

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة عين تموشنت بلحاج بوشعيب  
Université –Ain Temouchent- Belhadj Bouchaib  
Faculté des Sciences et de Technologie  
Département Électronique et Télécommunication



Projet de Fin d'Etudes  
Pour l'obtention du diplôme de Master en :  
Domaine : Génie Électrique  
Filière : Électronique  
Spécialité : Instrumentation  
Thème

## **Détection et Classification de Véhicule dans un Environnement Routier**

**Présenté Par :**

- 1) Melle. Boukernafa Chaimaa
- 2) Melle. Bouazza-abid Soumia

**Devant le jury composé de :**

Mme BOUTKHIL Malika	MAA	UAT.B.B (Ain Temouchent)	Présidente
Dr BENGANA Abdelfatih	MCB	UAT.B.B (Ain Temouchent)	Examineur
Dr BENTAIEB Samia	MCB	UAT.B.B (Ain Temouchent)	Encadrante

*Année Universitaire 2020/2021*

*Avec l'expression de ma reconnaissance, je dédie ce modeste travail a ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.à mes parents, les deux personnes qui m'ont donné les outils et les valeurs nécessaires pour être là où je suis aujourd'hui. à mes chères frères qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études .que dieu les protège et leurs offre la chance et le bonheur. Sans oublier ma partenaire chaimaa pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

***Soumia***

*Je dédie ce projet : A ma très chère mère, Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes cotes a toujours été ma source de force pour affronter les différents obstacles. A mon très cher père, Tu as toujours été a mes cotes pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection. A mon très cher frère Walid , malgré ta distance mais ton amour pour moi m'a rendue fier d'avoir un frère comme toi . . . j'espère que le Dieu Tout-Puissant vous guérira . A mon amie et sœur qui m'ont aidé pour réalisée ce projet mon partenaire Soumia. ma famille et toutes les personnes que j'aime de près ou de loin. Puisse Dieu vous donne santé, bonheur, courage et surtout réussite .*

**Chaimaa**

# Remerciements

Avant de présenter ce travail tout d'abord, nous tiens à rendre grâce à Dieu Le Tout Puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire, de nous avoir éclairé notre route et de nous avoir guidé dans le bon chemin.

Ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et les conseils du Dr. BENTAIEB Samia, nous la remercions de nous avoir donné l'opportunité de faire des recherches et de nous avoir fourni de précieux conseils tout au long de cette recherche. Ce fut un grand privilège et un honneur de travailler sous sa direction. Nous sommes extrêmement reconnaissantes pour ce qu'elle nous a offert.

Je tiens à remercier avec plus grande gratitude Madame BOUTKHIL Malika de l'honneur qu'elle nous fait d'avoir acceptée de présider le jury de ce mémoire. Nous remercions également Dr. BENGANA Abdelfatih d'avoir accepté de se joindre à ce jury comme examinateur.

# Table des matières

<b>Table des figures</b>	<b>vii</b>
<b>Liste des algorithmes</b>	<b>viii</b>
<b>Glossaire</b>	<b>x</b>
<b>1 Classification des Véhicules</b>	<b>7</b>
1.1 Introduction . . . . .	8
1.2 Systèmes Intelligents de Trafic Routier . . . . .	8
1.3 Détection et Classification des Véhicules . . . . .	11
1.3.1 Détection des véhicules . . . . .	11
1.3.2 Classification des véhicules . . . . .	13
1.4 La Détection et la Classification des Véhicules : État de l'Art . . . . .	14
1.5 Conclusion . . . . .	21
<b>2 L'Intelligence Artificielle pour la Classification des Véhicules</b>	<b>22</b>
2.1 Introduction . . . . .	23
2.2 L'intelligence Artificielle (IA) . . . . .	23
2.3 Les applications de l'intelligence artificielle . . . . .	24
2.4 Machine Learning (ML) . . . . .	24
2.4.1 L'apprentissage supervisé . . . . .	25
2.4.2 L'apprentissage non supervisé . . . . .	25
2.4.3 L'apprentissage semi supervisé . . . . .	25
2.5 Les principaux Algorithmes du Machine Learning (ML) . . . . .	26
2.5.1 K-Nearest Neighbor(K-NN) . . . . .	26

2.5.2	Naive Bayes (NB)	26
2.5.3	Support Vector Machine (SVM)	27
2.5.4	Deep Learning (DL)	27
2.6	Conclusion	28
<b>3</b>	<b>Détection et Classification des Véhicules</b>	<b>29</b>
3.1	Introduction	31
3.2	Approche Proposé	31
3.3	Traitement d'images	31
3.3.1	Conversion d'une image RGB en niveau de gris	32
3.3.2	Conversion du niveaux de gris en images binaires	32
3.3.3	Masque d'images	33
3.3.4	Filtre Gaussien	33
3.3.5	Dilatation et érosion	33
3.3.6	Segmentation par seuillage	34
3.3.7	Détection de contour	34
3.4	Histogram of Oriented Gradients (HOG)	35
3.5	K-Nearest Neighbor (KNN)	37
3.6	Support Vector Machine (SVM)	39
3.6.1	Hyperplan non-linéaire	40
3.6.2	Utilisation des SVM avec des kernels	41
3.6.3	Support Vector Machine multiclassés	43
3.7	Évaluation	44
3.7.1	Base de données	44
3.7.2	L'environnement de travail	45
3.7.3	Protocole d'évaluation	48
3.8	Conclusion	50
<b>4</b>	<b>Détection et Classification dans un Flux Routier</b>	<b>53</b>
4.1	Introduction	54
4.2	Background Subtraction	54
4.2.1	Les différents étapes de la soustraction d'arrière-plan	54
4.2.2	Classification des véhicules en utilisant HOG-RBF-SVM	57
4.3	Présentation de la Raspberry Pi	59
4.3.1	Raspbian	61

4.3.2	Configuration du système d'exploitation Raspbian sur les cartes Raspberry Pi . . . . .	61
4.3.3	Programmation Python . . . . .	62
4.4	Conclusion . . . . .	66
	<b>Bibliographie</b>	<b>70</b>

# Table des figures

1.1	Les composant d'un système de transport intelligent . . . . .	9
1.2	Détection du la trafic routier . . . . .	11
1.3	Les principales approches de détection de véhicules . . . . .	12
1.4	diagramme de l'approche proposée dans [Dong et al., 2015] . . . . .	15
1.5	Le réseau neuronal convolutif de classification des types de véhicule proposé dans [Dong et al., 2015] . . . . .	15
1.6	Extraction du véhicule : (a), (b) échantillon de l'image d'entrée (c), (d) échantillon de l'image d'extraction d'avant-plan [Seenouvong et al., 2016] . . . . .	16
1.7	Illustration du pipeline de segmentation avant détection, la détection et la classification propose dans [Audebert et al., 2017] . . . . .	18
1.8	Résultat de détection de véhicules (jour et nuit) [Tsai et al., 2018] . . . . .	18
1.9	Schéma fonctionnel du système proposé dans [Velazquez-Pupo et al., 2018] . . . . .	19
1.10	Schéma de la méthode MoG+SVM pour la détection et la classifica- tion des véhicules [Arinaldi et al., 2018] . . . . .	20
1.11	(a) Scène de péage TMII; (b) Scène de péage Kapuk; (c) Scène de trafic MIT; (d) Classes de véhicules de péage indonésiens; (e) Classes de véhicules de trafic MIT [Arinaldi et al., 2018] . . . . .	20
1.12	(a) Résultats sur la scène de péage de Kapuk; (b) Résultats sur la scène de péage du TMII; (c) Résultats sur la scène de trafic du MIT . . . . .	20
1.13	Organigramme de l'algorithme STSC amélioré [Wang et al., 2019] . . . . .	21
2.1	Différence entre ML et DL . . . . .	28

3.1	Structure des systèmes de classification . . . . .	32
3.2	Résultat de l'opérateur de dilatation utilisant un noyau $3 \times 3$ . . . . .	34
3.3	Résultat de l'opérateur érosion utilisant un noyau croisé $3 \times 3$ . . . . .	34
3.4	Image d'un véhicule et de ses différentes représentations HOG . . . . .	37
3.5	Méthode de classification KNN avec $K = 3$ . . . . .	38
3.6	Support Vector Machine . . . . .	39
3.7	Sélection d'hyperplans dans un espace de caractéristiques bidimensionnel. . . . .	40
3.8	Sélection d'un hyperplan dans un espace de caractéristiques à $n$ -dimensions, où représente la cartographie . . . . .	41
3.9	Illustration du SVM avec un noyau linéaire . . . . .	42
3.10	La méthodes one-to-rest : Classification entre 3 classes . . . . .	43
3.11	La méthodes one-to-one : Classification entre 3 classes . . . . .	44
3.12	small class images . . . . .	45
3.13	medium class images . . . . .	46
3.14	large class images . . . . .	46
3.15	La matrice de confusion du SVM- Linéaire-HOG . . . . .	50
3.16	La matrice de confusion du SVM-poly-HOG . . . . .	51
3.17	La matrice de confusion du SVM-RBF-HOG . . . . .	51
3.18	La matrice de confusion du KNN-HOG . . . . .	52
4.1	La différence entre l'image de vidéo et background . . . . .	56
4.2	Image filtrée par un noyau Gaussien . . . . .	57
4.3	Seuillage par la méthode d'Otsu . . . . .	57
4.4	Image après la dilatation . . . . .	58
4.5	Détection de véhicules dans une scène vidéo . . . . .	58
4.6	Quelque résultats de classification avec le nouveau modèle . . . . .	59
4.7	L'architecture de la Raspberru Pi 4 . . . . .	61
4.8	Menu de configuration Raspbian . . . . .	62
4.9	Un outil de configuration de la Raspberry Pi4 . . . . .	63
4.10	Thonny IDE application . . . . .	63
4.11	Exemple d'écriture sur l'application Thonny . . . . .	64
4.12	Accéder à un bureau Raspbian par le biais de l'application VNC Viewer . . . . .	64
4.13	Outils de programmation sur le bureau de Raspbian . . . . .	65
4.14	le chemin du dossier CAR Project . . . . .	65

4.15	Les fichiers envoyés vers la Raspberry . . . . .	66
4.16	Les fichiers stockés dans le dossier CAR Project . . . . .	66
4.17	Temps d'exécution de la détection et la classification sur la Raspberry Pi4 . . . . .	67
4.18	Les résultats de la détection de véhicules sur la Raspberry Pi4 . . . . .	67
4.19	Quelques résultats de classification sur la Raspberry Pi4 . . . . .	68

# Liste des tableaux

1.1	Nombre réel de véhicules dans cinq vidéos . . . . .	17
3.1	Opérateurs de détection de contours . . . . .	35
3.2	Paramètres du descripteur HOG . . . . .	36
3.3	Matrice de confusion d'un problème à deux classes. . . . .	48
3.4	Taux de classification des algorithmes KNN et SVM avec le descripteur HOG . . . . .	49
3.5	Taux de classification des algorithmes KNN et SVM sans le descripteur HOG . . . . .	49
4.1	Comparaison entre différents modèles de Raspberry . . . . .	60

# Liste des algorithmes

1	Algorithme du plus proche voisins K-NN . . . . .	37
---	--	----

# Abréviations

ACF : **A**ggregate **C**hannel **F**eatures  
AGI : **A**rtificiel **G**eneral **I**ntelligence  
AI : **A**rtificial **I**ntelligence  
ANFIS : **A**aptive **N**euro **F**uzzy **I**nterference **S**ystem  
CNN : **C**onvolutional **N**eural **N**etwork  
DL : **D**eep **L**earning  
GMM : **G**aussian **M**ixture **M**odel  
GPS : **G**lobal **P**ositioning **S**ystem  
HOG : **H**istogram of **O**riented **G**radient  
IDE : **I**ntegrated **D**evelopment **E**nvironment  
IDLE : **I**ntegrated **D**evelopment and **L**earning **E**nvironment  
ITS : **I**ntelligent **T**ransport **S**ystem  
KNN : **K**- **N**earest **N**eighbors  
ML : **M**achine **L**earning  
MoG : **M**ixture of **G**aussians  
NB : **N**aive **B**ayes  
OC-SVM : **O**ne **C**lass **S**upport **V**ector **M**achine  
PVANet : **P**erson **V**ehicle **A**nimal detection **N**etwork  
PTZ : **P**an- **T**ilt- **Z**oom  
R-CNN : **R**egion-based **C**onvolutional **N**eural **N**etwork  
ReLU : **R**ectified **L**inear **U**nits  
SMS : **S**hort **M**essage **S**ystem  
STSC : **S**patio **T**emporal **S**ample **C**onsensus  
SVM : **S**upport **V**ector **M**achine

VANet : **V**ehicular **A**dhor **N**etworks

VHR : **V**ery **H**igh **R**esolution

YOLO : **Y**ou **O**nly **L**ook **O**nce

## Résumé

La détection et la classification précise des véhicules joue un rôle important dans les systèmes de transport intelligents. La capacité de détecter les véhicules dans les scènes de circulation permet d'analyser le comportement des conducteurs, ainsi que de détecter les infractions au code de la route et les accidents. La détection et la classification des véhicules est une tâche difficile en raison des conditions météorologiques, de l'éclairage et de la diversité des types de véhicules. Plusieurs solutions utilisent des algorithmes d'extraction de caractéristiques ainsi que le classifieur Support Vector Machine. Dans ce projet, nous avons présenté les algorithmes principaux utilisés pour la classification des véhicules. L'histogramme de gradient orienté HOG a été implémenté pour extraire les caractéristiques des images représentées sous forme de vecteur HOG. La méthode de soustraction du background est utilisée pour détecter les véhicules. Le classifieur SVM avec le noyau RBF a été entraîné et testé pour classifier les véhicules sous plusieurs classe. Après avoir prouvé la fiabilité et la robustesse du modèle, la carte Raspberry Pi4 a été utilisée pour implémenter le système proposé.

**Mots-clés** : véhicule, classification, détection, HOG, KNN, SVM, Raspberry.

## المخلص

يلعب الكشف الدقيق عن المركبات وتصنيفها دورًا مهمًا في أنظمة النقل الذكية. تساعد القدرة على اكتشاف المركبات في المشاهد المرورية على تحليل سلوك السائق ، وكذلك اكتشاف المخالفات المرورية والحوادث. يعد اكتشاف المركبات وتصنيفها مهمة صعبة بسبب الظروف الجوية والإضاءة وتنوع المركبات. تستخدم العديد من الحلول خوارزميات استخراج الميزات بالإضافة إلى مصنف *SupportVectorMachine* . في هذا المشروع قدمنا الخوارزميات الرئيسية المستخدمة لتصنيف المركبات. تم تنفيذ المدرج التكراري الموجه *HOG* لاستخراج خصائص الصور المثلثة كمتجه *HOG* . يتم استخدام طريقة الطرح بخطوات مختلفة لاكتشاف المركبات. تم تدريب مصنف *RBFsvm* واختباره لتصنيف المركبات تحت عدة فئات. بعد إثبات الموثوقية وقوة النموذج ، تم استخدام لوحة *RaspberryPi4* لتنفيذ النظام المقترح.

الكلمات الرئيسية : المركبات ، التصنيف ، الكشف ، *HOG* ، *KNN* ، *SVM* ،

*Raspberry*

## Abstract

The accurate detection and classification of vehicles plays an important role in intelligent transportation systems. The ability to detect vehicles in traffic scenes allows for the analysis of driver behavior, as well as the detection of traffic violations and accidents. Detecting and classifying vehicles is a difficult task due to weather conditions, lighting, and the diversity of vehicle types. Several solutions use feature extraction algorithms as well as the Support Vector Machine classifier. In this project, we presented the main algorithms used for vehicle classification. The HOG oriented gradient histogram has been implemented to extract the features from the images represented as HOG vector. The background subtraction method is used to detect the vehicles. The SVM with RBF kernel classifier has been trained and tested to classify the vehicles under several classes. After proving the reliability and robustness of the model, the Raspberry Pi4 board was used to implement the proposed system.

**Keywords:** vehicle, classification, detection, HOG, SVM, Raspberry.

# Introduction générale

## Contexte

La détection et la classification d'objets, précisément les véhicules, par vision artificielle est un domaine actif de recherche. C'est également l'un des problèmes les plus difficiles de la vision par ordinateur. Il s'agit en même temps d'une étape essentielle pour la mise en œuvre de plusieurs applications actuelles qui nécessitent une interprétation de haut niveau des images. Notons le besoin croissant de développer des systèmes de détection et de classification de véhicules pour la grande quantité de vidéos qui sont générées.

L'objectif des systèmes de surveillance et de contrôle automatisés est de supprimer le besoin de main-d'œuvre humaine pour des tâches simples basées sur la vision qui peuvent être effectuées par un ordinateur ou un système automatisé. Les applications des systèmes de vision par ordinateur ont également été appliquées dans divers lieux publics tels que les routes, les aéroports etc. Une de ces applications consiste à analyser des scènes de trafic routier, avec un intérêt particulier pour la surveillance des autoroutes et des carrefours.

Les systèmes intelligents de détection d'images font partie d'une approche centralisée de la modernité de la gestion du trafic. Cela est dû à la nécessité d'une gestion du trafic plus rentable et un suivi efficace. Cela a permis d'accroître les possibilités de contrôle automatique des analyses de l'activité du trafic urbain à partir de la vidéo surveillance.

Le cœur de tout système de ce type qui peut être utilisé efficacement est la détection et la classification précises des véhicules en mouvement présents dans la vidéo. La détection et la classification réussies des classes de véhicules sont essentielles pour

extraire les informations importantes sur le flux de trafic dont ont à besoin les régulateurs, telles que le nombre de véhicules, la vitesse moyenne estimée des véhicules, le comportement des conducteurs et les violations du code de la route (comme les camions qui utilisent les voies à grande vitesse).

## Motivation

La détection et la classification des véhicules sont des tâches triviales pour les humains. Le système visuel humain est capable de distinguer, d'une part, les véhicules de l'arrière-plan d'une image et, d'autre part, le type de véhicule présent dans une scène. Dans le cas de la vision par ordinateur, le processus est beaucoup plus difficile.

La détection et la classification des véhicules jouent un rôle important dans l'analyse du comportement des véhicules et la détection des incidents de circulation pour les systèmes de vidéo surveillance établis sur les voies rapides. La nouvelle tendance des systèmes de transport intelligents comporte de nombreux éléments tels que la détection des véhicules, les panneaux de signalisation, la marque et le type des véhicules, etc. La détection et la classification des véhicules nous donne la possibilité d'améliorer les flux de circulation, de prévenir les accidents, et d'enregistrer les crimes et les violations du trafic, contrôle des infractions.

Les systèmes de transport intelligents consistent en un processus progressif et réussi qui doit pouvoir être adapté aux différentes conditions météorologiques et d'éclairage. Ses principales difficultés sont les ombres, les phares des véhicules et le mauvais éclairage et les mauvaises conditions météorologiques. Dans de telles situations, la détection du véhicule en mouvement sera un travail difficile.

La classification des véhicules peut fournir des informations précises sur la densité des véhicules sur une route urbaine ou une autoroute particulière, ce qui permet une surveillance et une gestion avancées du trafic pour mieux utiliser les réseaux routiers, prévoir les besoins futurs en matière de transport et améliorer la sécurité des transports.

L'objectif final de ce travail est de produire un système pour guider les opérateurs de surveillance et réduire les ressources humaines.

## Objectif

L'objectif principal de notre travail est de développer un système de surveillance intelligent en remplaçant la vieille méthode traditionnelle de surveillance utilisant des opérateurs humains et de concevoir un système intelligent de surveillance pour la détection et la classification des véhicules. La contribution majeure de ce travail consiste en la détection et la classification de véhicules à l'aide de technique de l'intelligence artificielle et de son implémentation dans un Raspberry Pi 4.

## Structure

Notre travail est composé de quatre chapitres structurés comme suit. Dans le chapitre 1, une brève revue de l'état de l'art des principales méthodes de détection et de classification des véhicules est présenté. Le chapitre 2 présente principalement l'intelligence artificielle et les principaux algorithmes du machine learning utilisés dans le domaine de la classification. Le chapitre 3 aborde le descripteur utilisé pour la représentation d'un véhicule. Les différents algorithmes du machine learning utilisés dans ce travail sont présentés, évalués et discutés. Le chapitre 4 est une mise en place du schéma utilisé dans la détection et la classification de véhicules dans un environnement routier sur la Raspberry Pi4. Enfin, une conclusion et directions possibles pour les travaux futurs sont présentées à la fin de ce manuscrit.

Chapitre **1**

# Classification des Véhicules

## Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>8</b>
<b>1.2</b>	<b>Systèmes Intelligents de Trafic Routier</b>	<b>8</b>
<b>1.3</b>	<b>Détection et Classification des Véhicules</b>	<b>11</b>
1.3.1	Détection des véhicules	11
1.3.2	Classification des véhicules	13
<b>1.4</b>	<b>La Détection et la Classification des Véhicules : État de l'Art</b>	<b>14</b>
<b>1.5</b>	<b>Conclusion</b>	<b>21</b>

---

## 1.1 Introduction

Le transport est l'un des facteurs les plus importants qui déterminent le développement économique de chaque pays. La mise en œuvre de chaque processus de transport, qu'il s'agisse du transport de passagers ou de marchandises, s'effectue en utilisant des réseaux de transport et en les complétant par différents types d'infrastructures. La dynamique des changements dans le monde moderne, la complexité de l'infrastructure routière ainsi que la mobilité sans cesse croissante de la société contemporaine constituent des conditions favorables à la conception de systèmes innovant en matière de transport. Par conséquent, les activités dans le domaine des réseaux et systèmes de transport nécessitent de nouvelles décisions qui sont prises en permanence en matière de planification, d'organisation de gestion et de contrôle du trafic. Ces décisions impliquent la nécessité de rechercher des solutions de plus en plus efficaces, d'entreprendre des nombreuses stratégies visant à répondre aux besoins actuels de la société.

Par ailleurs, les traumatismes dus aux accidents de la route constituent un problème de santé publique mondial majeur mais négligé, nécessitant des efforts concertés pour une prévention efficace et durable. Parmi tous les systèmes avec lesquels les gens doivent composer quotidiennement le transport routier est le plus complexe et le plus dangereux. Chaque année dans le monde, on compte 1,35 million de décès sur les routes pour 20 à 50 millions de blessés, soit 1 mort toutes les 21 secondes environ (compteur). L'organisation mondiale de la santé prévoit qu'en 2030, 2.3 millions de personnes mourront à la suite d'un accident de la route<sup>1</sup>. Notons qu'en Algérie, on comptait 1065 morts et 9708 blessés durant les 5 premiers mois de 2020<sup>2</sup>.

## 1.2 Systèmes Intelligents de Trafic Routier

Les Systèmes de Transport Intelligents (STI) sont une tendance mondiale qui suscite l'intérêt des professionnels du transport, de l'industrie automobile et de la communauté scientifique. Les STIs sont liés aux technologies avancées de communication, d'information et d'électronique pour résoudre les problèmes de transport tels que la congestion du trafic, la sécurité, l'efficacité du transport et la conservation de l'environnement.

---

1. <https://www.planetoscope.com/mortalite/1270-.html>

2. <https://www.aps.dz/societe/107774-accidents-de-la-route>

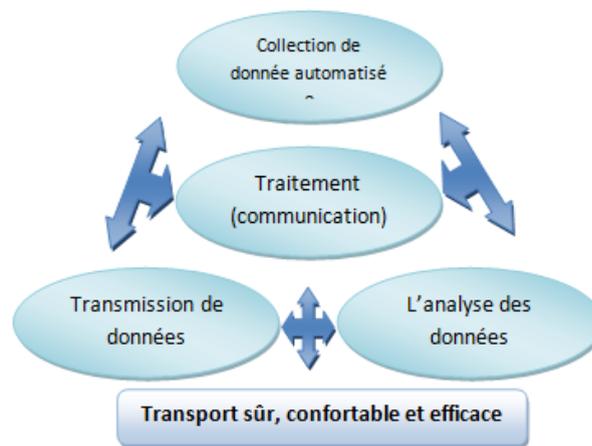


FIGURE 1.1 – Les composant d’un système de transport intelligent

Les composants d’un STI, illustrés dans la Figure 1.1, peuvent être définis comme suit :

- **Collecte automatisée des données** : elle nécessite une planification stratégique précise par le biais de matériel et de logiciels performants. Le localisateur de véhicules par GPS, les caméras, les capteurs, etc. Sont quelques-uns des matériels utilisés pour la collecte de données. Avec cette grande quantité de données, la surveillance et comptage du trafic, l’estimation de la vitesse, l’emplacement des véhicules et leur tracking peuvent être fait.
- **Transmission de données** : c’est un aspect essentiel de la communication rapide et en temps réel dans la mise en œuvre des STIs. L’information peut être communiquée par une annonce, relative au trafic, au voyageur par SMS, Internet, unités embarquées dans les véhicules, etc.
- **Analyse des données** : elle contient la rectification des erreurs, le nettoyage des données et les synthèses de données. Les données traitées sont ensuite analysées pour prévoir le trafic. Les informations en temps réel comme le temps de trajet, le retard, les accidents sur les routes, les changements d’itinéraire, les zones de travaux, les déviations, etc. Sont obtenues après l’analyse des données [Mandhare et al., 2018].
- **Système avancé de gestion du trafic** : est utilisé par les services de police de la circulation et les autorités de régulation du trafic comme un outil pour gérer et contrôler le trafic en surveillant le flux de circulation et en prenant

les décisions appropriées en temps opportun. Les systèmes de gestion du trafic optimisent le mouvement des véhicules, en utilisant des informations en temps réel pour interférer et ajuster les contrôles tels que les feux de circulation afin d'améliorer la fluidité du trafic.

- **Système de gestion des urgences** : est le domaine de recherche le plus récent dans le système de transport intelligent. Il s'intéresse principalement à l'application de différentes technologies de systèmes de transport intelligents pour développer un système de transport capable de fournir une aide dans des conditions d'urgence [Lan and Chang, 2014].

L'architecture et les différents modèles développés au fil des années dans les quatre branches des STIs, citées ci-dessus, ont été pris en compte par les chercheurs dans leurs études sur l'utilisation des communications entre Véhicules et Infrastructures (V2I) ou entre Véhicule et Véhicules (V2V). Afin de faciliter la qualité de vie dans les zones métropolitaines et urbaines, des stratégies de gestion complexes avec un contrôle du trafic à l'échelle du réseau sont nécessaires.

Avec l'urbanisation progressive et l'augmentation constante du nombre de véhicules, la congestion du trafic devient insoutenable dans les domaines urbains [Pi et al., 2019]. Elle a de graves effets négatifs sur la société moderne et l'environnement tels que la pollution atmosphérique due à la marche au ralenti des véhicules, l'augmentation du bruit, les problèmes de santé induits et le ralentissement du développement économique [de Souza et al., 2015]. Les autorités sont donc conscientes de l'importance de fournir des solutions efficaces aux problèmes de circulation routière. Elles tentent de les résoudre par le déploiement de systèmes de transport intelligents capables de détecter les embouteillages et de proposer des solutions pour soulager ou éviter ces problèmes.

Les différentes solutions utilisées pour la gestion du trafic peuvent être classées en trois catégories :

1. La première repose sur l'utilisation des capteurs, alors que la majorité des méthodes de détection conventionnelles nécessitent des équipements fixes, tels que les capteurs pour contrôler la vitesse des véhicules, calculer la vitesse moyenne sur une période donnée, puis la comparer à un seuil prédéfini pour estimer l'état de la route et déterminer le niveau de congestion [Cherrett et al., 2001] [Nguyen et al., 2014].
2. Pour la deuxième catégorie, un certain nombre de solutions basées sur les

communications entre véhicules ont été proposées pour la détection des embouteillages. En fait, Vehicular Adhoc Networks (VANETs) constituent une technologie prometteuse qui permet la communication entre véhicules afin de collecter et d'agréger en temps réel des informations de la vitesse et de la position des véhicules individuels [Zhang et al., 2011].

3. La troisième catégorie concerne la surveillance visuelle du trafic, qui constitue une solution attrayante et rentable puisque les caméras sont moins chères et fournissent des séquences vidéo de haute qualité.

## 1.3 Détection et Classification des Véhicules

### 1.3.1 Détection des véhicules

La détection des véhicules joue un rôle important dans la surveillance du trafic. Elle consiste à détecter les véhicules présents dans les séquences vidéo en se basant sur plusieurs caractéristiques. Les processus de détection des véhicules sur route sont généralement utilisés pour le comptage des véhicules, le calcul de la vitesse moyenne, le suivi des véhicules, l'analyse du trafic et la reconnaissance du type de véhicule. Ils peuvent être mis en œuvre dans différents environnements. Comme le montre la Figure 1.2.

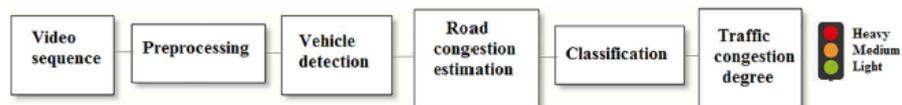


FIGURE 1.2 – Détection du la trafic routier

Notons que le défi majeur de cette phase est de séparer les véhicules de l'arrière-plan. La précision de la détection des véhicules a une grande importance pour le suivi des véhicules et peut affecter la pertinence des informations à déduire.

Les approches de détection de véhicules, illustrés dans la Figure 1.3, basées sur le traitement vidéo peuvent être divisées en trois catégories : les approches basées sur l'apparence, les approches basées sur le mouvement et les approches basées sur les objets.

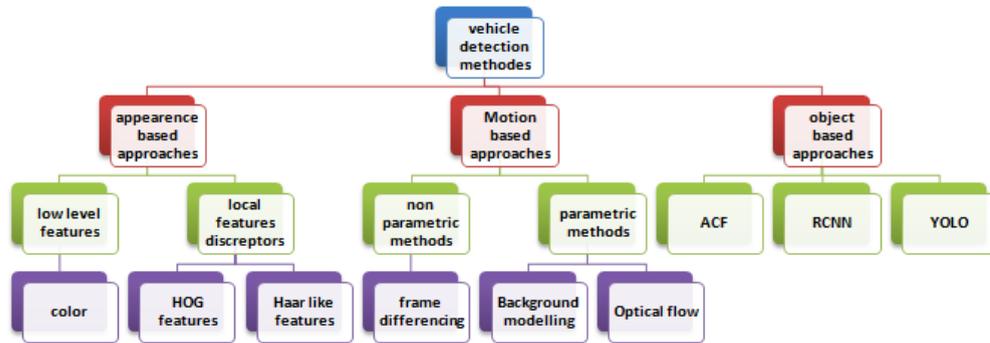


FIGURE 1.3 – Les principales approches de détection de véhicules

### Approches basées sur l'apparence

Dans les méthodes de détection des véhicules basés sur l'apparence, les caractéristiques des véhicules sont nécessaires, telles que la couleur [Tsai et al., 2007], la forme [Van Leeuwen and Groen, 2005], les bords [Betke et al., 2000], l'ombre [Ibarra-Arenado et al., 2017], la symétrie [Bensrhair et al., 2002], les coins [Bertozzi et al., 1997], la texture [Bucher et al., 2003] et la lumière du véhicule [Eichner and Breckon, 2007]. Les caractéristiques utilisées dans les approches basées sur l'apparence peuvent être divisées en deux sous-catégories. La première fait référence aux caractéristiques de bas niveau telles que la symétrie, la couleur, le bord et l'ombre. Ces caractéristiques sont puissantes et très utiles pour l'extraction des informations sur les véhicules. La seconde est celle des descripteurs de caractéristiques locales tels que les Histogram of Oriented Gradient (HOG) [Dalal and Triggs, 2005], les caractéristiques de type Harr [Viola and Jones, 2001]. Ces caractéristiques sont robustes et permettent de détecter rapidement et efficacement des objets dans les images.

### Approches basées sur le mouvement

Les véhicules sur la route sont régulièrement en mouvement, ce qui introduit des effets de mouvement relatif [Sivaraman and Trivedi, 2013]. Les méthodes de cette catégorie utilisent des séquences vidéo pour détecter les véhicules en fonction de leur mouvement, ce qui permet de les distinguer de l'arrière-plan sans aucune connaissance préalable [Barnich and Van Droogenbroeck, 2010]. La détection des véhicules basés sur le mouvement peut être divisée en deux sous-catégories. La première correspond aux méthodes non paramétriques telles que la différenciation de frame [Kim and Kim, 2003, Li et al., 2002] qui est considérée comme étant la plus simple ap-

proche de détection de mouvement. Son principe consiste à distinguer le premier plan de l'arrière-plan pixel par pixel sans inclure aucun paramètre. La deuxième sous-catégorie est celle des méthodes paramétriques telles que la modélisation du fond [Shehata et al., 2019] et le flux optique [Zhan and Ji, 2011].

Les modèles basés sur le mouvement ont joué un rôle important dans la détection des véhicules, ils visaient à réduire l'impact des arrière-plans dynamiques dans les images vidéo, comme le balancement des arbres et le battement des drapeaux [Zhang et al., 2007].

### Approches basées sur les objets

Les approche basées sur les objets nécessitent trois étapes pour la détection des véhicules : *(i)* la segmentation, *(ii)* l'apprentissage, *(iii)* la validation. La première étape consiste à extraire les objets du nombre total d'images. Ensuite, elle procède à l'optimisation des paramètres de segmentation. Enfin, la validation est réalisée sur des échantillons de test pour calculer la précision. Plusieurs méthodes dépendent de différents détecteurs d'objets basés sur l'apprentissage, tels qu'Aggregate Channel Features (ACF) [Byeon and Kwak, 2017], Region-Based Convolutional Neural Network (RCNN) [Girshick et al., 2014], FastRCNN, FasterRCNN [Ren et al., 2015], et You Only Look Once (YOLO)[Redmon et al., 2016].

### 1.3.2 Classification des véhicules

Le système de classification des véhicules a fait l'objet de recherches actives. De nombreuses méthodes ont été proposées en utilisant différentes approches et capteurs. Ces méthodes peuvent être classées en fonction des capteurs utilisés dans le système de classification à savoir les capteurs laser, les capteurs magnétiques et les capteurs de vision.

- Les capteurs laser ont la plus grande fiabilité car ils permettent de récupérer le profil tridimensionnel du véhicule comme information pour la tâche de classification du véhicule [Buch et al., 2008] avec une fiabilité et une précision très élevées. Cependant, les capteurs laser sont très coûteux par rapport aux capteurs basés sur la vision.
- Un capteur magnétique peut être utilisé pour extraire des caractéristiques dominantes de faible complexité, notamment le nombre de véhicules, la vitesse,

[Kaewkamnerd et al., 2009]. Une évaluation a montré que le résultat lorsque la classification est basée sur les tailles retourne une précision de (95%). Mais lorsque la classification est effectuée sur des véhicules de taille similaire le résultat est de 70% à 80%.

- Le système de classification des véhicules basé sur la vision utilise une caméra à faible coût pour capturer des images ou des séquences d'images de la scène d'un véhicule. Ensuite, le système de classification récupère les informations de l'image en se basant sur l'algorithme utilisé. Par conséquent, Les performances d'un système de classification de véhicules basé sur la vision dépendent en grande partie de l'algorithme et non du capteur. Les capteurs basés sur la vision peuvent être basés sur l'image ou vidéo. Dans de nombreux systèmes conventionnels de classification de véhicules basés sur la vision, les caractéristiques du véhicule utilisées pour la tâche de classification sont la hauteur, la longueur et la largeur du véhicule [Shi et al., 2007]. Des techniques plus sophistiquées ont adopté la détection des points clés sur les véhicules comme caractéristiques pour la tâche de classification [Ma and Grimson, 2005]. Ces techniques présentent un avantage en raison de la grande fiabilité des algorithmes de détection des points clés invariants à l'échelle.

## 1.4 La Détection et la Classification des Véhicules : État de l'Art

Dans cette section, nous passons en revue les approches les plus significatives et les plus représentatives dans le domaine de la détection et la classification de véhicules.

[Dong et al., 2015] ont recommandé une technique de Semi-Supervised Convolutional Neural Network CNN pour la classification des véhicules basée sur la vue frontale du véhicule (voir la Figure 1.4). Cependant, les caractéristiques formées par le CNN sont trop biaisées pour fonctionner en images tramées. Dans leur méthode, le réseau neuronal convolutif prend une image originale du véhicule comme entrée et les sorties la probabilité de chaque type de véhicule auquel le véhicule appartient. Le réseau contient deux blocs, et chaque bloc se compose de couche convolutionnelle, de couche de rectification absolue, de couche de normalisation du contraste local, de couche de regroupement moyen et de couche de sous-échantillonnage. La couche

convolutionnelle calcule les convolutions entre l'entrée et un ensemble de filtres. La couche de rectification absolue et la couche de normalisation du contraste local effectuent une transformation non linéaire sur la sortie de la couche convolutionnelle. La couche de mise en regroupement moyen et la couche de sous-échantillonnage réduisent la résolution spatiale de la représentation pour obtenir la robustesse à la fois aux distorsions géométriques et aux petits décalages.

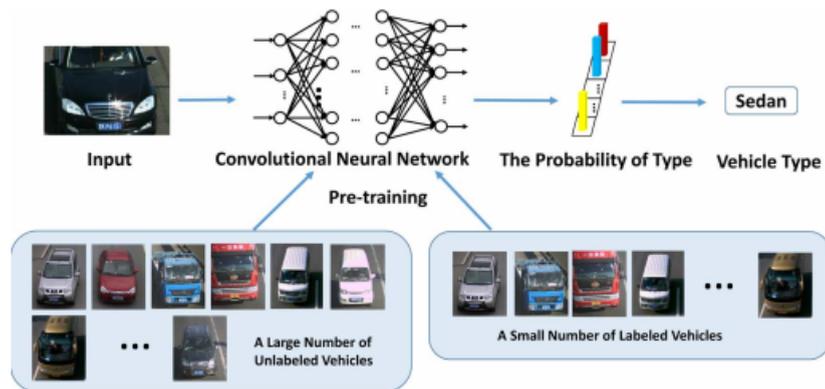


FIGURE 1.4 – diagramme de l'approche proposée dans [Dong et al., 2015]

L'architecture de ce réseau est comme le montre la Figure 1.5.

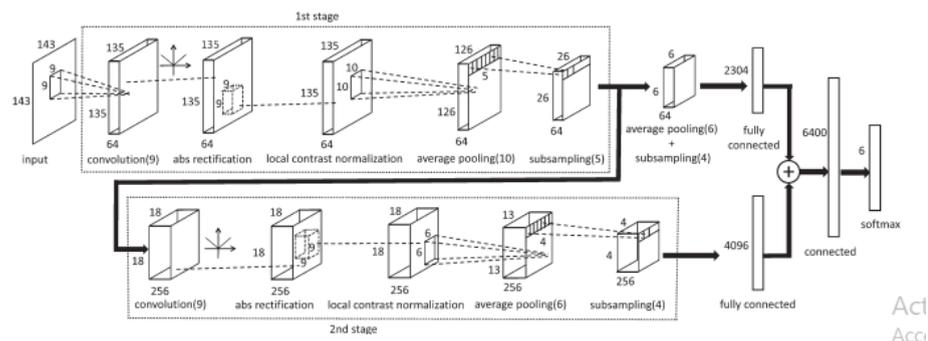


FIGURE 1.5 – Le réseau neuronal convolutif de classification des types de véhicule proposé dans [Dong et al., 2015]

Le processus de détection et de classification des véhicules en fonction d'une zone de détection virtuelle a été suggéré par [Seenouvang et al., 2016], qui comprend l'extraction de premier plan, la détection, l'extraction de caractéristiques et la classification. Un modèle Gaussian Mixture (GMM) est utilisé pour la détection des véhicules et certaines opérations sont également effectuées pour obtenir les objets de premier plan et la classification est faite, en utilisant le classificateur K-Nearest

Neighbor (KNN). Dans leur méthode, la soustraction de l'arrière-plan est la première étape pour détecter les véhicules en mouvement dans chaque image d'une vidéo. La Figure 1.6 présente des exemples d'images de circulation en entrée et les résultats du GMM après soustraction de l'arrière-plan.



FIGURE 1.6 – Extraction du véhicule : (a), (b) échantillon de l'image d'entrée (c), (d) échantillon de l'image d'extraction d'avant-plan [Seenouvong et al., 2016]

La classification du véhicule est effectuée sur la base de l'algorithme K-NN. L'idée est de rechercher les  $K$  données d'apprentissage les plus proches. Dans leur expérience,  $K$  est fixé à 5. Les caractéristiques des véhicules seront utilisées pour classer un véhicule en trois catégories : véhicules de petite taille (bicyclette, moto), de taille moyenne (voiture, camionnette, petit camion) et de grande taille (gros camion, bus). Cinq vidéos d'entrée (d'une durée de Sept à Neuf minutes) ont été utilisées pour tester la méthode proposée. Toutes les vidéos de circulation ont été enregistrées par une caméra numérique qui a été placée sur un survol de 9h à 17h. La résolution de chaque image dans les vidéos de circulation était de 30 images par seconde. Les vidéos ont été redimensionnées de  $1280 \times 720$  pixels à  $320 \times 180$  pixels. La moitié de chaque vidéo a été utilisée pour former le classifieur K-NN. Un nombre réel de

véhicules dans cinq vidéos d'entrée est indiqué dans le tableau 1.1. Les symboles S, M, et L e désignent respectivement les véhicules de petite, moyenne et grande taille. [Audebert et al., 2017] ont conféré un segment avant l'approche de détection en

TABLE 1.1 – Nombre réel de véhicules dans cinq vidéos

Vidéo $N^{\circ}$	Types de véhicules			Total
	S	M	L	
1	31	121	4	156
2	26	100	7	133
3	16	125	4	145
4	24	134	4	162
5	20	105	3	128
Total	117	585	22	724

utilisant des techniques de Deep Learning. La segmentation, suivie de la détection et de la classification de plusieurs variantes de véhicules à roues est testée pour les images de télédétection à haute résolution. Dans leur méthode, ils introduisent un pipeline de segments avant détection en trois étapes pour effectuer l'extraction et la classification des véhicules dans les données de télédétection à très haute résolution (VHR) sur les zones urbaines. Leur méthode se compose de trois parties, illustrées dans la Figure 1.7 :

1. Segmentation sémantique pour déduire les masques de classe au niveau des pixels en utilisant un fully convolutional network ;
2. Détection de véhicules par régression des boîtes englobantes des composants connectés ;
3. Classification au niveau de l'objet avec un réseau convolutional neural network ;

[Tsai et al., 2018] ont proposé une architecture optimisée de Réseau Neuronal Convolutif basée sur des algorithmes d'apprentissage profond pour la détection des véhicules et le système de classification utilisé pour les applications de transport intelligent. PVANET, en tant que réseau de base, est sélectionné et amélioré par un réglage fin pour obtenir une meilleure précision. Les auteurs ont optimisé un modèle de détection des véhicules basé sur une architecture R-CNN plus rapide. Ils ont réalisé le modèle de détection de véhicules proposé sur le GPU pour les applications en temps réel. Pour différentes classes de cibles et conditions météorologiques, le

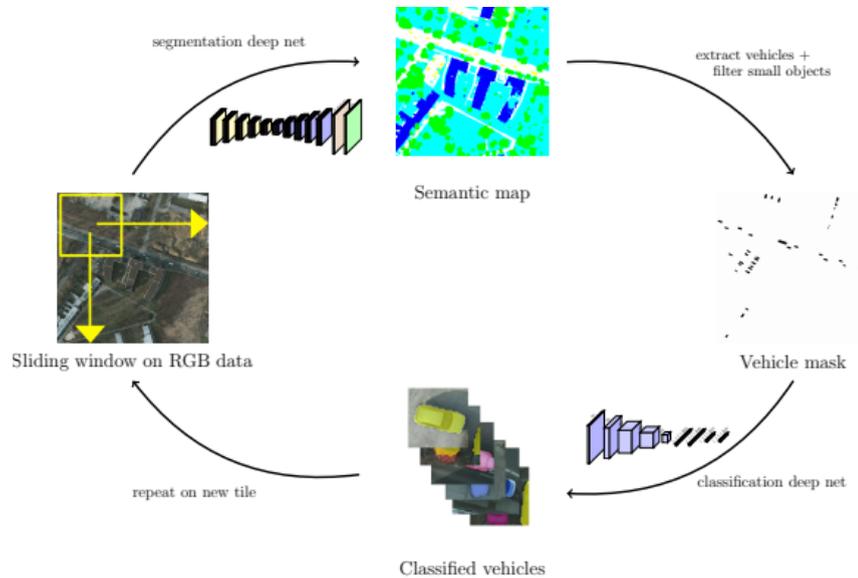


FIGURE 1.7 – Illustration du pipeline de segmentation avant détection, la détection et la classification propose dans [Audebert et al., 2017]

modèle proposé a permis d’atteindre un taux de détection supérieur à 90%. Quelques résultats de détection sont montrés dans la Figure 1.8



(a) Résultat de la détection d’une route (jour) (b) Résultat de la détection d’une route (nuit)

FIGURE 1.8 – Résultat de détection de véhicules (jour et nuit) [Tsai et al., 2018]

Dans la même année, [Velazquez-Pupo et al., 2018] ont présenté un modèle basé sur l’analyse de la vision avec une caméra fixe pour la surveillance du trafic, la détection des véhicules qui comprend le traitement des occlusions, le comptage, le suivi et la classification. Même si le meilleur classificateur est le SVM, ils ont quand même signalé que l’OC-SVM avec RBF Kernel a fourni des meilleurs résultats avec

une performance élevée.

Ils ont présenté un système de détection, de suivi et de classification des véhicules à partir de séquences vidéo, dont les performances sont supérieures à celles des méthodes utilisées dans la littérature. La Figure 1.9 montre le schéma fonctionnel du système. Dans l'apprentissage, les modèles de chaque classe de véhicules sont générés, pour cela, une vidéo d'apprentissage est utilisée. Avec les modèles, la classification est effectuée à l'aide de l'OC-SVM.

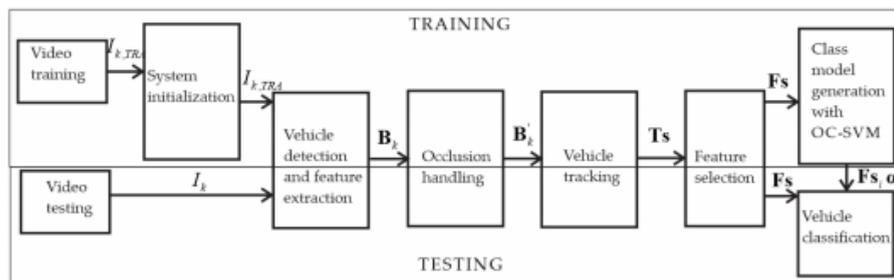


FIGURE 1.9 – Schéma fonctionnel du système proposé dans [Velazquez-Pupo et al., 2018]

[Arinaldi et al., 2018] ont présenté un système d'analyse vidéo du trafic basé sur des techniques de vision par ordinateur. Le cœur de ce système est la détection et la classification des véhicules pour lesquels ils ont développé deux modèles, le premier est un système basé sur des mélanges de Gaussienne pour la phase de détection du véhicule et le SVM pour la phase de classification (voir la Figure 1.10). Le second est basé sur Faster RCNN, une architecture d'apprentissage profond populaire pour la détection d'objets dans les images.

Dans leur expériences, les auteurs utilisent deux ensembles de données différents de vidéos de circulation. L'ensemble de données indonésien sur les centres de péages routiers a été pris sur deux importantes routes à savoir le péage de Jagorawi et celui de Kapuk (voir la Figure 1.11).

La Figure présente les résultats de la détection et de la classification des véhicules de Faster RCNN de détection et de classification des véhicules pour des données de nuit. La nuit représente un défi pour le détecteur de véhicules basé sur les MoG, en raison des lumières vives et des diverses réflexions dans la scène de trafic.

[Wang et al., 2019] ont développé un système de détection et de classification des véhicules en mouvement appelé "Improved Spatio-Temporal Sample Consensus",

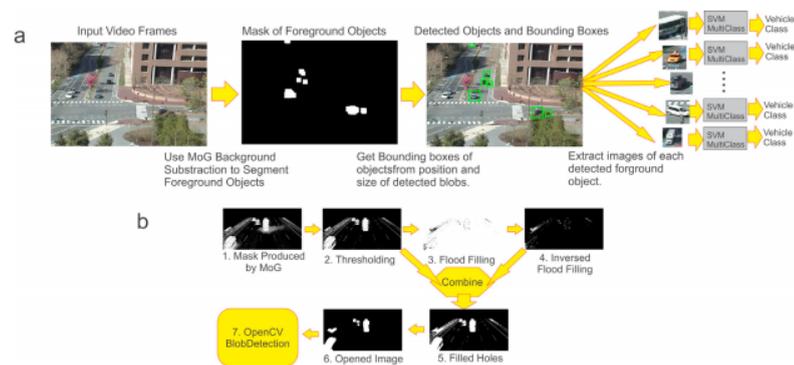


FIGURE 1.10 – Schéma de la méthode MoG+SVM pour la détection et la classification des véhicules [Arinaldi et al., 2018]



FIGURE 1.11 – (a) Scène de péage TMII; (b) Scène de péage Kapuk; (c) Scène de trafic MIT; (d) Classes de véhicules de péage indonésiens; (e) Classes de véhicules de trafic MIT [Arinaldi et al., 2018]

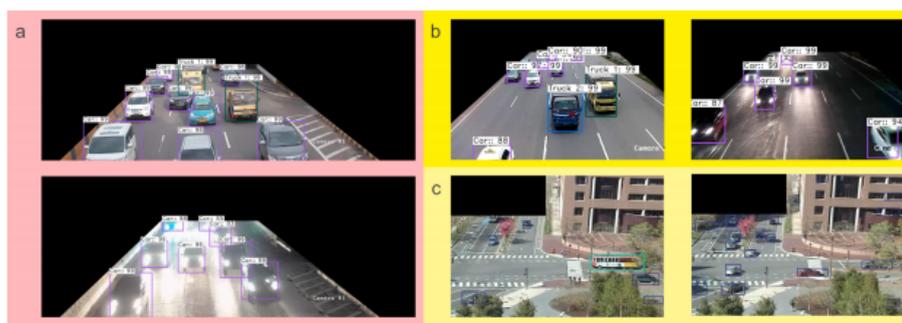


FIGURE 1.12 – (a) Résultats sur la scène de péage de Kapuk; (b) Résultats sur la scène de péage du TMII; (c) Résultats sur la scène de trafic du MIT

comme le montre la Figure 1.13. Tout d'abord, les véhicules en mouvement sont identifiés à l'aide de l'algorithme Spatio Temporal Sample Consensus, à partir de

l'intrusion de la variation de luminosité et de l'ombre des véhicules. En outre, grâce à des techniques de fusion de caractéristiques, les objets sont classés en fonction des caractéristiques de la surface de face, de la plaque d'immatriculation et de la symétrie du véhicule. La classification de l'objet en mouvement est faite par la fusion multi-fonctionnalité.

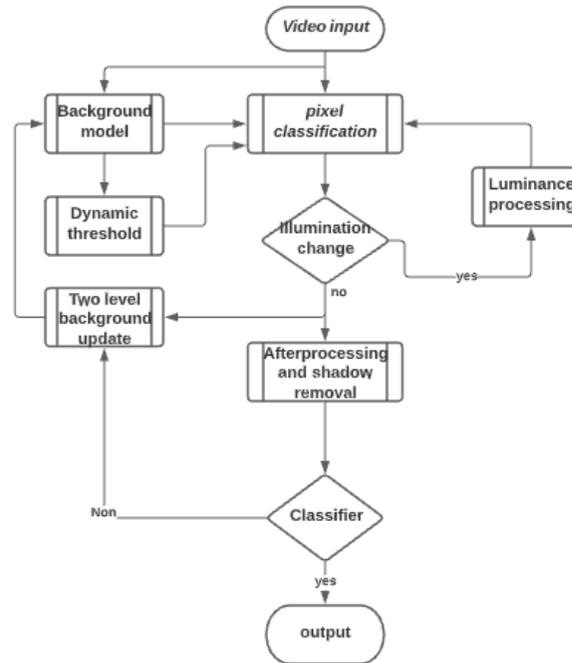


FIGURE 1.13 – Organigramme de l'algorithme STSC amélioré [Wang et al., 2019]

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté la détection et la classification des véhicules. Nous avons commencé par le système intelligent de trafic routier, puis les techniques traditionnelles de détection qui peuvent être divisées en modèles basés sur l'apparence, le mouvement et l'objet. Ensuite, nous avons donné une idée générale sur la classification des véhicules. Enfin, nous avons présenté une brève revue de l'état de l'art les approches les plus significatives et représentatives dans le domaine de la détection et de la classification des véhicules ont été présentées.

# Chapitre 2

## L'Intelligence Artificielle pour la Classification des Véhicules

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>23</b>
<b>2.2</b>	<b>L'intelligence Artificielle (IA)</b>	<b>23</b>
<b>2.3</b>	<b>Les applications de l'intelligence artificielle</b>	<b>24</b>
<b>2.4</b>	<b>Machine Learning (ML)</b>	<b>24</b>
2.4.1	L'apprentissage supervisé	25
2.4.2	L'apprentissage non supervisé	25
2.4.3	L'apprentissage semi supervisé	25
<b>2.5</b>	<b>Les principaux Algorithmes du Machine Learning (ML)</b>	<b>26</b>
2.5.1	K-Nearest Neighbor(K-NN)	26
2.5.2	Naive Bayes (NB)	26
2.5.3	Support Vector Machine (SVM)	27
2.5.4	Deep Learning (DL)	27
<b>2.6</b>	<b>Conclusion</b>	<b>28</b>

---

## 2.1 Introduction

Nous vivons dans l'ère des données, où tout autour de nous est connecté à une source de données, et tout ce qui fait notre vie est enregistré numériquement . Par exemple, le monde électronique actuel regorge de divers types de données, telles que l'internet des objets , les données des villes intelligentes, les données des entreprises, les données des Smartphones, les données des médias sociaux, les données de santé, les données COVID-19, et bien d'autres encore. Les types de données du monde réel et les techniques du Machine Learning augmente de jour en jour. L'extraction d'informations à partir de ces données peut être utilisée pour construire diverses applications intelligentes dans les domaines concernés. L'intelligence artificielle , et en particulier le machine learning ont connu une croissance rapide ces dernières années dans le contexte de l'analyse des données et de l'informatique qui permet généralement aux applications de fonctionner de manière intelligente .

Cette partie de notre travail concerne principalement l'intelligence artificielle et les principaux algorithmes d'un machine learning utilisés dans le domaine de la classification.

## 2.2 L'intelligence Artificielle (IA)

L'IA est un programme informatique, ou un robot qui a la capacité d'apprendre et de s'améliorer pour résoudre des problèmes, Il existe des nombreuses définitions de l'IA, les voici :

- L'IA est la capacité d'une machine à imiter un comportement humain intelligent.
- L'IA est une branche de l'informatique traitant de la simulation du comportement intelligent par ordinateur.
- L'IA est un système informatique capable d'effectuer des tâches qui requièrent normalement l'intelligence humaine , telles que la perception visuelle, la reconnaissance vocale, la prise de décision et la traduction entre les langues.

Henry Brighton a divisé l'IA en 2 formes : l'IA forte et l'IA faible.

L'IA forte, appelée aussi Intelligence Générale Artificielle (AGI) est une forme de machine intelligente qui peut effectuer toutes sortes de tâches comme un humain normal. L'IA faible est une forme d'IA plus faible que l'AGI, qui peut résoudre

certains problèmes spécifiques comme un humain normal, mais pas toutes.

## 2.3 Les applications de l'intelligence artificielle

Les applications de l'IA connaissent un essor rapide dans un certain nombre de secteurs où il est possible de détecter des modèles dans de grands volumes de données et de modéliser des systèmes complexes pour améliorer la prise de décision et réduire les coûts dans :

- ★ le secteur des transports, utiliser l'IA pour les aider à résoudre les problèmes de circulation, utiliser des caméras de circulation pour la détection et la classification des véhicules et calculent la densité du trafic sur la route
- ★ le domaine de santé publique, les systèmes d'IA permettent de diagnostiquer et de prévenir les maladies et les épidémies à un stade précoce, de découvrir des traitements, de proposer des interventions adaptées et d'alimenter les outils d'auto-surveillance.
- ★ les applications de sécurité numérique utilisent des systèmes d'IA pour aider à automatiser la détection des menaces et la réponse à celles-ci est de plus en plus en temps réel.
- ★ les applications de l'IA en agriculture comprennent la surveillance de la santé des cultures et des sols et la prévision de l'impact des facteurs environnementaux sur le rendement des cultures
- ★ les services financiers s'appuient sur l'IA pour détecter les fraudes, évaluer la solvabilité, réduire les coûts du service à la clientèle, automatiser les transactions et soutenir la conformité juridique.
- ★ en matière de marketing et de publicité, l'IA exploite les données sur le comportement des consommateurs pour cibler et personnaliser le contenu, la publicité, les produits et les services, les recommandations et les prix.

## 2.4 Machine Learning (ML)

ML est un sous-ensemble de l'intelligence artificielle qui traite de la conception d'algorithmes, la construction des modèles et la prise de décisions sur la base des ensembles de données d'entrée. Les modèles ML classent les données à l'aide d'une

matrice de caractéristiques. Les algorithmes d'un ML permettent de construire des modèles pour classer les données avec plus de précision. Ils ont été récemment utilisés pour résoudre de manière optimale une variété de problèmes techniques et scientifiques.

Les techniques de ML peuvent être classées en 3 principales catégories

### 2.4.1 L'apprentissage supervisé

Le terme "apprentissage supervisé" fait référence aux algorithmes qui apprennent à partir d'un ensemble de données de l'apprentissage et qui sont formés pour superviser le processus d'apprentissage. Les classes utilisées dans l'apprentissage supervisé sont connues, ce qui implique que toutes les données d'entrée sont étiquetées.

L'apprentissage supervisé est généralement utilisé lorsque nous souhaitons effectuer une classification (associer des étiquettes d'entrée à des étiquettes de sortie) ou une régression (associer des étiquettes d'entrée à des étiquettes de sortie continues).

### 2.4.2 L'apprentissage non supervisé

Dans l'apprentissage "non supervisé", les données d'entrée ne sont pas catégorisées. Sans connaître les étiquettes, l'algorithme d'apprentissage non supervisé modélise la distribution des données d'entrée [Bhatnagar et al., 2017]. L'algorithme doit définir et classer les données d'entrée avant de déterminer les modèles et la structure cachés. Ces algorithmes sont regroupés en groupes de clustering et d'association :

- ★ Clustering : Trouver des regroupements inhérents aux données d'entrée.
- ★ Association : Trouver diverses associations entre différents groupes de données d'entrée.

### 2.4.3 L'apprentissage semi supervisé

L'apprentissage semi-supervisé peut être défini comme une hybridation des méthodes supervisée et non supervisée puisqu'il fonctionne sur des données étiquetées et non étiquetées [Chen et al., 2011, Sarker et al., 2020]. Ainsi, il se situe entre l'apprentissage "sans supervision" et l'apprentissage "avec supervision". Dans le monde réel, les données étiquetées peuvent être rares dans plusieurs contextes, et les données non étiquetées sont nombreuses où l'apprentissage semi-supervisé est utile [Mohammed et al., 2016]. L'objectif ultime d'un modèle d'apprentissage semi-supervisé est

de fournir un meilleur résultat pour la prédiction que celui produit en utilisant uniquement les données étiquetées du modèle. Parmi les domaines d'application dans lesquels l'apprentissage semi-supervisé est utilisé comprend la traduction automatique, la détection des fraudes, l'étiquetage des données et la classification des textes.

## 2.5 Les principaux Algorithmes du Machine Learning (ML)

De nombreux algorithmes de classification ont été proposés dans les littérature sur l'apprentissage automatique et la science des données. Dans ce qui suit, nous résumons les méthodes les plus courantes et les plus populaires qui sont largement utilisées dans divers domaines d'application.

### 2.5.1 K-Nearest Neighbor(K-NN)

Le K-Nearest Neighbor(K-NN) est l'un des algorithmes les plus anciens, les plus simples et les plus précis pour la classification des modèles. Le K-NN a été proposé par [Fix and Hodges, 1951] puis modifié par [Cover and Hart, 1967], et a été identifié comme l'une des 10 meilleures méthodes dans l'exploration de données [Wu et al., 2008]. Par conséquent, le K-NN a été étudié au cours des dernières décennies et largement appliqué dans de nombreux domaines. Ainsi, le K-NN constitue le classifieur de base dans de nombreux problèmes de classification tels que la reconnaissance de formes, la catégorisation de textes, les modèles de classement, la reconnaissance d'objets et la reconnaissance d'événements.

### 2.5.2 Naive Bayes (NB)

Un classifieur naïf Bayes est un classifieur probabiliste simple basé sur l'application du théorème de Bayes avec des hypothèses d'indépendance fortes (naïves). Un terme plus descriptif pour le modèle de probabilité sous-jacent serait "modèle de caractéristiques indépendantes".

En termes simples, un classificateur naïf Bayes suppose que la présence (ou l'absence) d'une caractéristique particulière d'une classe n'est pas liée à la présence (ou à l'absence) d'une autre caractéristique. Selon la nature précise du modèle de probabilité, les classifieurs de Bayes naïfs peuvent être entraînés très efficacement dans un cadre

d'apprentissage supervisé. Dans de nombreuses applications pratiques, l'estimation des paramètres pour les modèles de Bayes naïfs utilise la méthode du maximum de vraisemblance. En dépit de leur conception naïve et de leurs hypothèses apparemment trop simples, les classificateurs naïfs Bayes ont assez bien fonctionné dans de nombreuses situations complexes du monde réel. En 2004, l'analyse du problème de la classification Bayésienne a montré qu'il existe certaines raisons théoriques pour l'efficacité apparemment déraisonnable des classifieurs naïfs Bayes [Zhang, 2004]. Pourtant, une comparaison exhaustive avec d'autres méthodes de classification en 2006 a montré que la classification de Bayes est surpassée par des approches plus courantes, telles que les forêts aléatoires [Caruana and Niculescu-Mizil, 2006].

### 2.5.3 Support Vector Machine (SVM)

Le SVM est un algorithme d'apprentissage automatique supervisé qui prend un ensemble connu de données en entrée. Il construit un modèle pour l'entrée connue. Le SVM est l'algorithme de ML le plus robuste conçu par [Cortes and Vapnik, 1995]. Il considère chaque caractéristique comme un point dans un espace à  $n$  dimensions et classe les données en deux classes en utilisant un hyperplan.

Le but du SVM est de trouver la séparation maximale entre deux classes. Les points de données qui se trouvent sur le bord du plan sont appelés Support Vector. Deux règles de base régissent la sélection :

**Étape 1** : sélectionnez l'hyperplan qui sépare les données en deux classes distinctes.

**Étape 2** : maximiser la distance entre les points de données les plus proches de l'une ou l'autre classe, de sorte qu'aucun point de données ne soit mal classé.

### 2.5.4 Deep Learning (DL)

DL est une technique de ML qui permet d'extraire des caractéristiques à partir de données telles que des images, des textes ou des sons. Les caractéristiques extraites par cette technique sont apprises et peuvent être utilisées pour les tâches suivantes. La principale différence entre le fonctionnement du ML et celui de l'apprentissage profond est illustrée par la Figure 2.1.

Alors que dans ML, la façon de faire une estimation correcte doit être enseignée à l'algorithme en fournissant plus d'informations. En d'autres termes, l'extraction des caractéristiques est effectuée par l'homme dans le ML, tandis que le modèle d'apprentissage profond le fait par lui-même.

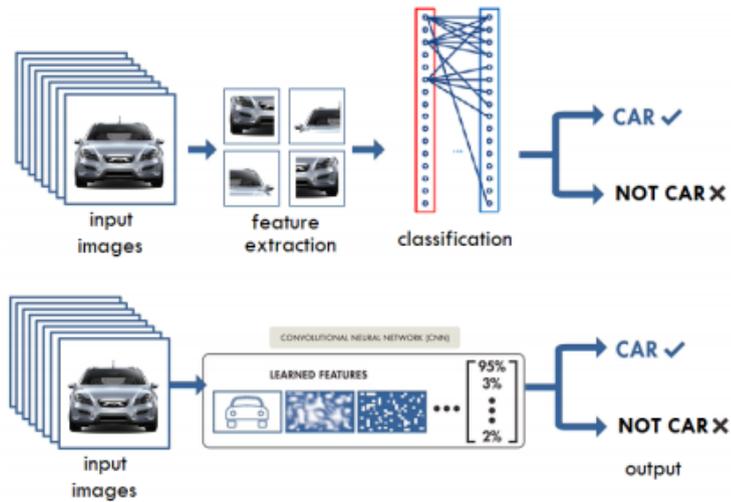


FIGURE 2.1 – Différence entre ML et DL

Bien que DL soit un nouveau domaine du ML, il a connu un grand développement en peu de temps et est devenu un sujet important dans le domaine de l'intelligence artificielle.

## 2.6 Conclusion

Dans ce chapitre, nous avons présenté l'intelligence artificielle et ses applications dans des beaucoup de domaines. Par la suite, nous avons donné une idée générale sur le machine learning et ses catégories. Nous avons commencé par l'apprentissage supervisé, l'apprentissage non supervisé, et semi supervisé. Nous avons terminé par les principaux algorithmes que nous avons mentionnés tels que K-Nearest Neighbor, naïfs Bayes, Support Vector Machine et Deep Learning.

# Chapitre 3

## Détection et Classification des Véhicules

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>31</b>
<b>3.2</b>	<b>Approche Proposé</b>	<b>31</b>
<b>3.3</b>	<b>Traitement d'images</b>	<b>31</b>
3.3.1	Conversion d'une image RGB en niveau de gris	32
3.3.2	Conversion du niveaux de gris en images binaires	32
3.3.3	Masque d'images	33
3.3.4	Filtre Gaussien	33
3.3.5	Dilatation et érosion	33
3.3.6	Segmentation par seuillage	34
3.3.7	Détection de contour	34
<b>3.4</b>	<b>Histogram of Oriented Gradients (HOG)</b>	<b>35</b>
<b>3.5</b>	<b>K-Nearest Neighbor (KNN)</b>	<b>37</b>
<b>3.6</b>	<b>Support Vector Machine (SVM)</b>	<b>39</b>
3.6.1	Hyperplan non-linéaire	40
3.6.2	Utilisation des SVM avec des kernels	41
3.6.3	Support Vector Machine multiclass	43
<b>3.7</b>	<b>Évaluation</b>	<b>44</b>
3.7.1	Base de données	44
3.7.2	L'environnement de travail	45

3.7.3	Protocole d'évaluation . . . . .	48
<b>3.8</b>	<b>Conclusion . . . . .</b>	<b>50</b>

---

## 3.1 Introduction

La plupart des tâches de vision par ordinateur, telles que la détection et la classification des véhicules, considèrent l'extraction de caractéristiques comme la première étape du traitement. La détection et la classification de véhicules présente un problème difficile dans le domaine de la vision par ordinateur et de l'analyse d'images. Il s'agit d'un sujet important utilisé principalement pour la surveillance et la sécurité routière.

Au cours des années précédentes, de grands succès ont été obtenus dans le domaine de la détection et de la classification générale des véhicules à l'aide d'algorithmes d'apprentissage automatique.

Cette partie de notre travail s'intéresse principalement aux algorithmes principaux utilisés pour la classification tels que le SVM, le KNN et le descripteur HOG.

## 3.2 Approche Proposé

La classification est un problème de reconnaissance de formes, qui tente d'assigner une étiquette (label) à chacune des classes données. C'est exactement ce qu'est la classification automatique de véhiculés, attribuer un label classe à l'image d'un véhicule donné.

Avant que le système ne soit capable de faire la distinction entre les véhicules, il doit d'abord apprendre à quoi ressemblent ces derniers et quelles sont les caractéristiques permettant de différencier un véhicule d'un autre. Dans le processus d'apprentissage (voir Figure 3.1), communément appelé apprentissage (training), un ensemble d'images et une méthodologie pour en extraire les caractéristiques sont fournis au système. Une fois que le système ait appris à différencier les véhicules, on pourrait y introduire une image afin d'obtenir une prédiction de sa classe.

## 3.3 Traitement d'images

Le traitement d'images est une méthode permettant d'effectuer certaines opérations sur une image, afin d'obtenir une image améliorée ou d'en extraire des informations utiles. Il s'agit d'un type de traitement du signal dans lequel l'entrée est

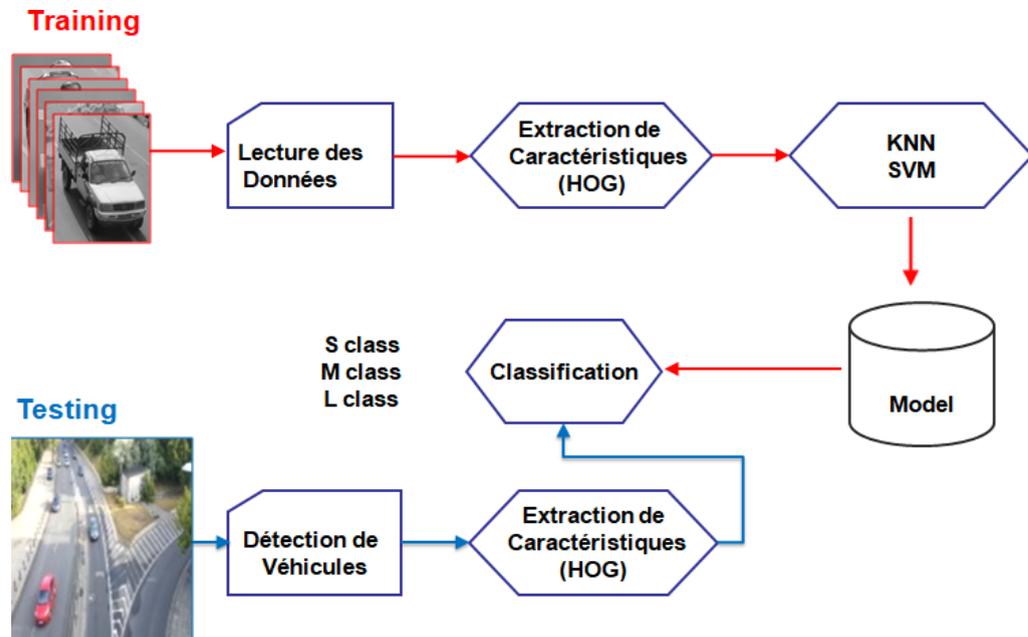


FIGURE 3.1 – Structure des systèmes de classification

une image et la sortie peut être une image ou des caractéristiques associées à cette image.

Dans cette section, la plupart des termes nécessaires seront discutés.

### 3.3.1 Conversion d'une image RGB en niveau de gris

Une image RGB est une image à trois composantes de couleur de taille  $M * N$  où les composantes sont le rouge (R), le vert (G) et le bleu (B). L'intensité de la couleur de chaque composante est représentée par un nombre de 8 bits.

Une image en niveaux de gris est simplement une image en noir et blanc où elle est représentée par une seule composante d'intensité. La couleur la plus claire étant le blanc et la plus sombre le noir. L'intensité de la couleur de l'image en niveaux de gris est représentée par un nombre de 8 bits où la valeur 0 signifie blanc et 255 signifie noir.

### 3.3.2 Conversion du niveaux de gris en images binaires

Les images binaires sont un type particulier d'image dont la valeur d'intensité n'a que deux valeurs possibles. L'image est constituée de deux couleurs, le noir et

le blanc, représentées numériquement par 0 et 1. La valeur d'intensité 0 signifie noir et 1 signifie blanc.

Les images binaires sont obtenues par le seuillage d'une image en niveaux de gris ou d'une image RGB en utilisant une valeur d'intensité appelée seuil. L'objectif de la binarisation est de séparer l'objet de l'image de fond ou la couleur de l'objet est blanche (couleur d'avant-plan), tandis que la couleur noire est appelée couleur d'arrière-plan.

### 3.3.3 Masque d'images

En traitement d'images, un noyau ou un masque est une petite matrice utilisée pour le filtrage, l'accentuation, la détection des contours, etc. Ceci est accompli en faisant une convolution entre un noyau et une image. Le masquage consiste à mettre à zéro certaines des valeurs des pixels d'une image, ou une autre valeur de fond.

### 3.3.4 Filtre Gaussien

Le filtre Gaussien est probablement le filtre le plus utilisé. Il est réalisé en convoluant chaque pixel de l'image d'entrée avec un noyau Gaussien avec des valeurs calculées avec l'équation 3.1 :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.1)$$

### 3.3.5 Dilatation et érosion

La dilatation et l'érosion sont deux opérations morphologiques courantes utilisées dans diverses tâches de vision par ordinateur et de traitement d'images. La dilatation et l'érosion sont toutes deux des opérations binaires. Elles prennent une image et un élément structurant (matrice de noyau binaire) en entrée et produisent une image en sortie.

L'opérateur de dilatation fonctionne en mettant en noir tous les pixels qui correspondent à la position des pixels du noyau translaté, ce qui a pour effet de fermer les trous dans l'image et de renforcer les petites caractéristiques. Un exemple est illustré dans la Figure 3.2

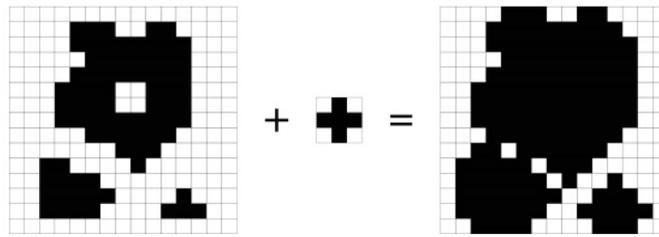


FIGURE 3.2 – Résultat de l’opérateur de dilatation utilisant un noyau  $3 \times 3$

L’opérateur d’érosion translate l’élément structurant en chaque pixel noir de l’image et vérifie à chaque position si tous les pixels actifs du noyau correspondent aux pixels noirs de l’image. Si c’est vrai, alors le pixel renvoyé est noir, sinon il est blanc. Un exemple de l’utilisation d’un tel opérateur est illustré à la Figure 3.3

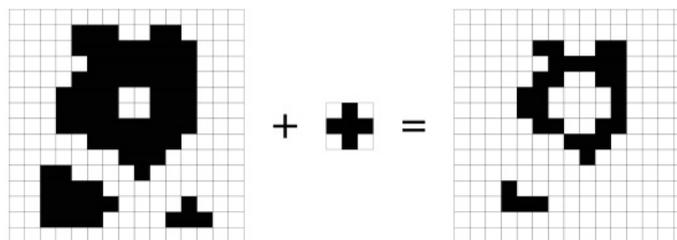


FIGURE 3.3 – Résultat de l’opérateur érosion utilisant un noyau croisé  $3 \times 3$

### 3.3.6 Segmentation par seuillage

Cette méthode permet de diviser une image en fonction des valeurs de l’histogramme. La méthode d’Otsu sélectionne le meilleur seuil en fonction de la séparabilité entre deux groupes de pixels. Elle est utilisée pour effectuer un seuillage automatique en supposant alors que l’image à binariser ne contient que deux classes, (Les objets et l’arrière-plan). La méthode d’Otsu calcule le seuil optimal  $T$  qui sépare ces deux classes afin que la variance intra-classe soit minimale et que la variance inter-classe soit maximale.

### 3.3.7 Détection de contour

Le contour peut être simplement expliqué comme étant la courbe fermée joignant tous les points continus (le long de la frontière), ayant la même couleur ou la même intensité. Les contours sont un outil utile pour l’analyse des formes et la détection

et la reconnaissance des objets.

L'idée principale sur laquelle repose la détection des contours est de trouver où il y a des changements brusques dans l'intensité d'une image. La dérivée première est utilisée pour trouver ces changements d'intensité en utilisant des opérateurs tels que Sobel, Prewitt et Roberts détaillés sur la Table 3.1.

Roberts		Prewitt				Sobel																																											
x	y	x			y			x			y																																						
<table border="1"><tr><td>-1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	-1	0	0	1	<table border="1"><tr><td>0</td><td>-1</td></tr><tr><td>1</td><td>0</td></tr></table>	0	-1	1	0	<table border="1"><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	-1	-1	-1	0	0	0	1	1	1	<table border="1"><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1	<table border="1"><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1	<table border="1"><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1
-1	0																																																
0	1																																																
0	-1																																																
1	0																																																
-1	-1	-1																																															
0	0	0																																															
1	1	1																																															
-1	0	1																																															
-1	0	1																																															
-1	0	1																																															
-1	-2	-1																																															
0	0	0																																															
1	2	1																																															
-1	0	1																																															
-2	0	2																																															
-1	0	1																																															

TABLE 3.1 – Opérateurs de détection de contours

### 3.4 Histogram of Oriented Gradients (HOG)

Les descripteurs HOG ont été introduits par N. Dalal et B. Triggs, [Dalal and Triggs, 2005] chercheurs à l'INRIA de Grenoble, à la conférence CVPR de juin 2005 dans leurs travaux sur la détection des piétons.

Le HOG est une méthode d'extraction de caractéristiques utilisée en vision par ordinateur pour la détection d'objets. Le descripteur HOG est basé sur le calcul d'histogrammes locaux de chaque orientation des gradients normalisés d'une image sur une grille dense. Une telle représentation répond de manière appropriée au véhicule, qui est un objet très articulé.

Le principe du HOG est de calculer un vecteur caractéristique global en divisant l'image en petites régions adjacentes appelées cellules et en calculant pour chaque cellule un vecteur caractéristique global. La fusion de ces hogs forme le descripteur HOG. Les étapes de calcul du HOG [Dalal et al., 2006] peuvent être résumées en trois étapes : extraction des gradients, histogramme et normalisation.

La première étape consiste à calculer les dérivées partielles  $Gr_x$  et  $Gr_y$  dans les directions  $x$  et  $y$  pour chaque pixel  $(x, y)$  de l'image  $I$  à l'aide des méthodes suivantes :

$$Gr_x(x, y) = I(x + 1, y) - I(x - 1, y) \quad (3.2)$$

et

$$Gr_y(x, y) = I(x, y + 1) - I(x, y - 1) \quad (3.3)$$

L'amplitude du gradient et l'orientation associée à chaque pixel  $(x, y)$  sont alors obtenues en utilisant :

$$M(x, y) = \sqrt{(Gr_x(x, y))^2 + (Gr_y(x, y))^2} \quad (3.4)$$

et

$$\theta(x, y) = \text{atan} \frac{Gr_y(x, y)}{Gr_x(x, y)} \quad (3.5)$$

Une fois les gradients calculés, l'espace des orientations possibles  $[0, 2\pi]$  est discrétisé en  $Nb$  intervalles appelés bins. L'image de gradient est ensuite divisée en régions spatiales rectangulaires appelées cellules. Dans chaque cellule, un histogramme est construit. L'histogramme est composé de  $Nb$  classes représentant les  $Nb$  intervalles d'orientations possibles et l'amplitude du gradient est accumulée pour chacune des classes.

La troisième étape est une normalisation des histogrammes. Les cellules sont regroupées en blocs. Chaque bloc est ensuite normalisé indépendamment des autres. Comme les blocs peuvent se chevaucher, une cellule peut participer plusieurs fois au vecteur du descripteur final de l'image. La dimensionnalité du vecteur caractéristique HOG est une fonction de la taille de l'image, des dimensions des blocs et des cellules. Elle varie de 144 à 1764 comme le montre le tableau 3.2 sur une image de taille  $64 \times 64$ . La Figure 3.4 montre une illustration des représentations HOG d'une image de véhicule avec  $Nb = 1$ , avec  $Nb = 9$  et différentes tailles de cellules et de blocs.

TABLE 3.2 – Paramètres du descripteur HOG

Taille des cellules	Taille du bloc	Dimension
(16,16)	(1, 1)	144
(16,16)	(2,2)	(324)
(8,8)	(1,1)	(576)
(8,8)	(2,2)	(1764)

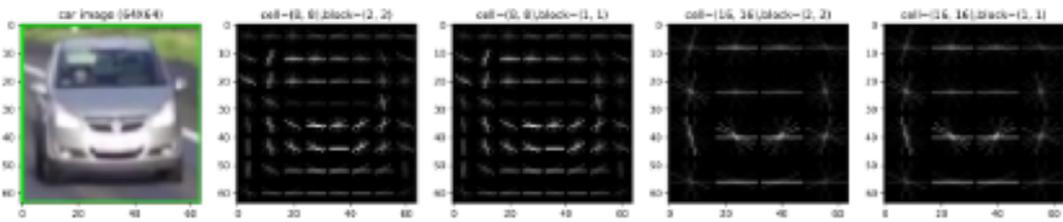


FIGURE 3.4 – Image d’un véhicule et de ses différentes représentations HOG

### 3.5 K-Nearest Neighbor (KNN)

L’algorithme de classification K-nearest neighbor est un algorithme d’exploration de données qui est théoriquement mature et peu complexe. L’idée de base est que, dans un espace d’échantillons, si la plupart de ses K-nearest neighbor appartiennent à une catégorie, alors l’échantillon appartient à la même catégorie. Le plus proche voisin se réfère au vecteur de caractéristiques unique ou multidimensionnel utilisé pour décrire l’échantillon le plus proche, et le critère de proximité peut être la distance Euclidienne du vecteur de caractéristiques.

La méthode classique du (kNN) est une méthode non paramétrique qui affecte un objet  $x$  inconnu à la classe à laquelle appartient la majorité de ses  $K$  voisins les plus proches. La Figure 3.5 illustre le processus de classification utilisant le KNN. Le cercle entoure les trois objets considérés pour  $K = 3$ , l’objet inconnu  $x$  est classé dans la classe 2 car deux de ses trois voisins appartiennent à la classe 2.

L’algorithme KNN fonctionne comme suit :

---

**Algorithm 1** Algorithme du plus proche voisins K-NN

---

- 1: Calculer la distance entre  $\vec{x} = (x_1, x_2, \dots, x_j)$  et tous les objets de  $Y$ , où  $Y$  est la matrice des objets d’apprentissage.
  - 2: Trier les objets selon les distances en ordre croissant et compter le nombre d’étiquettes de chaque classe se trouvant parmi les  $K$  premiers objets triés.
  - 3: Classifier dans la classe à laquelle la majorité de ses  $K$  plus proches voisins appartiennent.
- 

La distance Euclidienne est la plus courante [Fukunaga and Hostetler, 1973]-[Todeschini,

1989]. Elle est calculée comme suit :

$$d_E(\vec{x}, \vec{y}) = \sqrt{\sum_{j=1}^j (x_i - y_j)^2} \quad (3.6)$$

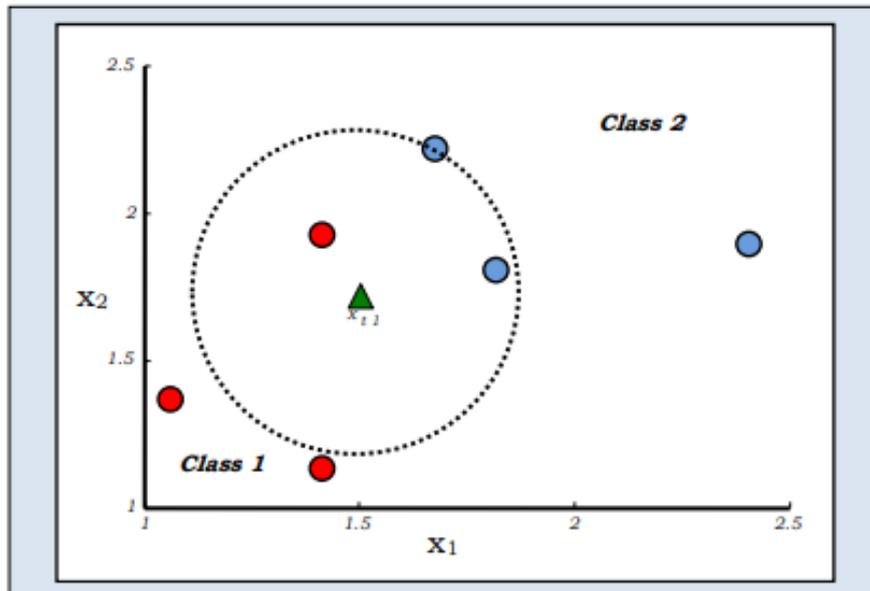


FIGURE 3.5 – Méthode de classification KNN avec  $K = 3$ .

Il existe plusieurs approches pour trouver le  $K$  approprié [Latourette, 2000]. La plus courante consiste à tester plusieurs valeurs de  $K$  par validation croisée [Duda et al., 2001] et de conserver le  $K$  qui donne le plus faible taux d'erreur de classification. Dans les problèmes de classification binaire, il est utile de choisir  $K$  comme un nombre impair, car cela évite les égalités. Ainsi, la valeur de  $K$  est définie de telle sorte qu'elle produise le taux de classification correcte le plus élevé [Bender and Kowalski, 1974].

### 3.6 Support Vector Machine (SVM)

Support vector machine (SVM) est l'un des méthodes de classification les plus efficaces. Il traite les fonctions linéaires et non linéaires à l'aide de différents noyaux (Kernel). Les SVM peuvent être utilisés avec des ensembles de données ayant un plus grand nombre de caractéristiques sans ajouter trop de complexité au système [Vural and Dy, 2004]. Le SVM qui classe les points de données en deux groupes est montré dans la Figure 3.6

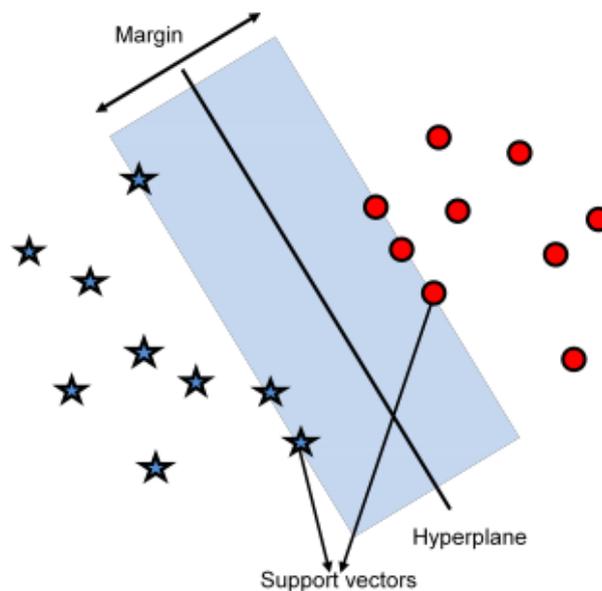


FIGURE 3.6 – Support Vector Machine

Dans la Figure 3.7, en sélectionnant l'hyperplan  $B$ , un point de données est mal classé. La Figure montre que l'hyperplan  $A$  est mieux adapté pour classer les données.

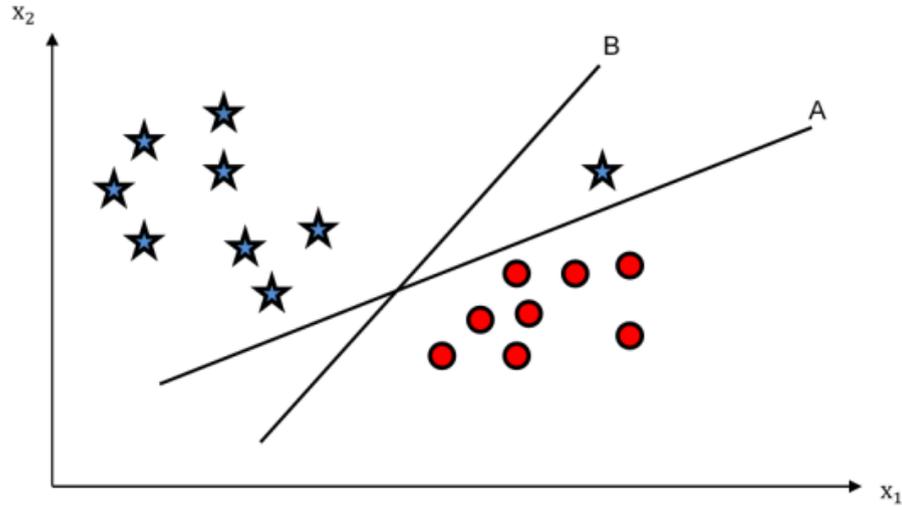


FIGURE 3.7 – Sélection d'hyperplans dans un espace de caractéristiques bidimensionnel.

### 3.6.1 Hyperplan non-linéaire

Pour un ensemble de données à deux dimensions, il est relativement facile de trouver un hyperplan. Dans ce cas, les noyaux utilisés pour générer les hyperplans, sont les fonctions mathématiques utilisées pour transformer des points bidimensionnels dans un espace à  $n$  dimensions. Une fois les points de données transformés dans un espace à  $n$  dimensions, un hyperplan est alors construit de manière à séparer les données en ensembles distincts Comme le montre la Figure 3.8.

Avec  $\vec{x}(x_1, x_2, \dots, x_n)$  et  $\vec{y}(y_1, y_2, \dots, y_n)$ , la fonction noyau utilisée pour générer l'hyperplan est de la forme :

$$x^2 + y^2 = z^2 \tag{3.7}$$

L'hyperplan correspondant est un cercle qui sépare les deux points de données sans erreur de classification.

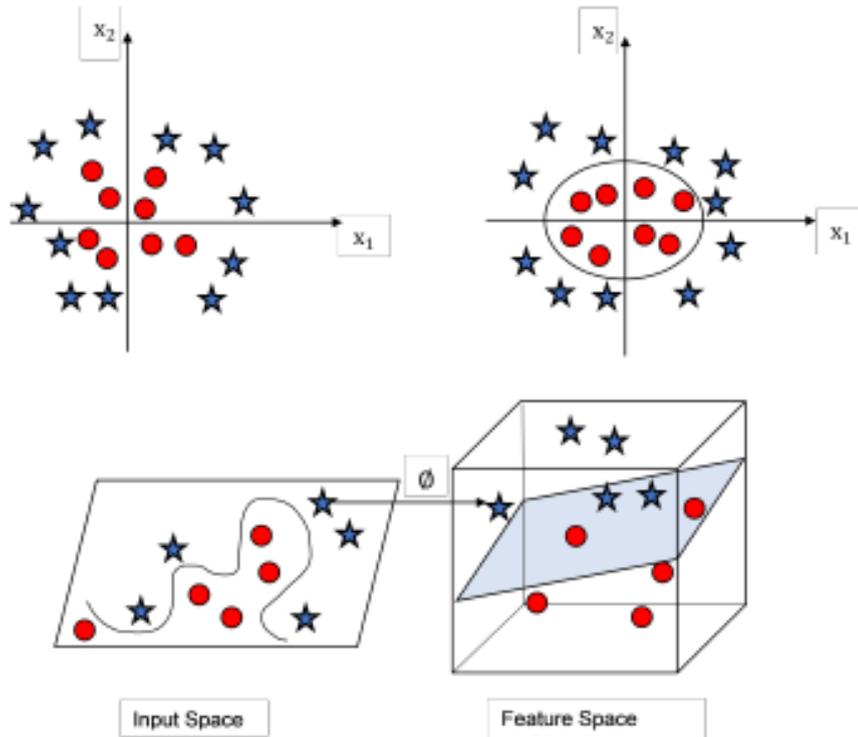


FIGURE 3.8 – Sélection d'un hyperplan dans un espace de caractéristiques à  $n$ -dimensions, où représente la cartographie

### 3.6.2 Utilisation des SVM avec des kernels

#### Kernel linéaire

Le SVM linéaire correspond à un noyau linéaire où

$$\theta(x) = x \quad (3.8)$$

et

$$\theta(x') = x' \quad (3.9)$$

Souvent utilisé pour la classification dans des problèmes d'optimisation linéaire, ce noyau est beaucoup plus rapide que les autres [Sánchez A, 2003]. Il est typiquement utilisé lorsque le nombre de caractéristiques est important et que la cartographie non linéaire ne permet pas d'améliorer les performances en raison de l'espace important des caractéristiques. Comme le montre la Figure 3.9, la limite de décision sépare les points de données correctement et incorrectement classés.

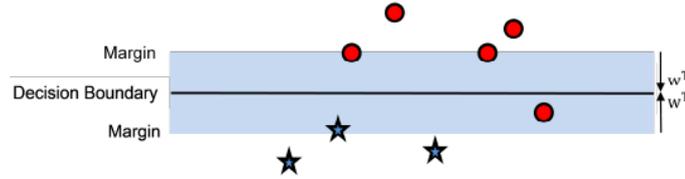


FIGURE 3.9 – Illustration du SVM avec un noyau linéaire

### Kernel polynomial

Le noyau polynomial utilise une fonction polynomiale pour les variables d'entrée afin de trouver la similarité des vecteurs d'apprentissage. Ces types de noyaux non linéaires sont largement utilisés pour les problèmes où les données d'apprentissage sont normalisées. Sa forme généralisée est définie comme suit :

$$\Theta(\vec{x}, \vec{y}) = (x^T y + 1)^p \quad (3.10)$$

Où  $p$  est le degré du polynôme et  $\Theta$  est la fonction noyau. Un noyau polynomial est souvent utilisé pour la reconnaissance visuelle des formes et les classifications du traitement du langage naturel. Le degré du noyau polynomial régit la flexibilité du classifieur résultant.

### Kernel Gaussian RBF

Le noyau Gaussian (RBF) est la fonction noyau la plus utilisée dans la classification SVM, Elle est utilisée lorsqu'il n'existe aucune connaissance préalable sur les données d'entrée [Liu and Parhi, 2016]. Il est défini comme :

$$\Theta(\vec{x}, \vec{y}) = \exp\left[-\frac{\|x - y\|^2}{2\sigma^2}\right] \quad (3.11)$$

Le paramètre  $\sigma$  joue le même rôle dans le noyau RBF Gaussian que  $p$  dans le noyau polynomial. Il contrôle la flexibilité du classifieur.

### 3.6.3 Support Vector Machine multiclass

Le SVM a été initialement conçu pour la classification binaire [Cortes and Vapnik, 1995]. Cependant, il peut être efficacement étendu à la classification multi-classe en décomposant le problème en une série de problèmes de classification binaire. Cette technique reste un domaine de recherche actif.

Les deux méthodes SVM multi classes les plus populaires sont le «one-to-rest» et le «one-to-one».

La méthode one-to-rest construit un nombre  $k$  de modèles SVM où  $k$  est le nombre de classes (La Figure 3.10). Le  $i^{eme}$  SVM avec  $1 \leq i \leq k$  est entraîné avec toutes les instances de la  $i^{eme}$  classe avec des étiquettes positives et le reste des instances avec des étiquettes négatives.

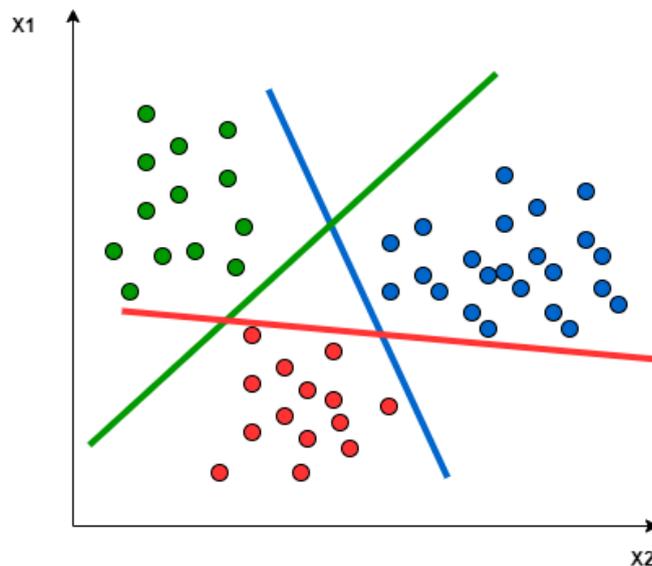


FIGURE 3.10 – La méthodes one-to-rest : Classification entre 3 classes

La méthode one-to-one (La Figure 3.11) a pour but de se débarrasser du problème de déséquilibre de la méthode one-to-rest en formant des classifieurs binaires uniquement avec les données appartenant à deux classes originales désignées par chaque classifieur. La méthode one-to-one construit  $k(k-1)/2$  étant donné que  $k$  est le nombre de classes [Knerr et al., 1990, Friedman, 1996].

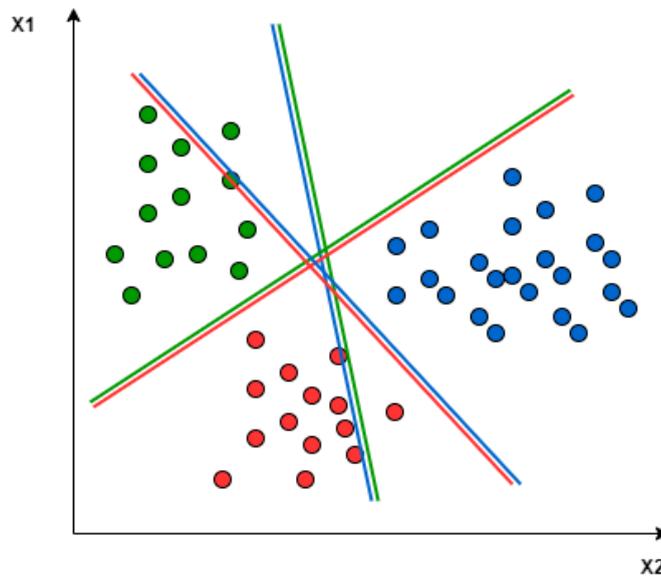


FIGURE 3.11 – La méthodes one-to-one : Classification entre 3 classes

## 3.7 Évaluation

### 3.7.1 Base de données

Dans tout système de classification, une bonne base de données est essentielle, il en va de même pour un système de classification de véhicules. La base de données a une importance majeure puisqu'elle est l'un des éléments responsables d'une haute performance de classification.

nous avons utilisé la base de données Nepal<sup>1</sup>, Cet ensemble de données d'images a été collecté dans le cadre d'un projet de premier cycle sur la détection des véhicules et la gestion de la congestion du trafic routier à l'aide du traitement des images. Un total de 30 vidéos de circulation, chacune d'environ 4 minutes, provenant de différentes rues de Katmandou ont été prises et les images des véhicules ont été manuellement découpées à partir des images vidéo.

Notre base de données contient 4727 images de véhicules sous plusieurs angles divisé en 3 classes : Small class illustrés dans la Figure 3.12 contenant 1815 images, Medium class illustrée dans la Figure 3.13 contenant 1461 images et Large class illustrée dans la Figure 3.14 comprenant 1451 images. L'ensemble de données utilisées dans ce travail contient des images de différentes tailles. Les algorithmes de classifi-

1. <https://www.kaggle.com/sdevkota007/vehicles-nepal>

cation attendent généralement des images d'entrée de taille égale. Par conséquent, toutes les images sont redimensionnées à  $(64 \times 64)$  pixels.

L'étape suivante consistait à diviser la base de données en un ensemble d'apprentissage et un ensemble de test avec répartition de 75% et 25% respectivement. Au final, il y a 3545 images constituant l'ensemble d'apprentissage (Training) et 1182 images constituant l'ensemble de test (Testing).

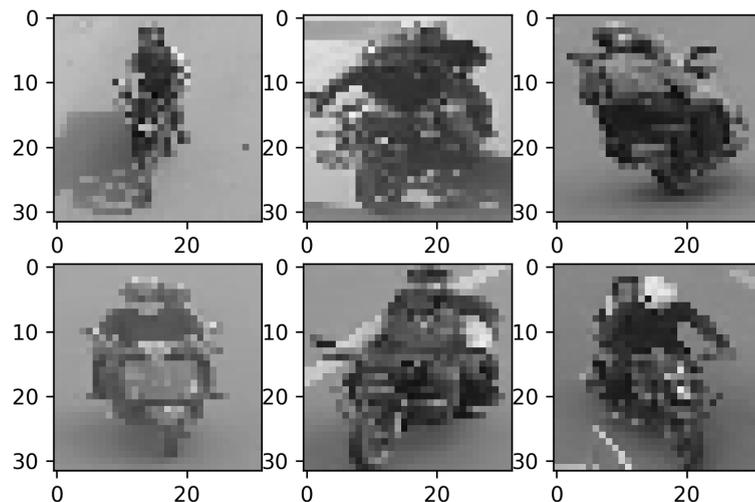


FIGURE 3.12 – small class images

### 3.7.2 L'environnement de travail

Avant d'obtenir les données, il est nécessaire de mettre en place l'environnement de travail du projet. Pour ce projet, nous proposons Google Colab .

Google Colaboratory (Colab) est un environnement Jupyter notebook gratuit qui améliore les compétences en langage de programmation Python. Créé comme un projet de recherche Google, il est basé sur le cloud et ne nécessite aucune installation. Colaboratory permet aux utilisateurs d'écrire et d'exécuter du code et d'accéder à des puissantes ressources informatiques gratuitement à partir du navigateur. Plus important encore, Colab fournit généreusement le GPU qui permet d'accélérer considérablement le processus d'apprentissage. Pour ces raisons, Colab est devenu très populaire parmi les amateurs du Deep Learning et du Data Science qui ne possèdent pas nécessairement un PC équipé de GPU coûteux.

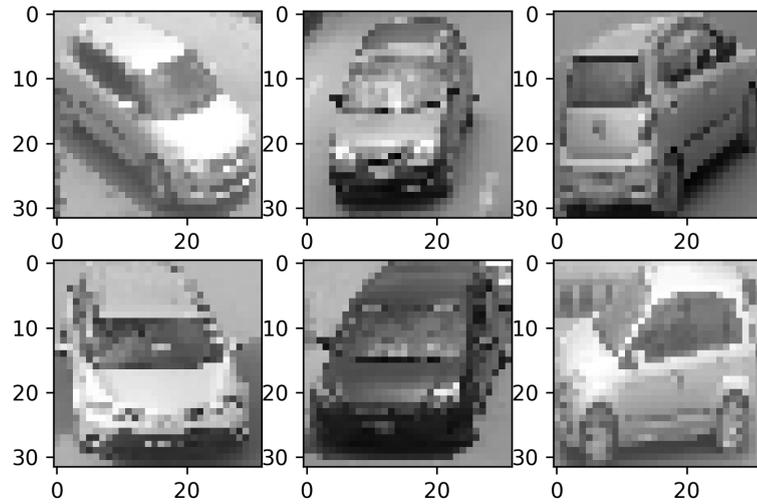


FIGURE 3.13 – medium class images

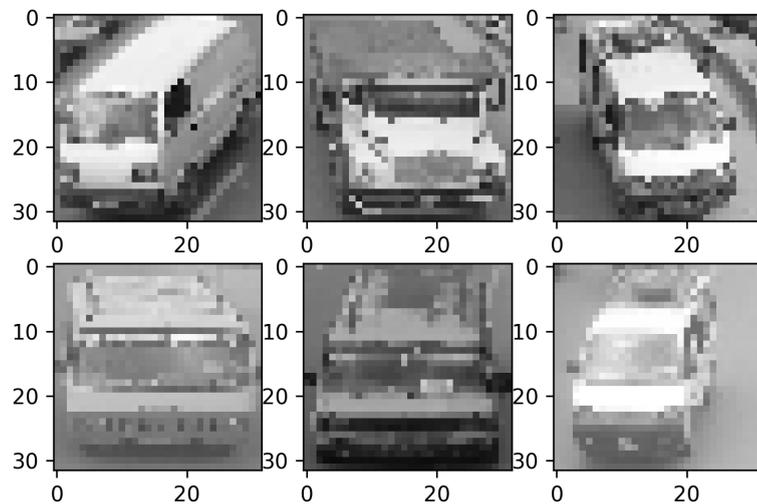


FIGURE 3.14 – large class images

Python est un langage de programmation de haut niveau largement utilisé, créé par Guido van Rossum à la fin des années 1980 [Chan, 2015]. Le langage met fortement l'accent sur la lisibilité et la simplicité du code, ce qui permet aux programmeurs de développer rapidement des applications. Comme tous les langages de programmation de haut niveau, le code Python ressemble à la langue anglaise que

les ordinateurs sont incapables de comprendre. La plupart des programmes en Python nécessitent considérablement moins de lignes de code pour effectuer la même tâche que d'autres langages tels que le C. Cela permet de réduire les erreurs de programmation et le temps de développement nécessaire. En outre, Python est livré avec une vaste collection de ressources qui permettent d'étendre les capacités du langage. Ainsi, Python peut être utilisé pour une grande variété de tâches, comme les applications de bureau, les applications de base de données, la programmation réseau, la programmation de jeux, le développement mobile et même la classification d'objets.

Pour notre projet, les bibliothèques Python populaires que nous avons utilisées sont :

- ♥ **Numpy** est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion plus facile des matrices de nombres, des fonctions sophistiquées
- ♥ **Open Source Computer Vision Library (Open CV)** est une bibliothèque offrant un ensemble de plus de 2500 algorithmes de vision par ordinateur spécialisés dans le traitement d'images, accessibles via des API pour les langages C, C ++ et Python.
- ♥ **Matplotlib** est une bibliothèque de traçage pour le langage de programmation Python. Elle fournit une API orientée objet pour intégrer des graphiques dans des applications utilisant des outils d'interface utilisateur graphique.
- ♥ **Scikit-Learn** également connu sous le nom de sklearn, est un package open source pour Python, utilisé pour l'exploration et l'analyse de données, l'apprentissage automatique et la science des données. Le paquetage sklearn propose divers algorithmes d'apprentissage automatique et des outils à des fins variées, telles que l'estimation, le prétraitement et le fractionnement.
- ♥ **Seaborn** est une bibliothèque de visualisation de données basée sur Matplotlib. Les graphiques statistiques, comme les cartes thermiques peuvent être construits facilement avec Seaborn.
- ♥ **Scikit-Image** est une bibliothèque open-source de traitement d'images pour Python. Elle comprend des algorithmes de segmentation, de transformations géométriques, de manipulation de l'espace couleur, d'analyse, de filtrage, de morphologie, de détection de caractéristiques, etc.

### 3.7.3 Protocole d'évaluation

#### matrice de confusion

La matrice de confusion est utilisée pour aider à dériver un ensemble de mesures de performance pour évaluer la performance des classificateurs. Toutes les métriques traditionnelles sont calculées sur la base de la matrice de confusion, à savoir les vrais positifs ( $TP$ ), les vrais négatifs ( $TN$ ), les faux positifs ( $FP$ ) et les faux négatifs ( $FN$ ).

$TP$ ,  $TN$ ,  $FP$  et  $FN$  identifient les enregistrements qui sont correctement détectés comme positifs, correctement détectés comme négatifs, incorrectement détectés comme positifs, et incorrectement détectés comme des points de données négatifs, respectivement. Dans le cas de la cybersécurité, le True Negative ( $TN$ ) est le nombre d'enregistrements qui sont correctement prédits comme étant de classe normale et le True Positive ( $TP$ ) est le nombre d'instances qui sont correctement prédites comme des attaques.

TABLE 3.3 – Matrice de confusion d'un problème à deux classes.

		Classe prédite	
		Positive	Negative
Classe réelle	Positive	True Positive $TP$	False Negative $FN$
	Negative	False Positive $FP$	True Negative $TN$

La précision est identifiée comme le rapport entre les classifications correctes et le nombre total d'échantillons, et définie par la formule suivante :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.12)$$

#### Résultat et discussion

Pour reconnaître une image, il faut d'abord en extraire les caractéristiques. Chaque caractéristique peut être étiquetée ou décrite et ces caractéristiques inconnues ou description sont comparées à des caractéristiques connues pour l'identification. Le résultat de cette comparaison fournira une reconnaissance complète de l'image.

Ce chapitre présente les résultats des expériences. Tout d'abord, l'algorithme SVM avec différents noyaux à savoir linéaire, polynomial et RBF sans descripteur de caractéristiques et le SVM avec les noyaux su-cités avec le descripteur HOG sont

entraînés et testés pour la classification des véhicules. Les résultats expérimentaux, présentés dans les tableaux 3.4 et 3.5 montrent que l’algorithme de classification de véhicules HOG-SVM proposé dans ce projet donne des taux de classification plus élevés que ceux obtenus par le SVM.

Nous avons effectué la même expérience en utilisant l’algorithme KNN et les résultats confirmaient le choix du HOG pour la description du véhicule et par conséquent la classification.

TABLE 3.4 – Taux de classification des algorithmes KNN et SVM avec le descripteur HOG

Algorithme ML	Taux de classification (%)
KNN(K=7)	95.68
SVM Linéaire	92.13
SVM RBF	95.68
SVM Polynomiale	95.09

TABLE 3.5 – Taux de classification des algorithmes KNN et SVM sans le descripteur HOG

Algorithme ML	Taux de classification (%)
KNN(K=7)	82.65
SVM Linéaire	82.99
SVM RBF	92.21
SVM Polynomiale	87.30

En effet, le classifieur SVM de type RBF et le KNN ont classé les images de véhicules avec le meilleur taux de classification atteignant 95.68%. La performance du classifieur SVM linéaire est nettement inférieure à celle des autres classifieurs avec un taux de 92.13%. Le classifieur SVM de type polynomial se situe entre les performances des classifieurs SVM RBF et linéaire enregistrant un taux de classification de 95.09%.

## Comparaison des performances des classifieurs

Les performances des classifieurs sont analysées à l'aide de l'ensemble de données des images de véhicule dans python. Les images suivantes spécifient la matrice de confusion pour les classifieurs SVM de types linéaire, polynomial et RBF et le classifieur KNN illustrés dans les Figures 3.15, 3.16, 3.17 et 3.18 respectivement.

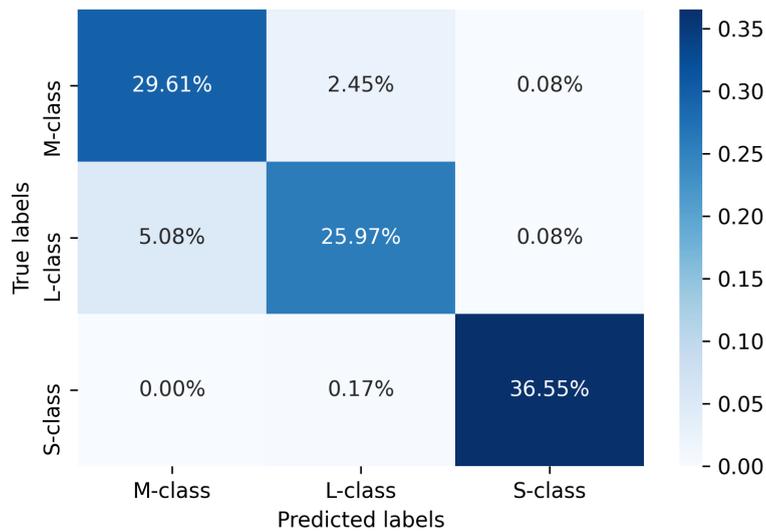


FIGURE 3.15 – La matrice de confusion du SVM- Linéaire-HOG

On remarque que l'algorithme SVM-RBF-HOG a très peu d'erreurs de classification comparé avec le SVM-Poly-HOG. A titre d'exemple, Le SVM-RBF-HOG classe les L-class comme M-class avec un taux de 2.62%, alors que le SVM-Poly-HOG enregistre un taux d'erreur de 3.13% sous les mêmes hypothèses. Les taux d'erreur de classification des algorithmes SVM-RBF-HOG et SVM-Linéaire-HOG sont comparables. Néanmoins, il faut noter que le SVM avec un noyau linéaire a une complexité plus inférieure que le SVM avec un noyau RBF.

## 3.8 Conclusion

Dans ce chapitre, nous avons abordé la méthodologie de notre recherche. Nous avons commencé tout d'abord par le descripteur HoG qui extrait les caractéristiques des images afin de les utiliser comme entrée pour entraîner notre modèle. Par la suite, nous avons donné une description des classifieurs KNN SVM.



FIGURE 3.16 – La matrice de confusion du SVM-poly-HOG



FIGURE 3.17 – La matrice de confusion du SVM-RBF-HOG

Enfin, Nous avons défini l'environnement de développement utilisé dans notre projet. Nous avons évalué le modèle et noté que la meilleure précision était obtenue en utilisant le SVM-RBF-HOG.

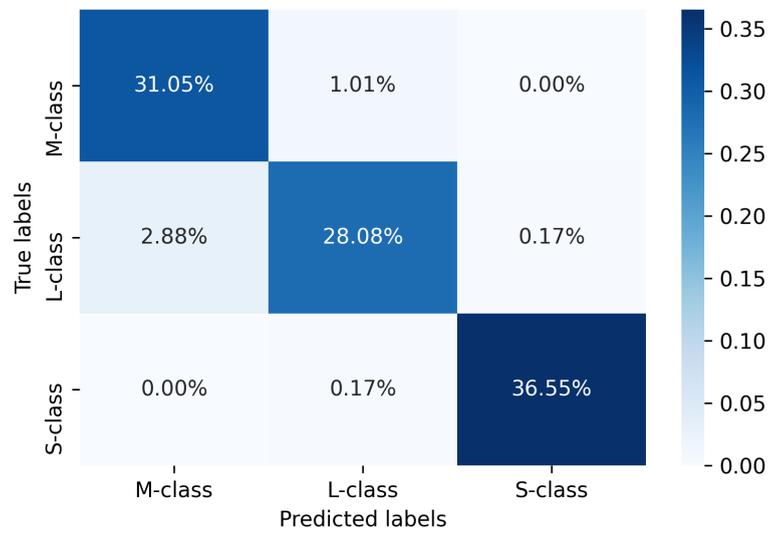


FIGURE 3.18 – La matrice de confusion du KNN-HOG

# Chapitre 4

## Détection et Classification dans un Flux Routier

### Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>54</b>
<b>4.2</b>	<b>Background Subtraction</b>	<b>54</b>
4.2.1	Les différents étapes de la soustraction d'arrière-plan	54
4.2.2	Classification des véhicules en utilisant HOG-RBF-SVM	57
<b>4.3</b>	<b>Présentation de la Raspberry Pi</b>	<b>59</b>
4.3.1	Raspbian	61
4.3.2	Configuration du système d'exploitation Raspbian sur les cartes Raspberry Pi	61
4.3.3	Programmation Python	62
<b>4.4</b>	<b>Conclusion</b>	<b>66</b>

---

## 4.1 Introduction

Dans les applications de surveillance visuelle, des caméras fixes ou des caméras Pan-Tilt-Zoom (PTZ) sont utilisées pour surveiller des activités sur des sites extérieurs ou intérieurs. Puisque la caméra est stationnaire, la détection d'objets en mouvement peut être réalisée en comparant chaque nouvelle image avec une représentation de la scène arrière plan. Ce processus est appelé background subtraction et la représentation de la scène est appelée modèle d'arrière-plan. La scène est ici supposée être stationnaire.

Généralement, le processus de soustraction de l'arrière-plan constitue la première étape des systèmes de surveillance visuelle automatisés ainsi que d'autres applications, telles que la capture de mouvement, l'analyse sportive, etc.

Les résultats de ce processus sont utilisés pour un traitement ultérieur. L'un des principaux avantages de la détection d'objets par soustraction de l'arrière-plan est le suivant : le résultat est une segmentation précise des régions de premier plan par rapport à l'arrière-plan de la scène.

Les sections suivantes de ce document exploreront les bases théoriques sur lesquelles notre solution a été construite.

## 4.2 Background Subtraction

La soustraction d'arrière-plan (Background subtraction) est une approche largement utilisée pour détecter les objets en mouvement dans les vidéos provenant de caméras statiques. La logique de cette approche est de détecter les objets en mouvement à partir de la différence entre l'image actuelle et une image de référence, souvent appelée image de fond ou modèle de fond.

Un algorithme de soustraction d'arrière-plan robuste doit être capable de gérer les changements d'éclairage et les changements de scène à long terme. Les analyses suivantes utilisent la fonction  $V(x, y, t)$  en tant que séquence vidéo où  $t$  est la dimension temporelle,  $x$  et  $y$  sont les variables d'emplacement spatial des pixels.

### 4.2.1 Les différents étapes de la soustraction d'arrière-plan

L'objectif de ce processus est la segmentation des objets d'avant-plan statiques et mobiles dans un flux vidéo. Cette tâche fournit un bon compromis en termes de

qualité de détection et de temps de calcul. Les différentes étapes de la soustraction de l'arrière-plan sont détaillées dans ce qui suit.

Afin de détecter les objets en mouvement à partir de la différence entre les images de vidéo et une image référence, nous devons d'abord estimer cette image de référence.

Dans ce travail, nous obtenons le modèle de l'arrière plan  $B$  en prenant l'image moyenne de l'ensemble des  $N$  images, selon l'équation 4.1 :

$$B(x, y) = \frac{1}{N} \sum_{n=1}^N I(x, y, n) \quad (4.1)$$

L'image soustraite  $I'$  du fond est obtenue selon l'équation 4.2.

$$I'(x, y) = I(x, y) - B(x, y) \quad (4.2)$$

La Figure 4.1a montre une image extraite d'une vidéo, la Figure 4.1b montre l'image de background calculée en utilisant le processus sus-cité et la Figure 4.1c montre le résultat de soustraction.

Les images obtenues sont généralement bruitées, soit à cause de la poussière sur les véhicules, soit à cause d'une compression bruyante, soit à cause de la poussière sur la caméra.

Nous utilisons un filtre Gaussien de noyau de taille  $9 \times 9$  pour filtrer l'image de différence  $I'$  (Figure 4.2). Nous avons ensuite utilisé le seuillage d'Otsu afin de binariser l'image en fonction de l'intensité des pixels (les négatifs à 255, puisque les véhicules en mouvement ont la luminance la plus faible, et tous les pixels positifs à 0) (Figure 4.3), .

L'image binaire présentée dans la Figure 4.3 est suffisamment connectée pour permettre une bonne segmentation et une bonne classification. Mais ce n'est pas toujours le cas lorsqu'il y a de grandes zones sur un véhicule avec de grandes variations d'intensité.

Il y'a deux opérations fondamentales, appelées érosion et dilatation. L'érosion rétrécit une image en agrandissant les trous et les espaces entre les pixels. La dilatation fait exactement le contraire, elle ajoute des pixels dans les vides, comblant ainsi tous les vides de l'image binaire. La combinaison des deux entraîne l'ouverture ou



(a) Image de vidéo



(b) Image de background



(c) Résultat de soustraction

FIGURE 4.1 – La différence entre l'image de vidéo et background

la fermeture d'une image, L'ouverture tend à éliminer les petits objets au premier plan, tandis que la fermeture élimine les petits trous au premier plan.

L'opération de fermeture est réalisée en effectuant d'abord une dilatation, suivie d'une érosion en utilisant le même élément structurant. Le résultat de l'opération de dilatation est illustré dans la Figure 4.4.

Un objet peut être localisé à partir de l'ensemble des pixels de son contour. De ce fait, la détection de contours est utilisée pour trouver les limites de régions. Un algorithme de remplissage est utilisé pour étiqueter tous les pixels chaque composante connectée contenant chaque pixel. Une boîte de délimitation est ensuite dessinée autour de tous les groupes connectés comme le montre le Figure 4.5.

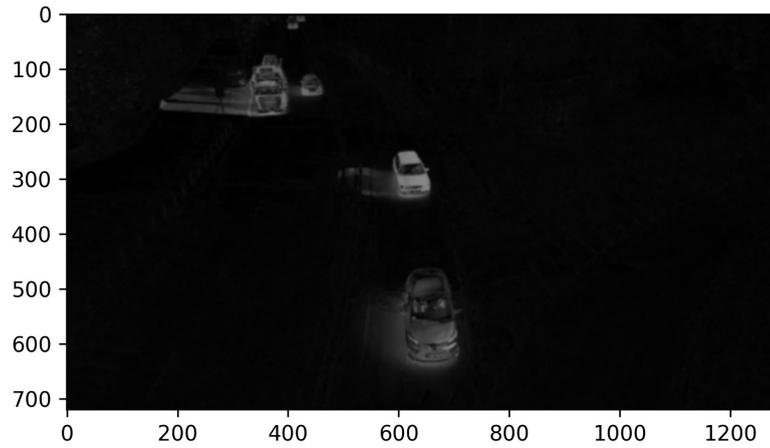


FIGURE 4.2 – Image filtrée par un noyau Gaussien

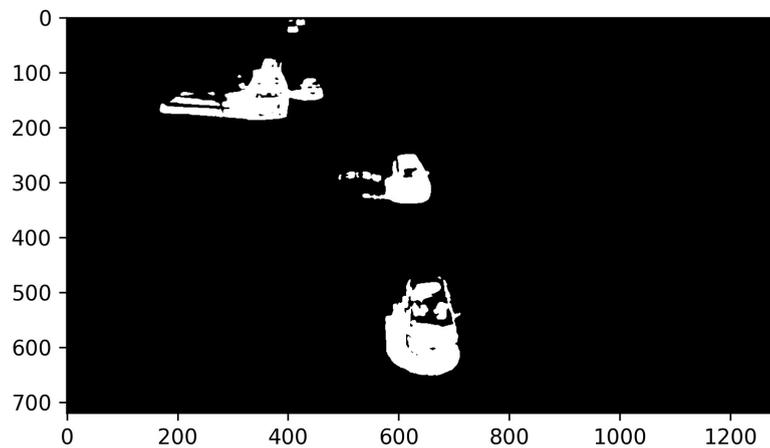


FIGURE 4.3 – Seuillage par la méthode d'Otsu

## 4.2.2 Classification des véhicules en utilisant HOG-RBF-SVM

Pour permettre l'implémentation de l'approche adoptée sur des scène réelles de trafic routier sur la Raspberry Pi4, nous avons procédé comme suit L'ajout d'un

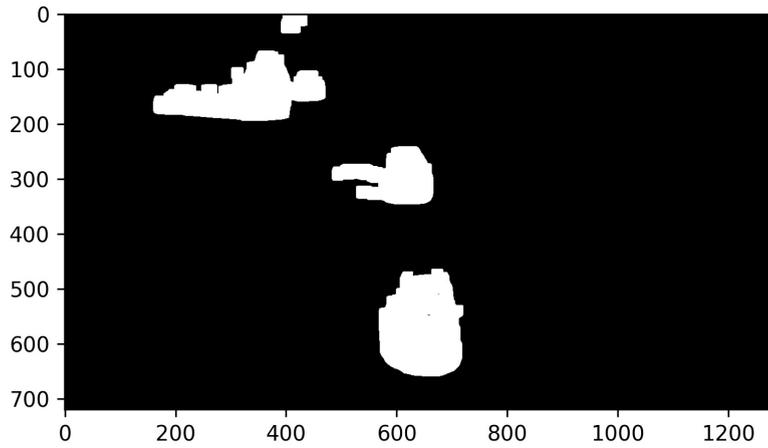


FIGURE 4.4 – Image après la dilatation

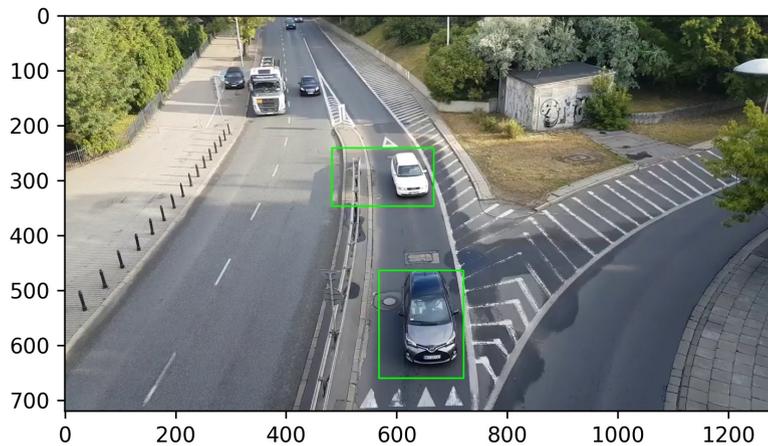


FIGURE 4.5 – Détection de véhicules dans une scène vidéo

plus grand nombre d'exemples à l'ensemble d'apprentissage augmente la précision du modèle. Selon Peter Norvig, nous n'avons pas de meilleurs algorithmes. Nous avons simplement plus de données.

Par conséquent, pour mieux s'assurer de l'efficacité du classifieur SVM qui a été préalablement entraîné avec un pourcentage de 75% et testé avec un pourcentage de

25% de la base de données et pour renforcer le modèle entraîné nous avons testé ce dernier avec un ensemble des données différent.

Pour ceci, nous avons adopté un protocole de qui consiste à prendre la totalité de la base de données pour l'apprentissage et évaluer le modèle appris sur une image de trafic sur la Raspberry Pi4.

Notons, nous avons utilisé le descripteur HOG sans tenir compte des composantes de couleur, car la couleur d'un véhicule ne nous renseigne pas beaucoup sur sa taille. Quelques résultats de classification des véhicules avec le nouveau modelé sont visualisés dans la Figure 4.6

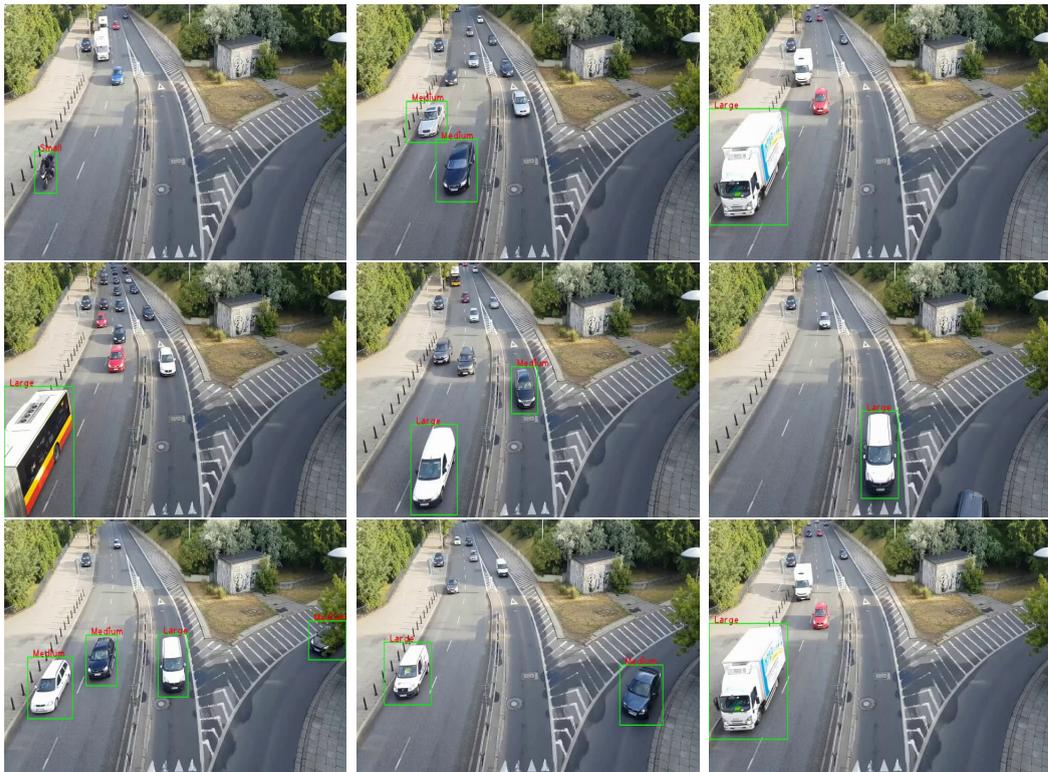


FIGURE 4.6 – Quelques résultats de classification avec le nouveau modèle

### 4.3 Présentation de la Raspberry Pi

Le Raspberry Pi est un mini-ordinateur que vous pouvez utiliser comme un ordinateur normal. Cette carte a été développée par la Fondation Raspberry Pi au Royaume-Uni. Actuellement, le Raspberry Pi fonctionne sur le dessus d'un processeur ARM. Le Raspberry Pi est conçu pour être de petite taille afin d'effectuer

certaines processeurs. Nous pouvons mettre des dispositifs de capteurs et d'actionneurs sur la carte Raspberry Pi, puis effectuer des mesures et des détections.

La première carte Raspberry Pi Model B est sortie en février 2012. Le dernier modèle, la carte Raspberry Pi 4 model B, est sorti en 2019. Certains modèles de Raspberry Pi fournissent également des piles réseau telles qu'Ethernet, WiFi et Bluetooth. Le Raspberry Pi peut également être connecté à un écran via un connecteur HDMI.

Différents modèles Raspberry Pi existent. Dans ce qui suit, nous allons présenter une comparaison entre les cartes les plus utilisées dans le tableau 4.1

TABLE 4.1 – Comparaison entre différents modèles de Raspberry

<b>Type de modele</b>	Pi 3 Model B	Pi 3 Model B+	Pi 4 Model B
<b>type de CPU</b>	ARM Cortex-A53	ARM Cortex-A53	ARM Cortex-A72
<b>RAM Size</b>	1GB LPDDR2	1GB LPDDR2	4GB LPDDR4
<b>WI-FI Intégré</b>	2.4GHz	5GHz	5GHz
<b>Gigabit Ethernet</b>	Non	Over usb2	Oui
<b>Bluetooth</b>	4.1 BLE	4.2BLU	5BLE
<b>USB</b>	4*USB2	4*USB2	2*USB3,USB2
<b>Video Decode</b>	H.264	H.264	H.265
<b>Video Encode</b>	H.264	H.264	H.264

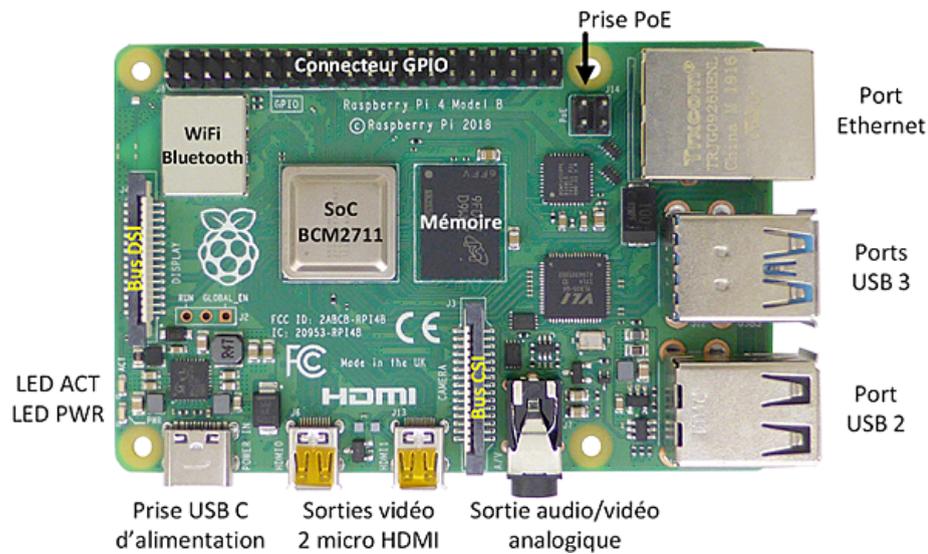


FIGURE 4.7 – L’architecture de la Raspberri Pi 4

### 4.3.1 Raspbian

Raspbian est un système d’exploitation open source basé sur Debian et optimisé pour le matériel Raspberry Pi. Ce système est initié par Mike Thompson et Peter Green. Lorsque la carte Raspberry Pi est commercialisée, Raspbian OS devient le système d’exploitation officiel des cartes Raspberry Pi, qui possède toutes les caractéristiques similaires à un ordinateur telles que le processeur, la mémoire et le pilote graphique.

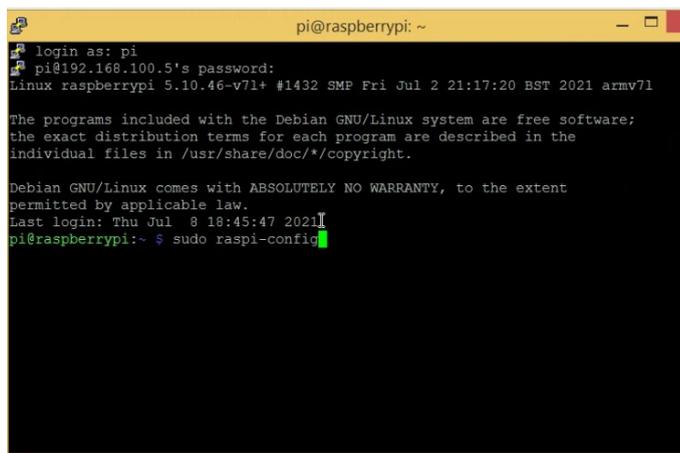
### 4.3.2 Configuration du système d’exploitation Raspbian sur les cartes Raspberry Pi

Les modèles Raspberry Pi n’ont pas de stockage interne à moins que vous n’utilisiez la Raspberry Pi Compute Module. Nous devrions attacher un stockage externe, comme une carte SD ou une carte microSD ou une clé USB. Pour configurer la Raspberry Pi, nous devons :

- ❖ **Téléchargez l’image du système d’exploitation** la dernière version de l’image Raspbian peut être téléchargée par le lien suivant <sup>1</sup>

1. <https://www.raspberrypi.org/software>

- ❖ **Gravez l’image du système d’exploitation sur la carte SD** La carte SD est insérée dans le lecteur de cartes du PC, puis l’image du système d’exploitation est gravée sur la carte SD en utilisant l’un des outils appropriés comme Win32Image ou Etcher. Lorsque cette opération est terminée, la carte SD est insérée dans la Raspberry Pi. Le clavier et la souris sont connectés aux ports USB du Raspberry Pi. Le câble HDMI est également connecté à la prise correspondante sur la Raspberry et l’écran.
- ❖ **Configurer le système** Le Raspbian peut être configuré en utilisant la commande `sudo raspi-config` à partir du terminal comme présenté dans la Figure 4.8. Un outil de Configuration de la Raspberry Pi (Figure 4.9) est lancé avec une navigation dans les différentes parties du menu.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.100.5's password:  
Linux raspberrypi 5.10.46-v7l+ #1432 SMP Fri Jul 2 21:17:20 BST 2021 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Jul 8 18:45:47 2021  
pi@raspberrypi:~ $ sudo raspi-config
```

FIGURE 4.8 – Menu de configuration Raspbian

### 4.3.3 Programmation Python

Un IDE (Integrated DeveLopment Environment) est un programme qui regroupe les fonctions fréquemment utilisées par un développeur. De nombreux IDE existent pour Python tels que Thonny considéré comme un IDE qui convient aux débutants.

Thonny est un IDE Python utilisé pour développer des programmes Python. On peut écrire des scripts Python et les enregistrer dans des fichiers `.py`. Il fournit également un outil d’exécution qui permet de voir la sortie du programme à partir d’un IDE. L’IDE Thonny est installé par défaut dans le bureau Raspbian Scratch (voir la Figure 4.10). Une fois lancé, Thonny offre une fenêtre avec une barre de menus usuels, une rangée de boutons, et deux panneaux correspondant à

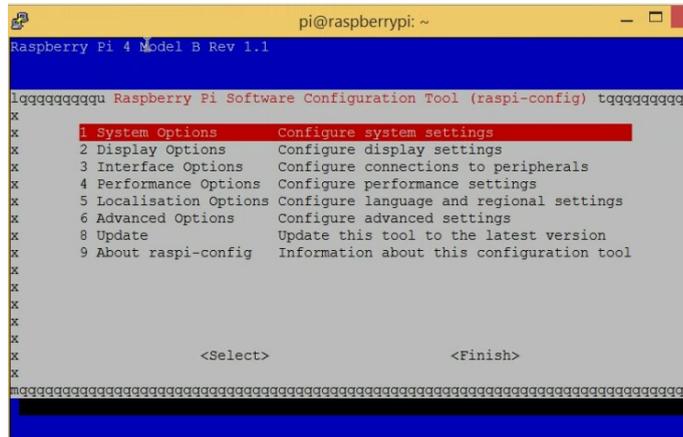


FIGURE 4.9 – Un outil de configuration de la Raspberry Pi4

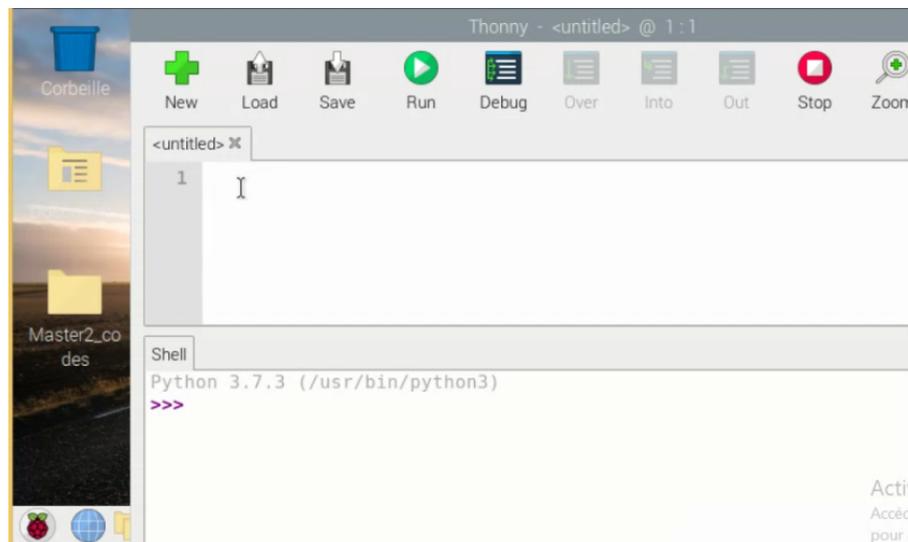


FIGURE 4.10 – Thonny IDE application

- l'éditeur (onglet nommé <untitled>)
- l'interpréteur (onglet nommé Shell)

On peut y créer des programmes, les sauvegarder et les exécuter en cliquant sur la flèche verte. La sortie du programme est affichée au bas de l'IDE, comme le montre la Figure 4.11

Pour l'implémentation du modèle de classification de véhicule entraîné et testé sur la carte Raspberry Pi4, nous y avons accéder par le système de visualisation VNC viewer. La Figure 4.12 montre un bureau Raspbian par le biais de l'application VNC Viewer et la Figure 4.13 montre les outils de programmation sur le bureau de

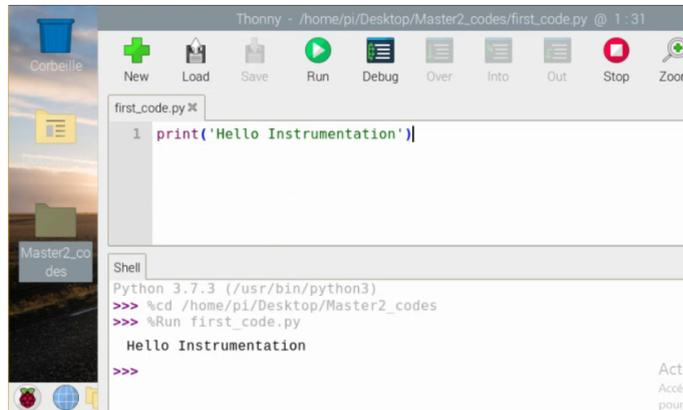


FIGURE 4.11 – Exemple d’écriture sur l’application Thonny

Raspbian.



FIGURE 4.12 – Accéder à un bureau Raspbian par le biais de l’application VNC Viewer

Après l’affichage de la Raspberry Pi4, nous avons créer un dossier sous le nom "CAR Project" dans un dossier nommé "Master2\_codes" créer précédemment dans le Desktop de la carte. La Figure 4.14 montre le chemin du dossier.

Pour transférer le modèle entraîné, la vidéo du trafic routier et le code Python vers la Raspberry Pi4, nous avons procédé comme l’illustre la Figure 4.15. Les éléments transférés sont ensuite stockés dans le dossier CAR Project (voir la Figure 4.16).



FIGURE 4.13 – Outils de programmation sur le bureau de Raspbian

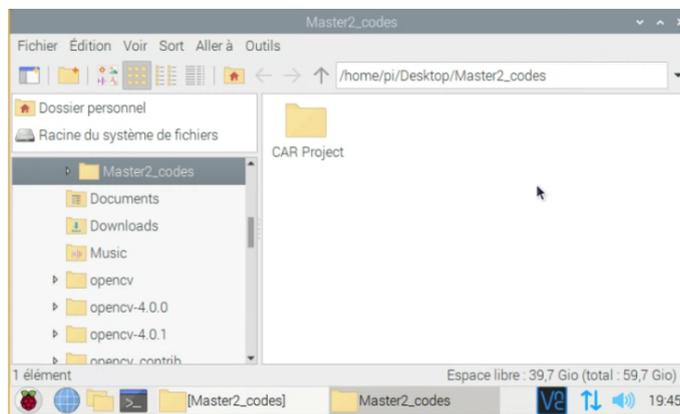


FIGURE 4.14 – le chemin du dossier CAR Project

Les résultats obtenus des différentes étapes de détection de véhicules sur la Raspberry Pi4 sont indiqués sur la Figure 4.18.

Le temps de traitement nécessaire à la carte pour la détection des véhicules est plus important. Pour chaque image, un temps moyen de 0.17 secondes est nécessaire pour la détection et la classification des différents véhicules présents dans une scène routière. (Figure 4.17) .

Enfin, des résultats de classification de véhicules sous Raspberry Pi4 sont montrés dans la Figures 4.19.

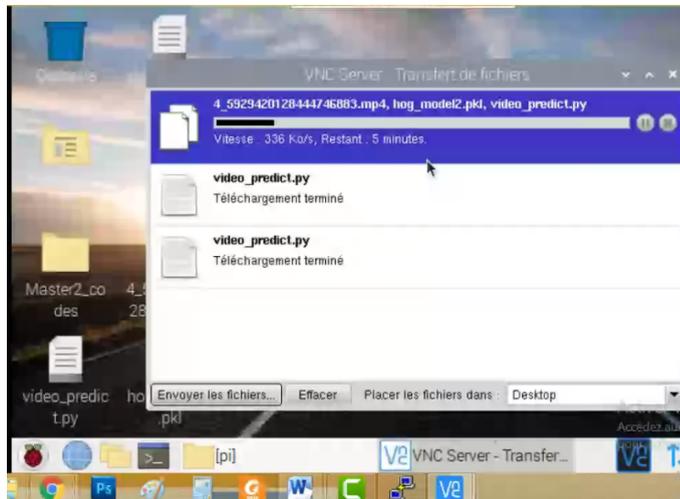


FIGURE 4.15 – Les fichiers envoyés vers la Raspberry

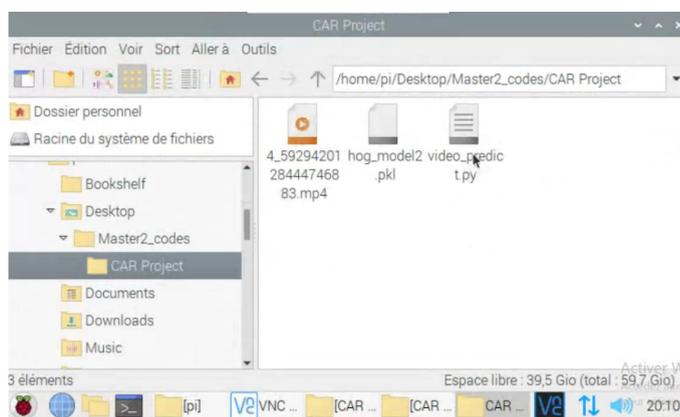


FIGURE 4.16 – Les fichiers stockés dans le dossier CAR Project

## 4.4 Conclusion

Dans ce chapitre, nous avons commencé tout d’abord par la méthode de la soustraction arrière-plan et les étapes que nous avons utilisées pour détecter les véhicules. Puis, nous avons donné quelques résultats de classification avec le nouveau modèle. Par la suite, nous avons présenté la Raspberry Pi4. Nous avons défini la configuration du système d’installation Rasbian et les différentes bibliothèques. Enfin, nous avons implémenté le modèle sur la carte Raspberry Pi4 afin d’intégrer notre système.

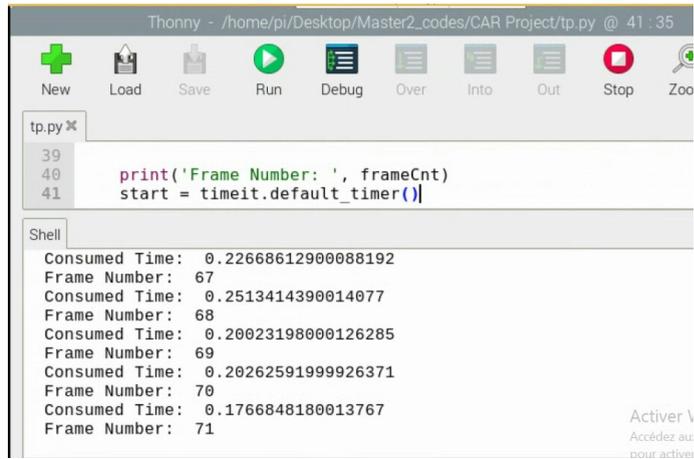


FIGURE 4.17 – Temps d'exécution de la détection et la classification sur la Raspberry Pi4

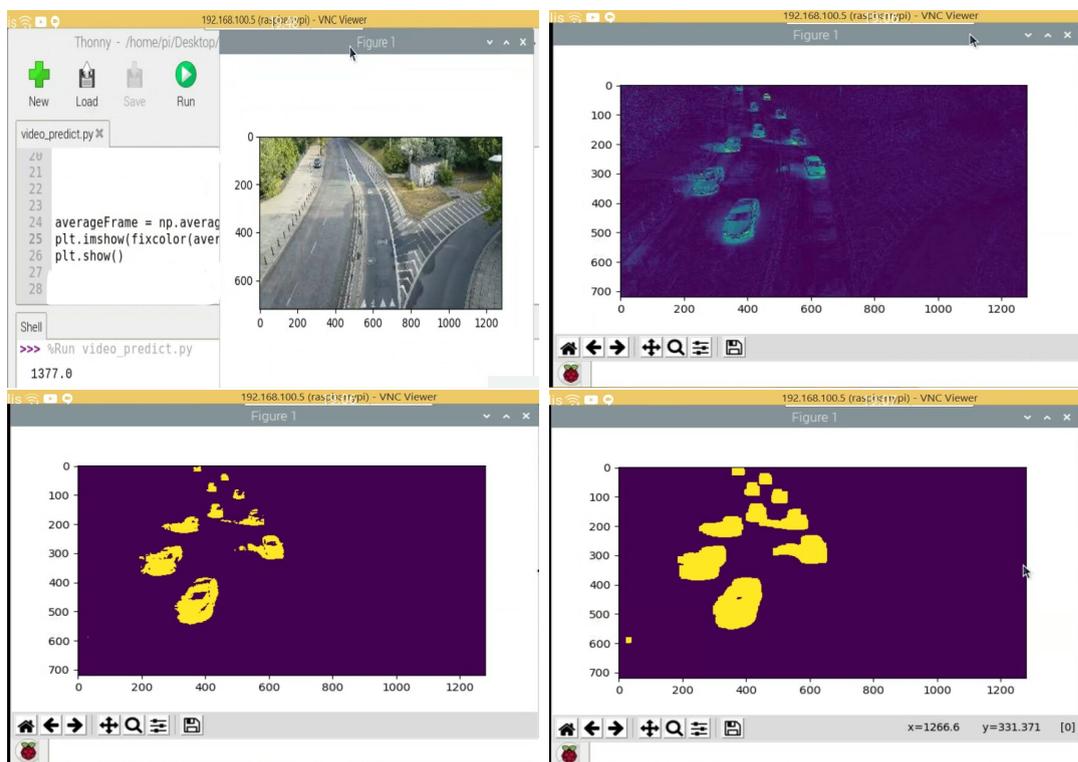


FIGURE 4.18 – Les résultats de la détection de véhicules sur la Raspberry Pi4

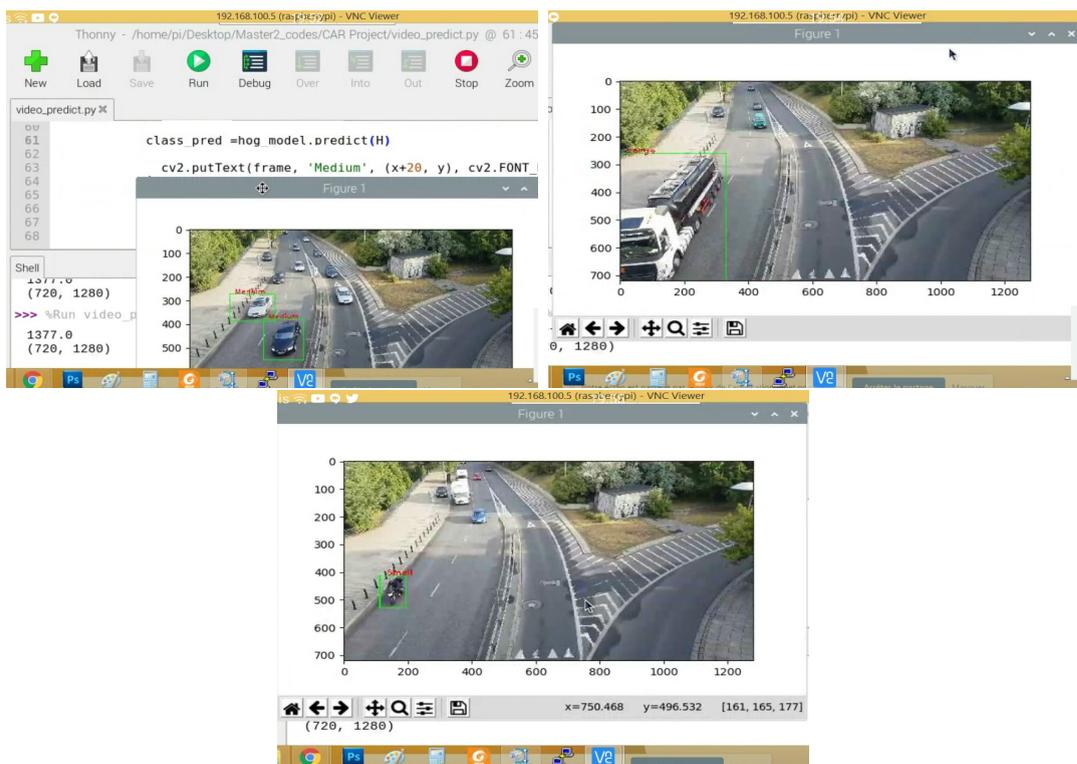


FIGURE 4.19 – Quelques résultats de classification sur la Raspberry Pi4

# Conclusion Générale

Les caractéristiques extérieures du véhicule sont propres à chaque type de véhicule, ce qui permet de le distinguer des autres. De nombreuses études dans différents domaines tels que la sécurité (prévention de la criminalité, problèmes de surveillance), systèmes de transport intelligents (systèmes d'aide à la conduite, systèmes de stationnement intelligents) et la gestion du trafic (paramètres du trafic), continuent de suggérer qu'un système de détection et de classification des véhicules est un bon moyen d'aider à résoudre certains de ces problèmes.

Ce projet s'est concentré sur la mise en œuvre d'une méthode de détection et de classification des véhicules sous plusieurs classe basée sur l'extraction des caractéristique de véhicule par le descripteur HOG qui sont ensuite combinées pour obtenir un système de classification de véhicules robuste par le classifieur support vector machine(RBF). Plusieurs classifieurs ont été testés conduisant à des résultats expérimentaux différent. Le classifieur SVM de type RBF a donné le taux de classification le plus élevé.

Le système proposé dans ce travail est beaucoup moins cher et plus facile à utiliser que d'autres méthodes de classification. Il utilise le traitement d'image comme approche pour l'analyse de la route et utilise un algorithme simple pour la classification embarquée sur une carte Raspberry Pi4 pour mettre en œuvre le système.

# Bibliographie

- [Arinaldi et al., 2018] Arinaldi, A., Pradana, J. A., and Gurusinga, A. A. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia computer science*, 144 :259–268.
- [Audebert et al., 2017] Audebert, N., Le Saux, B., and Lefèvre, S. (2017). Segment-before-detect : Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4) :368.
- [Barnich and Van Droogenbroeck, 2010] Barnich, O. and Van Droogenbroeck, M. (2010). Vibe : A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6) :1709–1724.
- [Bender and Kowalski, 1974] Bender, C. and Kowalski, B. (1974). Multiclass linear classifier for spectral interpretation (pattern recognition). *Analytical Chemistry*, 46(2) :294–296.
- [Bensrhair et al., 2002] Bensrhair, A., Bertozzi, A., Broggi, A., Fascioli, A., Mousset, S., and Toulminet, G. (2002). Stereo vision-based feature extraction for vehicle detection. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 465–470. IEEE.
- [Bertozzi et al., 1997] Bertozzi, M., Broggi, A., and Castelluccio, S. (1997). A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 43(1-5) :317–325.
- [Betke et al., 2000] Betke, M., Haritaoglu, E., and Davis, L. S. (2000). Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine vision and applications*, 12(2) :69–83.
- [Bhatnagar et al., 2017] Bhatnagar, B. L., Singh, S., Arora, C., and Jawahar, C. (2017). Unsupervised learning of deep feature representation for clustering ego-

- centric actions. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1447–1453.
- [Buch et al., 2008] Buch, N., Orwell, J., and Velastin, S. A. (2008). Detection and classification of vehicles for urban traffic scenes.
- [Bucher et al., 2003] Bucher, T., Curio, C., Edelbrunner, J., Igel, C., Kastrup, D., Leefken, I., Lorenz, G., Steinhage, A., and von Seelen, W. (2003). Image processing and behavior planning for intelligent vehicles. *IEEE Transactions on Industrial electronics*, 50(1) :62–75.
- [Byeon and Kwak, 2017] Byeon, Y.-H. and Kwak, K.-C. (2017). A performance comparison of pedestrian detection using faster rcnn and acf. In *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 858–863. IEEE.
- [Caruana and Niculescu-Mizil, 2006] Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168.
- [Chan, 2015] Chan, J. (2015). *Learn Python in One Day and Learn it Well : Python for Beginners with Hands-on Project : the Only Book You Need to Start Coding in Python Immediately*. CreateSpace Independent Publishing.
- [Chen et al., 2011] Chen, H.-Y., Chuang, C.-H., Yang, Y.-J., and Wu, T.-P. (2011). Exploring the risk factors of preterm birth using data mining. *Expert systems with applications*, 38(5) :5384–5387.
- [Cherrett et al., 2001] Cherrett, T., Bell, H., and McDonald, M. (2001). Estimating vehicle speed using single inductive loop detectors. In *Proceedings of the Institution of Civil Engineers-Transport*, volume 147, pages 23–32. Thomas Telford Ltd.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3) :273–297.
- [Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1) :21–27.
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

- [Dalal et al., 2006] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer.
- [de Souza et al., 2015] de Souza, A. M., Yokoyama, R. S., Botega, L. C., Mene-guette, R. I., and Villas, L. A. (2015). Scorpion : A solution using cooperative rerouting to prevent congestion and improve traffic condition. In *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*, pages 497–503. IEEE.
- [Dong et al., 2015] Dong, Z., Wu, Y., Pei, M., and Jia, Y. (2015). Vehicle type classification using a semisupervised convolutional neural network. *IEEE transactions on intelligent transportation systems*, 16(4) :2247–2256.
- [Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern classification second edition john wiley & sons. *New York*, 58 :16.
- [Eichner and Breckon, 2007] Eichner, M. L. and Breckon, T. P. (2007). Real-time video analysis for vehicle lights detection using temporal information. In *4th European Conference on Visual Media Production*, pages 1–1. IET.
- [Fix and Hodges, 1951] Fix, E. and Hodges, J. L. (1951). Nonparametric discrimination : Consistency properties. *Randolph Field, Texas, Project*, pages 21–49.
- [Friedman, 1996] Friedman, J. H. (1996). Another approach to polychotomous classification. *Technical Report, Statistics Department, Stanford University*.
- [Fukunaga and Hostetler, 1973] Fukunaga, K. and Hostetler, L. (1973). Optimization of k nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3) :320–326.
- [Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- [Ibarra-Arenado et al., 2017] Ibarra-Arenado, M., Tjahjadi, T., Pérez-Oria, J., Robla-Gómez, S., and Jiménez-Avello, A. (2017). Shadow-based vehicle detection in urban traffic. *Sensors*, 17(5) :975.
- [Kaewkamnerd et al., 2009] Kaewkamnerd, S., Pongthornseri, R., Chinrungrueng, J., and Silawan, T. (2009). Automatic vehicle classification using wireless magnetic

- sensor. In *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications*, pages 420–424. IEEE.
- [Kim and Kim, 2003] Kim, J. B. and Kim, H. J. (2003). Efficient region-based motion segmentation for a video monitoring system. *Pattern recognition letters*, 24(1-3) :113–128.
- [Knerr et al., 1990] Knerr, S., Personnaz, L., and Dreyfus, G. (1990). Single-layer learning revisited : a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer.
- [Lan and Chang, 2014] Lan, C.-L. and Chang, G.-L. (2014). A traffic signal optimization model for intersections experiencing heavy scooter–vehicle mixed traffic flows. *IEEE Transactions on Intelligent Transportation Systems*, 16(4) :1771–1783.
- [Latourrette, 2000] Latourrette, M. (2000). Toward an explanatory similarity measure for nearest-neighbor classification. In *European Conference on Machine Learning*, pages 238–245. Springer.
- [Li et al., 2002] Li, X., Liu, Z.-Q., and Leung, K.-M. (2002). Detection of vehicles from traffic scenes using fuzzy integrals. *Pattern Recognition*, 35(4) :967–980.
- [Liu and Parhi, 2016] Liu, Y. and Parhi, K. K. (2016). Computing rbf kernel for svm classification using stochastic logic. In *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, pages 327–332. IEEE.
- [Ma and Grimson, 2005] Ma, X. and Grimson, W. E. L. (2005). Edge-based rich representation for vehicle classification. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 2, pages 1185–1192. IEEE.
- [Mandhare et al., 2018] Mandhare, P. A., Kharat, V., and Patil, C. (2018). Intelligent road traffic control system for traffic congestion a perspective. *International Journal of Computer Sciences and Engineering*, 6(07) :2018.
- [Mohammed et al., 2016] Mohammed, M., Khan, M. B., and Bashier, E. B. M. (2016). *Machine learning : algorithms and applications*. Crc Press.
- [Nguyen et al., 2014] Nguyen, V. C., Dinh, D. K., et al. (2014). Length and speed detection using microwave motion sensor. In *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, pages 371–376. IEEE.

- [Pi et al., 2019] Pi, M., Yeon, H., Son, H., and Jang, Y. (2019). Visual cause analytics for traffic congestion. *IEEE transactions on visualization and computer graphics*.
- [Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. *arXiv preprint arXiv :1506.01497*.
- [Sánchez A, 2003] Sánchez A, V. D. (2003). Advanced support vector machines and kernel methods. *Neurocomputing*, 55(1-2) :5–20.
- [Sarker et al., 2020] Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., and Ng, A. (2020). Cybersecurity data science : an overview from machine learning perspective. *Journal of Big data*, 7(1) :1–29.
- [Seenouvong et al., 2016] Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K., and Ohnishi, N. (2016). Vehicle detection and classification system based on virtual detection zone. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5. IEEE.
- [Shehata et al., 2019] Shehata, M., Abo-Al-Ez, R., Zaghlool, F., and Abou-Kreisha, M. T. (2019). Vehicles detection based on background modeling. *arXiv preprint arXiv :1901.04077*.
- [Shi et al., 2007] Shi, S., Qin, Z., and Xu, J. (2007). Robust algorithm of vehicle classification. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 3, pages 269–272. IEEE.
- [Sivaraman and Trivedi, 2013] Sivaraman, S. and Trivedi, M. M. (2013). Looking at vehicles on the road : A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE transactions on intelligent transportation systems*, 14(4) :1773–1795.
- [Todeschini, 1989] Todeschini, R. (1989). k-nearest neighbour method : The influence of data transformations and metrics. *Chemometrics and intelligent laboratory systems*, 6(3) :213–220.

- [Tsai et al., 2018] Tsai, C.-C., Tseng, C.-K., Tang, H.-C., and Guo, J.-I. (2018). Vehicle detection and classification based on deep neural network for intelligent transportation applications. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1605–1608. IEEE.
- [Tsai et al., 2007] Tsai, L.-W., Hsieh, J.-W., and Fan, K.-C. (2007). Vehicle detection using normalized color and edge map. *IEEE transactions on Image Processing*, 16(3) :850–864.
- [Van Leeuwen and Groen, 2005] Van Leeuwen, M. B. and Groen, F. C. (2005). Vehicle detection with a mobile camera : spotting midrange, distant, and passing cars. *IEEE robotics & automation magazine*, 12(1) :37–43.
- [Velazquez-Pupo et al., 2018] Velazquez-Pupo, R., Sierra-Romero, A., Torres-Roman, D., Shkvarko, Y. V., Santiago-Paz, J., Gómez-Gutiérrez, D., Robles-Valdez, D., Hermosillo-Reynoso, F., and Romero-Delgado, M. (2018). Vehicle detection with occlusion handling, tracking, and oc-svm classification : A high performance vision-based system. *Sensors*, 18(2) :374.
- [Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE.
- [Vural and Dy, 2004] Vural, V. and Dy, J. G. (2004). A hierarchical method for multi-class support vector machines. In *Proceedings of the twenty-first international conference on Machine learning*, page 105.
- [Wang et al., 2019] Wang, Y., Ban, X., Wang, H., Wu, D., Wang, H., Yang, S., Liu, S., and Lai, J. (2019). Detection and classification of moving vehicle from video using multiple spatio-temporal features. *IEEE Access*, 7 :80287–80299.
- [Wu et al., 2008] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1) :1–37.
- [Zhan and Ji, 2011] Zhan, W. and Ji, X. (2011). Algorithm research on moving vehicles detection. *Procedia Engineering*, 15 :5483–5487.
- [Zhang, 2004] Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2) :3.

- [Zhang et al., 2011] Zhang, L., Wang, R., Cui, L., et al. (2011). Real-time traffic monitoring with magnetic sensor networks. *Journal of information science and engineering*, 27(4) :1473–1486.
- [Zhang et al., 2007] Zhang, W., Fang, X., Yang, X., and Wu, Q. J. (2007). Spatiotemporal gaussian mixture model to detect moving objects in dynamic scenes. *Journal of Electronic Imaging*, 16(2) :023013.