l’instant t dans la variable frame(Capture image par image) ret, frame = cap.Read()

Afficher l’image contenue dans ‘frame’ cv2.imshow (’output’, frame)

Sauvegarder l’image (frames) en tant que fichier JPEG cv2.imwrite ("frame%d.jpg" % count, image)

Quitter le programme et fermer toutes les fenêtres ouvertes cap. Release () cv2.destroyAllWindows ()

IV.3.4 Pour le traitement d’image

IV.3.4.1 Transformer une image en niveau de gris

transformer une image couleur en niveau de gris

IV.3.4.2 Découpage des différents objets de l’image

comment retirer le fond d’une image, il serait bien qu’on apprenne à isoler les objets présents dans celle-ci. Cela se fait grâce aux fonctions findContours, drawContours et boundingRect d’OpenCV.

D’après la figure IV.2, le programme va prendre une image de la vidéo que nous avons Choisie pour la traiter :



Figure IV.2 – image prise de la vidéo M6 Motorway Traffic



Figure IV.3 – image prise de la vidéo alger.mp4

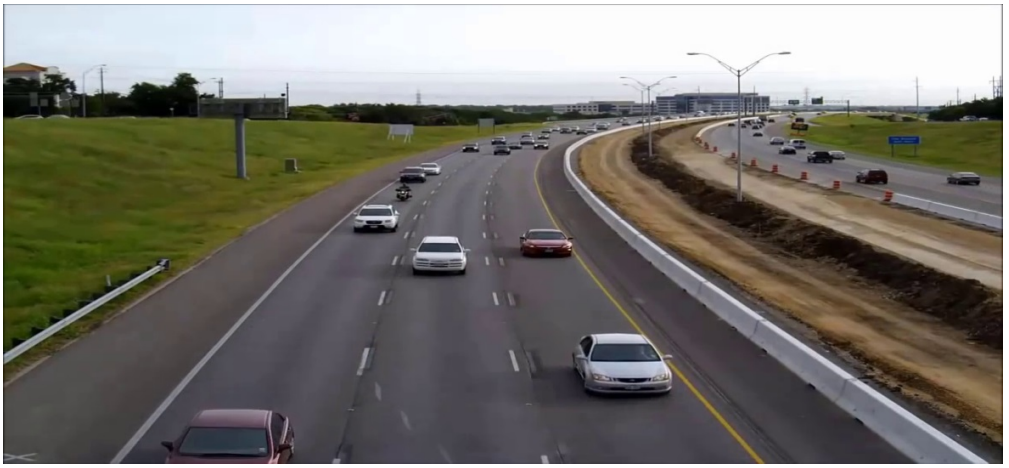


Figure IV.4 – image prise de la vidéo route.mp4

IV.3.4.3 Segmentation :

Dans cette étape l'image sera convertis en binaire (noir et blanc) pour mieux détecté l'objet cherché, le résultat de cette opération est une image noir et blanc dans laquelle les véhicules sont en blanc et tout le reste est en noir avec la présente de quelques trous en blanc

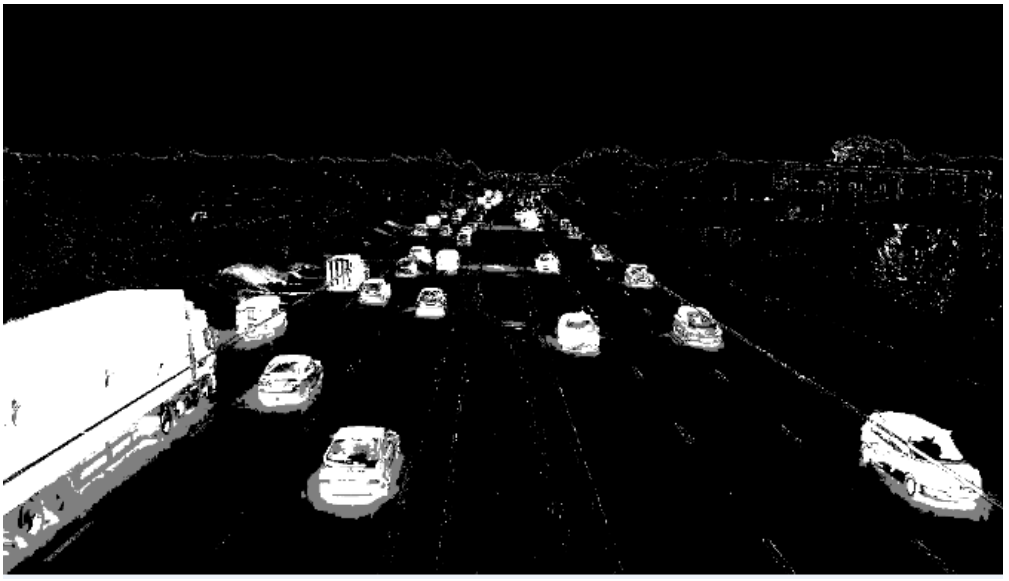


Figure IV.5 – segmentation image de vidéo M6 Motorway Traffic

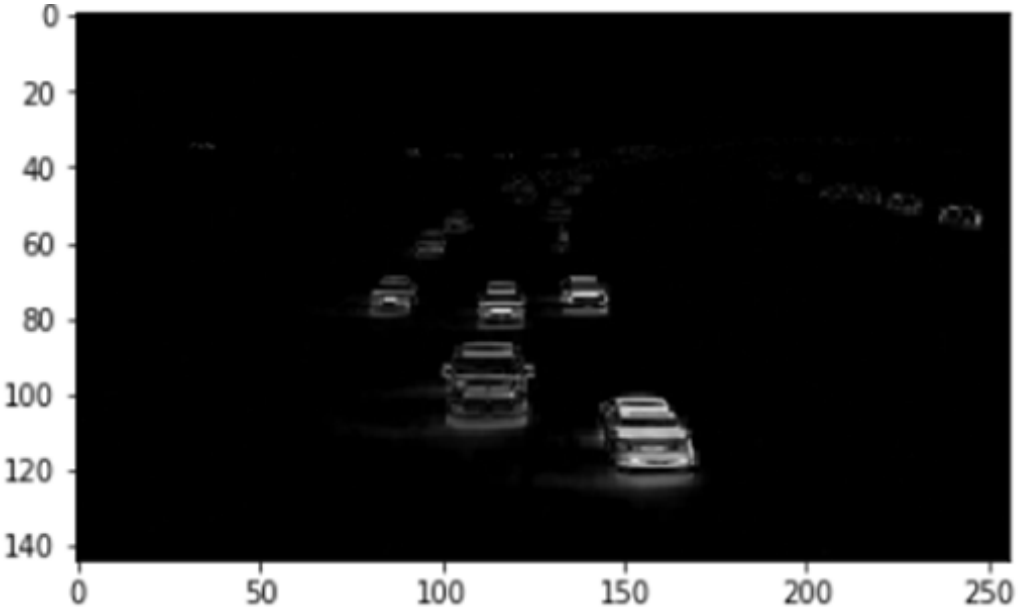


Figure IV.6 – segmentation image de vidéo route.mp4.

IV.3.4.4 Minimisations des bruits :

La segmentation du premier plan n'est pas parfaite et comporte souvent des trous blanc Indésirables. Pour cela on utilise le filtre d'ouverture morphologique (érosion, dilatation) pour Éliminer le bruit et pour combler les lacunes dans les objets détectés.

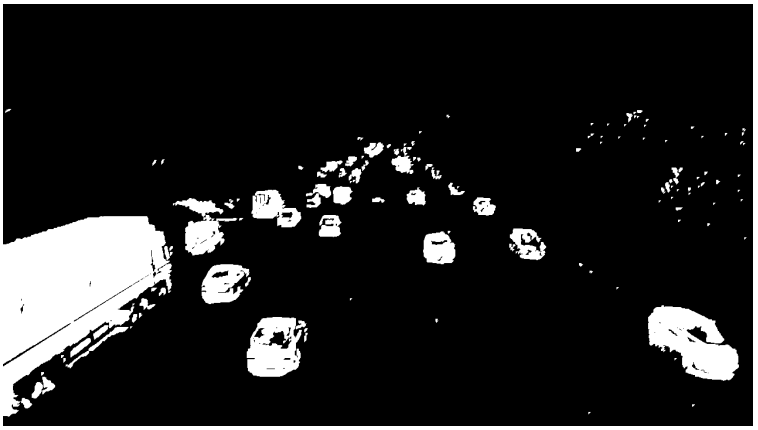


Figure IV.7 – éliminé les pixels isolés vidéo M6 Motorway Traffic



Figure IV.8 – éliminé les pixels isolés vidéo alger.mp4

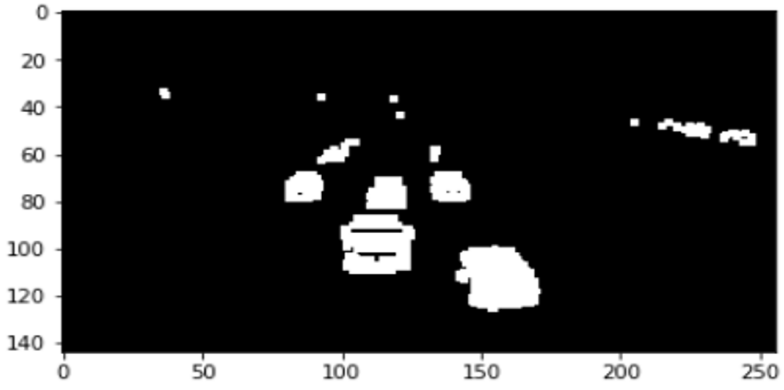


Figure IV.9 – éliminé les pixels isolés vidéo route.mp4

IV.3.4.5 Détection de formes :

Une forme peut être prise comme le contour d'un objet. La détection de forme est Équivalente à détecter la surface de véhicule et pour mettre en évidence les véhicules détectés, Nous dessinons des cases bleues autour d'elles.

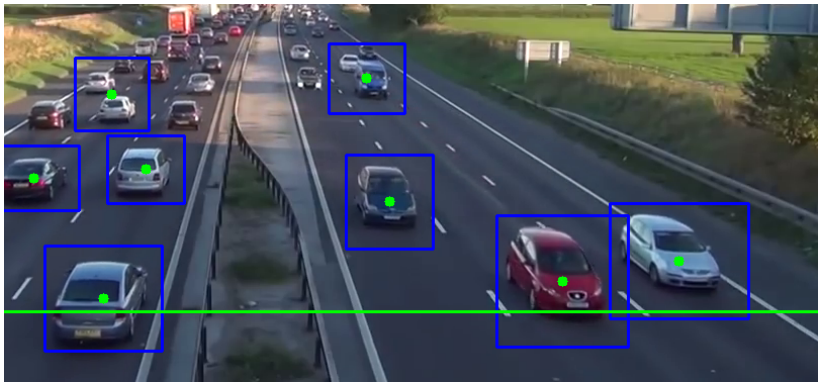


Figure IV.10 – Détection de forme vidéo M6 Motorway Traffic

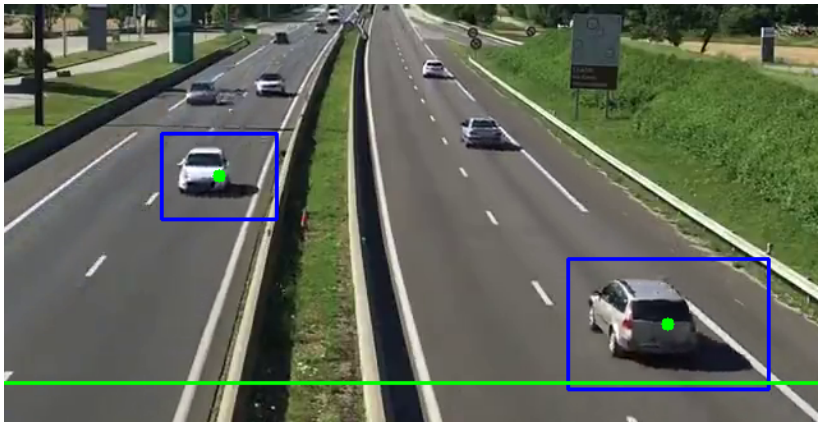


Figure IV.11 – Détection de forme vidéo alger.mp4

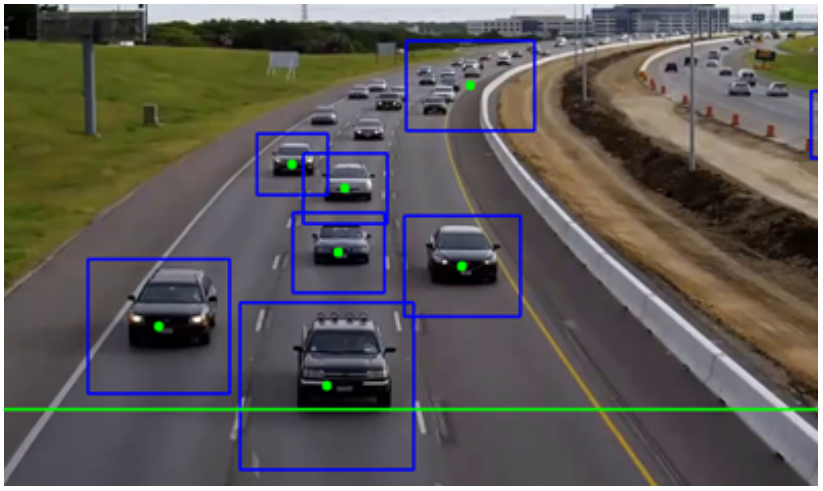


Figure IV.12 – Détection de forme vidéo route.mp4

IV.4 Les resultat de la tecnique OMG2

IV.4.1 Arrière-planSubtractorMOG2

Il s'agit également d'un algorithme de segmentation d'arrière-plan / de premier plan basé sur un mélange gaussien. Il est basé sur deux articles de Z.Zivkovic, "Amélioration

du modèle de mélange adaptatif de Gausian pour la soustraction d'arrière-plan" en 2004 et "Estimation efficace de la densité adaptative par pixel d'image pour la tâche de soustraction d'arrière-plan" en 2006. Une caractéristique importante de cet algorithme est que il sélectionne le nombre approprié de distribution gaussienne pour chaque pixel. (Rappelez-vous, dans le dernier cas, nous avons pris une K distributions gaussiennes dans tout l'algorithme). Il offre une meilleure adaptabilité aux différentes scènes en raison des changements d'éclairage, etc [16]

On a utilisé la technique OMG2 avec la vidéo choisi aaaaaa.mp4 et on trouve les résultats suivante et on compare avec notre technique choisi :

IV.4.2 Segmentation :

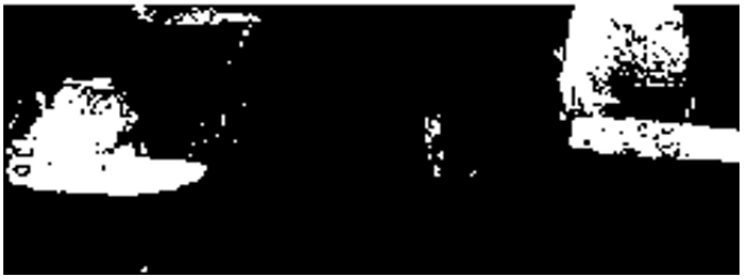


Figure IV.13 – Segmentation OMG vidéo route.mp4

IV.4.3 Minimisations des bruits :

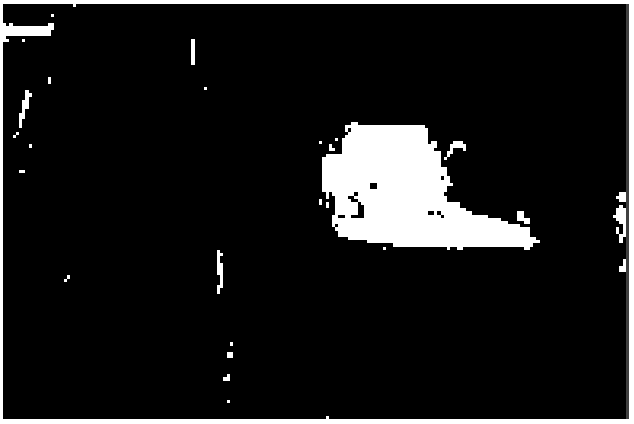


Figure IV.14 – Minimiser le bruit vidéo route.mp4

IV.4.4 Détection de formes :

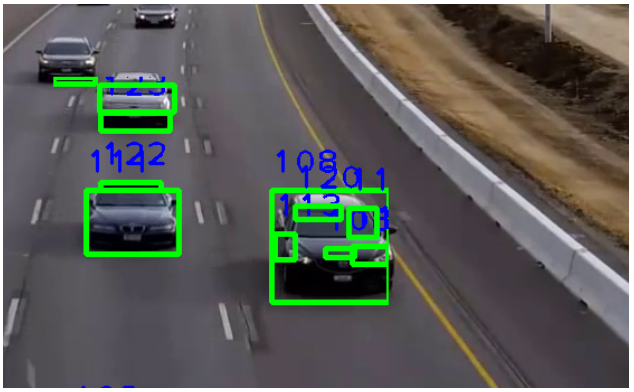


Figure IV.15 – Détection de forme OMG vidéo route.mp4

IV.5 comptages

les résultats de notre comptage obtenus des videos Alger.mp4 et route.mp4 sont présentés respectivement dans les figures suivantes :

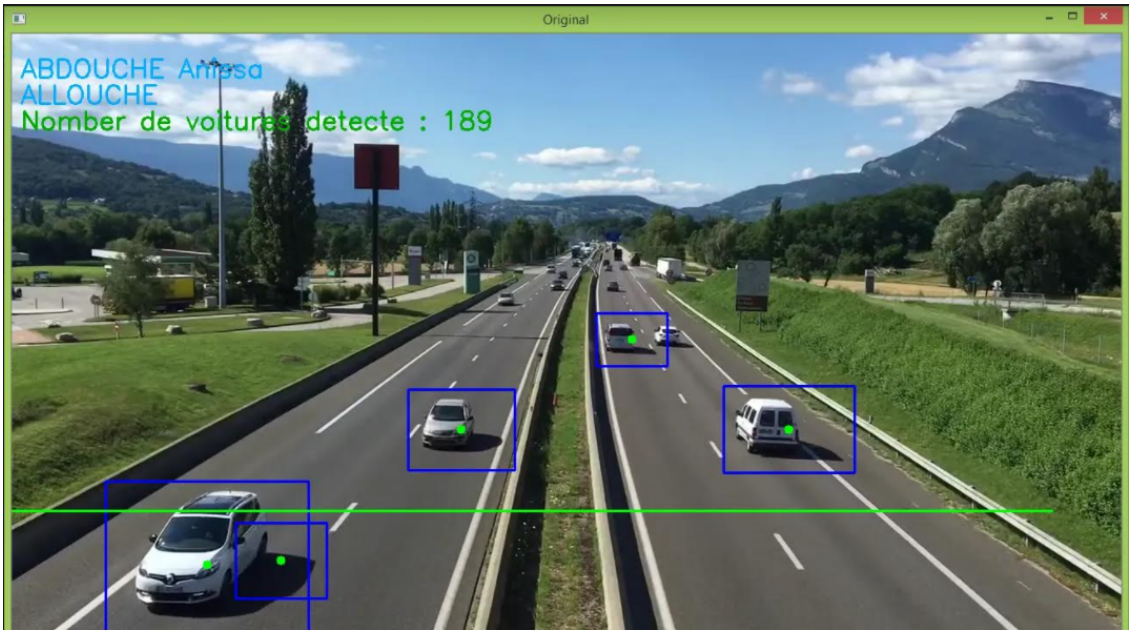


Figure IV.16 – Résultat du comptage de la video Alger.mp4

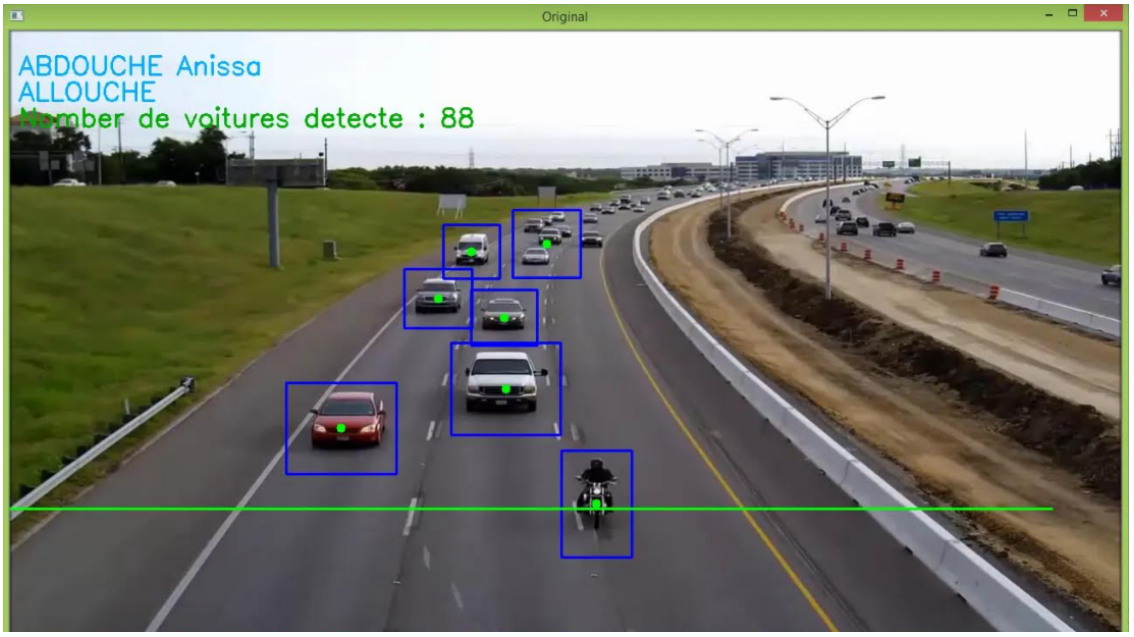


Figure IV.17 – Résultat du comptage de la video route.mp4

nous avons comparé notre algorithme avec l'algorithme OMG2 et nous avons tester avec les cinq vidéos :

Séquences Vidéo	Nombre voitures real	Notre technique	pourcentage	Technique OMG2	Pourcentage
ROUTE.MP4	90	88	97,77 %	70	77.77 %
Alger.MP4	192	189	97.91 %	160	83.33 %
LONDON .MP4	370	359	97.02 %	330	89.18 %
CARVIDEO.mp4	46	44	95.65 %	39	84.78 %

Table IV.1 – Montre une comparaison avec les résultats.

IV.6 conclusion

Dans ce chapitre, on a cité les outils qui nous avons utilisé pour développer notre code ensuite, nous avons présenté quelques résultats expérimentaux concernant les techniques de la détection et le suivi du objet en mouvement.
nous avons arriver à une bonne détection des véhicules dans différentes vidéos grâce à une bonne segmentation.

Conclusion Générale

Conclusion Générale

Dans ce mémoire, nous avons présenté Un méthode de comptage de véhicules a été développé pour suivre les mouvements des véhicules, Cette méthode peut compter automatiquement le nombre de véhicules. Le comptage et la surveillance des véhicules tâches. Basé sur des technologies de pointe de traitement séquentiel de l'image, adaptatif de détection et de comptage de véhicules en temps réel est proposé, la technique produit des résultats de détection très précis. L'adoption d'une méthode de comptage intelligente, qui coopère avec les méthodes de comptage, permet d'améliorer la précision de comptage. Il est simplement compté. J'ai réussi à atteindre la résolution la plus élevée de 97,72 % lors de l'utilisation d'une vidéo. Cette technologie joue un rôle important dans la détection des composés puisque le comptage est la cible de la détection. L'objet "voiture" a la plus grande précision pour compter "moto", "bus", "camion". La fréquence d'images de la vidéo a également un impact sur les performances car elle représente l'intégrité des informations traitées par la technologie, et en général, ce travail a été réalisé avec un bon résultat. À l'avenir, toute amélioration doit être apportée pour parvenir à une meilleure technologie. Les résultats montrent que la précision de détection et de comptage est suffisamment élevée. Nous avons validé comme étant efficace et robuste et meilleur par rapport comptage manuel.

Perspectives

Perspectives

Comme perspectives nous proposons :

- Tester notre application sur une caméra stationnaire.
- Développer la solution proposée en utilisant un langage de programmation évolué.
- Utilisation de vidéo haute résolution (HD).
- Evaluer l'application pour l'utilisé comme un radar qui permet de détectés la vitesse des véhicules et leurs matricule.
- Comptage globale des véhicules dans toute la séquence vidéo pour détecter les vrais encombrements.
- Elimination des véhicules redondantes qui sont détectés plus d'une fois.

Bibliographie

Bibliographie

[1] [1]https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0186098#sec001

[2] Object detection with deep learning : A review ZQ Zhao, P Zheng, S Xu, X Wu - IEEE transactions on neural, 2019 - ieeexplore.ieee.org

[3] An Approach to Counting Vehicles from Pre-Recorded Video Using Computer Algorithms - M Majumder - 2020 - digitalcommons.lsu.edu

[4] Vehicle_Counting_Based_on_Vehicle_Detection_and_Tracking_from_Aerial_Videos Vehicle Counting Based on Vehicle Detection and Tracking from Aerial VideosAugust 2018 https://www.researchgate.net/publication/326869620

[5] Vision-based vehicle detection and counting system using deep learning in highway scenes H Song, H Liang, H Li, Z Dai, X Yun - European Transport Research ..., 2019 - Springer

[6] [6]https://pjreddie.com/darknet/yolo/

[7] Embedded CNN based vehicle classification and counting in non-laned road traffic MS Chauhan, A Singh, M Khemka, A Prateek... - Proceedings of the ..., 2019 - dl.acm.org

[8] Convolutional Neural Network(CNN) with Practical Implementation, Amir Ali,May 22,2019.

[9] Yann LeCun, Leon Bottou, Yoshua Bengio, and Pat- rick Haffner, "Gradientbased learning applied to document recognition," Proceedings of the IEEE, 86(11) :2278–2324, 1998, pp. 1-46

[10] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Ima- genet classification with deep convolutional neural networks," Advances in neural information pro- cessing systems, 2012, pp. 1097–1105

[11] Simonyan, Karen and Andrew Zisserman, "VeryDeep Convolutional Networks for Large-Scale Im-age Recognition," arXiv :1409.1556v6, 2015

[12] Learn Image Classification on 3 Datasets using Convolutional Neural Networks (CNN) MOHD SANAD ZAKI RIZVI, FEBRUARY 18, 2020

[13] Transfer Learning with Convolutional Neural Networks in PyTorch-will koehersen Nov26,2018.

[14] https://www.analyticsvidhya.com/blog/2020/04/vehicle-detection-opencv-python/

[15] Zaynab Al-Ariny, Mohamed A Abdelwahab, Mahmoud Fakhry, El-Sayed Hasanee 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), 232-237, 2020.

[16] https://try2explore.com/questions/fr/11235050