

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945-Guelma-
Faculty of Mathematics, Computer Science and Science of Matter
Department of Computer Science



Thèse de Master

Spécialité: Informatique

Option:

sciences et technologies de l'information et de
La communication

Thème

Détection d'objets et deep Learning dans un
trafic routier

Présenté par:
Benkamouch Wiam

supervisé par:
Dr. Madi Leyla

Members du jury:
Dr. Benhamza karima
Dr. Benhamida Nadjete

Examineur
président

June 2023.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ اهُرَأ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ * خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ * اهُرَأ وَرَبُّكَ الْأَكْرَمُ * الَّذِي عَلَّمَ بِالْقَلَمِ * عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ ﴾

[سورة العلق 1-5]

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ يَرْفَعُ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۗ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ۗ ﴾

[سورة المجادلة 11]

قال رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ

« مَنْ سَلَكَ طَرِيقًا يَطْلُبُ فِيهِ عِلْمًا، سَلَكَ اللَّهُ بِهِ طَرِيقًا مِنْ طُرُقِ الْجَنَّةِ، وَإِنَّ الْمَلَائِكَةَ لَتَضَعُ أُنْجُبَهَا رِضًا لَطَالِبِ الْعِلْمِ، وَإِنَّ الْعَالِمَ لَيَسْتَغْفِرُ لَهُ مَنْ فِي السَّمَاوَاتِ، وَمَنْ فِي الْأَرْضِ، وَالْحَيَاتَانُ فِي جَوْفِ الْمَاءِ، وَإِنَّ فَضْلَ الْعَالِمِ عَلَى الْعَابِدِ، كَفَضْلِ الْقَمَرِ لَيْلَةَ الْبَدْرِ عَلَى سَائِرِ الْكَوَاكِبِ، وَإِنَّ الْعُلَمَاءَ وَرَثَةُ الْأَنْبِيَاءِ، وَإِنَّ الْأَنْبِيَاءَ لَمْ يُورَثُوا دِينَارًا وَلَا دِرْهَمًا، وَرَثَتُوا الْعِلْمَ، فَمَنْ أَخَذَهُ أَخَذَ بِحِطِّ وَافِرٍ ».

اللَّهُمَّ إِنِّي أَسْأَلُكَ عِلْمًا نَافِعًا، وَقَلْبًا خَاشِعًا، وَرِزْقًا مَبْرُكًا، وَعَمَلًا زَكِيًّا مُتَقَبَّلًا.

ربنا افتح لنا أبواب رحمتك، وسهل لنا ما رزقتنا.

اللهم كما انعمت فزد وكما زدت فبارك وكما باركت فتمم وكما أتممت فثيب.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ وَقُلْ رَبِّ زِدْنِي عِلْمًا ﴾

[سورة طه 114]



Benkamouchi wiam

**Projet de fin d'étude de Master : Détection d'objet et deep
Learning dans un trafic routier.**

Dirigé par : Mme Madi Leyla

**Université 8 Mai 1945 Guelma
Faculté MI-SM
Département d'Informatique**

Remerciements



*Au nom de Dieu, le plus gracieux, le plus miséricordieux.
Qui m'a donné la force, le courage, et la détermination Nécessaire
pour terminer ce travail.*

*La rédaction de ce mémoire et cette soutenance marque la fin d'une
aventure à plusieurs facettes : aventure dans le monde de la recherche,
qui ne devrait pas en rester là, aventure humaine, aventure familiale.
Différentes personnes m'ont accompagnée tout au long de ce parcours*

Au début, je souhaite adresser mes remerciements les plus sincères

Mon Enseignante Et Encadreur

Et J'exprime toute ma gratitude à Madame Madi Leyla

Docteur à l'université, 08 Mai 1945 Guelma.

Qui a accepté de m'encadrer

Pour l'aide précieuse qu'il m'a apportée

*J'ai eu l'honneur d'être parmi vos élèves et de bénéficier de votre
riche enseignement.*

Aux membres du jury

Madame Benhamza Karima . Vous nous avez fait un grand honneur
en acceptant de siéger parmi les membres de jury de cette mémoire.

A notre maître et Examineur de jury. Madame Benhamida

*Mes sincères remerciements Pour avoir accepté de présider le jury,
vous nous offrez le grand honneur et le grand plaisir*

*Mes sincères remerciements à tous les enseignants du département
d'Informatique Guelma.*

*Mes remerciements aussi à Monsieur «**Boughida Adel**»*

*Mes Parents, Ma sœur, Mes frères, Mes camarades, Ma famille et tous
ceux qui m'ont soutenu, aidé et contribué de près ou de loin à la
réalisation de ce travail.*



Dédicace

*Je commence mes dédicaces au nom de Dieu et puis de son prophète
Mohamed.*

*Louange à Dieu tout puissant, qui m'a permis de voir ce jour tant
attendu.*

*Par ces quelques modestes mots je souhaite témoigner ma
reconnaissance à tous ceux qui m'ont aidé à lancer ce travail.*

Je tiens à dédier ce modeste travail.

*Aux êtres les plus chers à mes yeux qui m'ont soutenu durant toutes
mes études,*

A l'homme à qui je dois ma vie, ma réussite et tout mon respect.

A mon très cher père « Messaoud Benkamouch ».

*De tous les pères, tu es le meilleur. Grâce à toi papa j'ai appris le sens
du travail et de la responsabilité. Tes conseils ont toujours guidé mes
pas vers la réussite.*

*Vous êtes et vous resterez pour moi ma référence, et la lumière qui
illumine mon chemin.*

*Et j'espère que vous y trouverez le fruit vos efforts et le témoignage de
ma grande fierté de vous avoir comme père.*

*Je te dois ce que je suis aujourd'hui et ce que je serai demain et je ferai
toujours de mon mieux pour rester ta fierté et ne jamais te décevoir.*

A mon adorable mère « Fouzia Yousfi »

*Tu m'as comblé avec ta tendresse et affection tout au long de mon
parcours. Tu n'as cessé de me soutenir et de m'encourager durant
toutes les années de mes études. Ta prière et ta bénédiction m'ont été
d'un grand secours pour mener à bien mes études et tout au long de
ma vie.*

*En ce jour mémorable, pour moi ainsi que pour toi, reçoit ce travail en
signe de ma vive reconnaissance et ma profonde estime. Puisse le tout
puissant te donner santé, bonheur et longue vie afin que je puisse te
combler à mon tour.*

J'espère ne jamais te décevoir, ni trahir ta confiance et tes sacrifices.

A ma belle sœur « Manar »

*Aucune dédicace ne peut exprimer mon amour et ma gratitude, tu es
la sœur qui assures sa rôle comme il faut, tu es été à mes côtés pendant
toutes les étapes de ce travail je vous suis très reconnaissante.*

Je te dis merci.



*A mes chers frères «Abd Rahmen» «Abd djalil» et
«Mohamed»*

Merci d'avoir me soutenu et merci pour tous les bons moments que nous avons passés ensemble. Que Dieu nous rassemble pour toujours.

À mon cher fiancé «Marwen»

Pour la patience et le soutien dont il a fait preuve pendant toute la durée de ce travail et à qui je voudrais exprimer mes affections et mes gratitudes. Merci infiniment. Que dieu vous garde pour moi.

Sans oublier ma seconde famille «Nekaa»

Et ma chère tante «Amel & Aaicha»

À ma belle «Nouna»

Pour l'amour et l'affection qui nous unissent.

Je trouve en toi les conseils de la sœur et soutien de l'amie.

Tu comptes énormément pour moi.

*Je prie Dieu le tout puissant de préserver notre attachement mutuel,
et d'exaucer tous nos rêves.*

Merci pour vos aides, et vos supports.

À ma belle «madjeda»

Merci pour vos aides et vos supports dans les moments difficiles.

Tu comptes énormément pour moi.

Je prie Dieu le tout puissant de préserver notre attachement mutuel,

À ma belle «Lina Djihen»

Pour l'amour et l'affection qui nous unissent.

Je te dis merci et Je te dédie ce travail avec mes vœux de réussite, de prospérité et de bonheur.

À mes chères amis .

Pour leurs soutiens moraux et vos conseils précieux tout au long de mes études. Vous êtes pour moi des sœurs et des amies sur qui je peux compter. En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble.

À toute personne qui a participé de près ou de loin à la réalisation de ce travail,

Spécialement Mme « Dr Madi Leyla»,

À tous mes amis et mes collègues, A toute ma famille et tous ceux qui me sont chers, Il me serait difficile de vous citer tous, vous êtes dans mon cœur,

A vous les lecteurs de ces lignes.

L'objectif consiste à concevoir un système destiné à l'analyse automatique de scènes autoroutières.

Le réseau routier joue un rôle essentiel dans le développement d'un pays car il représente une base sur laquelle se fonde plusieurs secteurs tels que le transport des biens et des personnes ; donc il est le moyen vital de l'économie et du développement social d'un pays. C'est pourquoi l'intégration de systèmes d'aide à la conduite est cruciale lors de la conception de nouveaux véhicules.

Dans ce travail nous avons conçu une application de détection de véhicule basée sur vidéo. Pour atteindre à cet objectif, nous avons implémenté trois algorithmes. Le premier fait partie de ML (SVM) et les deux autres font partie du deep learning (YOLOV5 et YOLOV8).

Mots-clés : Détection d'objet, machine learning ,Deep learning , apprentissage ,video.

The objective is to design a system for automatic analysis of highway scenes.

The road network plays a crucial role in the development of a country as it serves as the foundation for various sectors such as transportation of goods and people. Therefore, the integration of driving assistance systems is crucial in the design of new vehicles.

In this work, we have developed a vehicle detection application based on video. To achieve this goal, we have implemented three algorithms. The first one is a part of machine learning (SVM), and the other two are part of deep learning (YOLOV5 and YOLOV8).

Keywords: Object detection, machine learning, deep learning, learning, video.

LISTE DES ABRÉVIATIONS

ML Apprentissage automatique.

DL Apprentissage profond.

CNN Réseau de neurones convolutif.

SVM Support Vector Machine.

YOLO You only look once .

RNN Réseau de neurones récurrent.

LSTM Mémoire à court terme à long terme.

GRU Unité récurrente à portes.

GAN Réseau antagoniste génératif.

GPU Unité de traitement graphique .

TPU Unité de traitement de tenseurs.

Liste des matières

Résumé I

Abstract II

Liste des abréviations	III
Liste des matières	III
Introduction générale	1
CHAPTER I Image numérique	2
I.1 Image numérique	2
I.1.1 Définition 1	2
I.1.2 Définition 2	3
I.2 Caractéristique d'une image	3
I.2.1 La résolution	3
I.2.2 Pixele de l'image	4
I.2.3 voisinage	4
I.2.4 Le contraste	4
I.2.5 Le Contour	5
I.2.6 Histogramme	5
I.3 Type des images	6
I.3.1 Images en couleurs (Polychromes)	6
I.3.2 Image binaire (noir et blanc)	7
I.3.3 Images à niveaux de gris (Monochromes)	7
I.4 Représentation numérique de l'image	7
I.4.1 Type de l'image numérique	7
I.4.1.1 Image vectoriel	8
I.4.1.2 Image matriciel	8
I.4.2 Les espaces colorimétriques	8
I.4.2.1 L'espace RVB	8
I.4.2.2 L'espace lab	9
I.5 Traitement d'images numérique	10
I.5.1 Acquisition	10
I.5.2 Segmentation	11
I.5.3 Filtrage	11
I.6 Domaines D'application	12
I.6.1 Imagerie aérienne et spatiale	12
I.6.2 Technologies biomédicales	12
I.6.3 La robotique	12
I.6.4 La télésurveillance	12
I.7 CONCLUSION	13
CHAPTER II Détection des objets	14

II.1	INTRODUCTION	14
II.2	Définition de détection d'objets	14
II.3	L'extraction des caractéristiques d'objet	15
II.3.1	Forme	15
II.3.2	Couleur	15
II.3.3	La texture	15
II.4	Apprentissage automatique(machine learning)	15
II.4.1	Support Vector Machine (SVM)	16
II.4.2	K-Nearest Neighbors (kNN)	17
II.5	Apprentissage profond (deep learning)	17
II.5.1	Convolution Neural Network(CNN)	18
II.5.2	You only look once : YOLO	18
II.6	Comparison entre les versions YOLO	19
II.6.1	Les réseaux de capsules (Caps Nets)	19
II.6.2	Réseaux neuronaux récurrents(RNN)	20
II.6.3	Réseaux de neurones génératifs adverses (GAN)	20
II.7	déférence entre machine Learning et deep Learning	20
II.8	CONCLUSION	20
CHAPTER III Conception et Implimentation		22
III.1	Introduction	22
III.2	Outil de développement	23
III.2.1	Jupyter	23
III.2.2	Google Colab	23
III.2.3	Python	24
III.2.4	Bibliothèques utilisées	24
III.2.4.1	OpenCV	24
III.2.4.2	SciPy	25
III.2.5	Materiel et limites	25
III.3	Description de l'application	25
III.3.1	Système basé sur deep Learning	26
III.3.1.1	Algorithme 1(YOLOV5)	26
III.3.1.2	Algorithme 2 (YOLOV8)	27
III.3.2	algorithme 3 (Apprentissage ML)	30
III.3.2.1	Extraction des caractéristiques	31
III.3.2.2	Apprentissage	33
III.4	Présentation du Système	34
III.4.1	Intarface	34
III.4.2	l'espace de YOLOV5	34

III.4.3 l'espace de YOLOV8	34
III.4.4 l'espace de Machine Learning	35
III.5 Expérimentation et résultats	35
III.6 Bases de Données	36
III.7 Expérimentation	36
III.8 Résultats	37
III.8.1 Résultats de Machine Learning	37
III.8.2 Résultats de Deep Learning	38
III.8.2.1 Résultats de YOLOV5	38
III.8.2.2 Résultats de YOLOV8	39
III.9 Conclusion	39
CONCLUSION GENERAL	39

List of Figures

I.1	Image numérique [1]	3
I.2	résolution d'un image[2]	4
I.3	Représentation Pixels d'une image [3].	4
I.4	Contraste d'une image [4].	5
I.5	Image bruitée [5].	5
I.6	Contour d'une image [6].	6
I.7	Image avec histogramme [6].	6
I.8	Images couleurs (polychrome)[7].	7
I.9	Image binaire (noir et blanc)[8]	7
I.10	image Monochromes (256 couleurs)[8]	8
I.11	la représentation del'espace RGB[9].	10
I.12	La représentation del'espace lab[9].	10
I.13	la segmentation d'images (figure1 : Image d'origine, figure2 : l'image segmentée)[10].	11
I.14	Figure I.14: Exemple de filtrage d'une image bruitée (à gauche l'image d'origine, à droitel'image filtrée) [11].	12
II.1	Apprentissage Automatique ML[12].	16
II.2	Exemple illustrative d'un modèle SVM [12].	17
II.3	Apprentissage profond DL[12].	17
II.4	Architecture générale du CNN [13].	18
III.1	Environnement de jupyter	23
III.2	Environnement de Google Colab	24
III.3	L'architecture de modèle YOLOv5.	28
III.4	L'architecture de modèle YOLOv8.	30
III.5	Schéma de l'algorithme basé sur ML	31
III.6	Extraction des caractéristiques	32
III.7	Normalisation des caractéristiques	32
III.8	l'application de HOG.	33
III.9	L'interface de l'application	34
III.10	détection avec YOLOV5	35
III.11	détection avec YOLOV8.	36

III.12déttection avec ML 37

List of Tables

I.1	Comparaison entre l'image vectorielle et matricielle [14].	9
II.1	Les différences entr versions de YOLO (You Only Look Once)	19
II.2	les différences entre Machine learning et deep Learning	21
III.1	Paramètres de HOG	32
III.2	Matrice de confusion du système ML	37
III.3	Résultats du notre système ML	38
III.4	Résultats de YOLOV5	38
III.5	Résultats de YOLOV8	39

INTRODUCTION GÉNÉRALE

Afin de faciliter le processus de conduite et d'éviter les accidents de la route, qui sont devenus incrémentales ces derniers temps, et qui sont devenus dangereux pour la vie humaine en général et pour la vie des enfants en particulier, donc un être humain fait une grande attention dans la recherche d'une solution appropriée pour sortir de ce problème ou au moins de le minimiser.

Alors la solution est de franchir le domaine de la vision par ordinateur que nous étudions et utilisons pour identifier ce type d'images. Le transport routier est le mode de transport le plus utilisé générant ainsi un trafic important qui ne cesse d'augmenter. Malheureusement, cette augmentation du trafic routier a des conséquences sur la santé publique. Cependant, le facteur humain représente la première cause des accidents. C'est pourquoi l'intégration de systèmes d'aide à la conduite est cruciale lors de la conception de nouveaux véhicules. Le présent mémoire se situe dans le contexte de l'analyse d'images numériques appliquée à la détection d'objet au trafic routier. Les avancées récentes en matière de vision par ordinateur et le deep learning ont permis d'obtenir des résultats promoteurs dans ce domaine. Notre travail consiste à implémenter deux systèmes de nature différente (Machine Learning et Deep Learning) qui permettent de détecter les véhicules à partir d'une vidéo. Le but de ce travail c'est de faire une comparaison entre apprentissage automatique et profond. Pour cela nous avons implémenté trois algorithmes. Le premier algorithme est le modèle SVM en utilisant différents types de caractéristiques. Les deux autres algorithmes d'apprentissage profond se manifestent par le modèle personnalisé, de YOLOv5 et YOLOv8, pour la détection des véhicules.

Le manuscrit est organisé de la manière suivante:

- **Chapitre 1** : Des généralités sur le domaine de traitement d'images.
- **Chapitre 2** : La détection d'objets et l'apprentissage.
- **Chapitre 3** : Conception et réalisation.

sectionIntroduction L'appellation "image numérique" désigne toute image (dessin, icône, photographie...) acquise, créée, traitée et stockée sous forme binaire. Elle peut être convertie à partir d'une vue réelle en utilisant des convertisseurs analogique-numérique tels que des capteurs présents dans des dispositifs tels que les scanners, les appareils photo ou les caméscopes numériques. Elle peut également être créée directement par des programmes informatiques à l'aide d'une souris, de tablettes graphiques ou par modélisation 3D (images de synthèse). Une fois stockée sur un support informatique (disque dur, etc.), elle peut être traitée à l'aide d'outils graphiques pour effectuer diverses transformations telles que la modification de la taille (compression), des couleurs, l'ajout ou la suppression d'éléments, l'application de filtres variés (restauration et amélioration), le codage, etc. L'objectif de ce traitement est de préparer l'image pour l'analyse en extrayant les informations pertinentes. L'image traitée est décrite de manière numérique (structurelle), sans lien direct avec la réalité qu'elle représente, tandis que l'image symbolique conserve un lien avec la scène observée. En interprétant les informations extraites, il est possible de prendre des décisions et d'engager des actions en conséquence.

L'objectif du traitement d'images est d'extraire l'information utile à analyser.

I.1 Image numérique

I.1.1 Définition 1

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques). La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal

qu'elle représente (une infinité de valeurs dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur [14].

I.1.2 Définition 2

L'image est d'abord un signal 2D (x, y) , généralement cette image représente la réalité 3 (x, y, z) .

D'un point de vue mathématique :

- Une image est une matrice de nombres représentant un signal.
- Plusieurs outils sont utilisés pour manipuler ce signal.

D'un point de vue humain

- Une image contient plusieurs informations sémantiques.
- Il faut interpréter le contenu au-delà de la valeur des nombres[15].

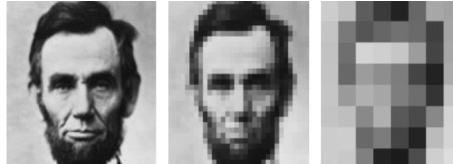


Figure I.1: Image numérique [1]

I.2 Caractéristique d'une image

Une image peut être définie par trois caractéristiques de base :

I.2.1 La résolution

La résolution de l'image est le nombre de pixels par unité de longueur. Elle s'exprime en (pixel par pouce). Ainsi, la résolution peut établir la relation entre la définition en pixels. D'une image et sa taille réelle représentée sur le support physique[8]

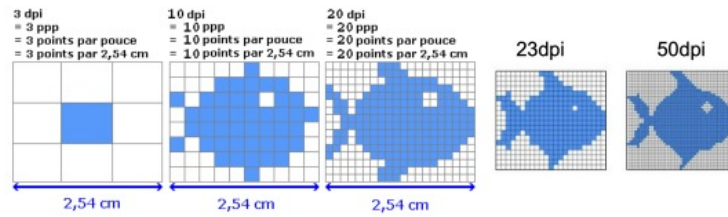


Figure I.2: résolution d'une image[2]

I.2.2 Pixele de l'image

Les pixels sont approximativement rectangulaires, parfois carrés. Leur dimension peut être changée en réglant l'écran ou la carte graphique. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image[8].



Figure I.3: Représentation Pixels d'une image [3].

I.2.3 voisinage

La surface de l'image est représentée par un ensemble de surfaces rectangulaires. Deux types de quartiers sont distingués :

- Voisinage 4 : Pixels qui partagent un bord avec le pixel central[4].
- Voisinage 8 : Tout pixel situé autour du pixel central[4].

I.2.4 Le contraste

Le contraste est une propriété inhérente aux images qui quantifie.

La luminosité entre les parties claires et sombres d'une image.

- Image de contraste : présente une bonne dynamique dans la distribution Valeurs de niveaux de gris sur toute la plage des valeurs possibles, avec un blanc très net et des noirs profonds.

- Images à faible contraste: avec une faible plage dynamique, la plupart des pixels ont des valeurs de niveaux de gris très proches [8].



Figure I.4: Contraste d'une image [4].

Dans une image, le bruit est toute fluctuation parasite ou dégradation que subit l'image de l'instant de son acquisition jusqu'à son enregistrement à cause de variation de l'intensité d'un pixel par rapport à ses voisins. Ce qui donne certains défauts (petits nuages, poussière et la diminution de l'intensité électrique sur les capteurs), Les sources de bruit sont multiples, certaines sont physiques liées à la qualité de l'éclairage de la scène, et électroniques liées à la stabilité du capteur de l'image durant l'acquire[5].



Figure I.5: Image bruitée [5].

I.2.5 Le Contour

Le contour est la frontière qui sépare des objets dans une image qui ont des pixels dont les niveaux de gris différents, ou la limite des objets qui marquant des changements d'intensité[5].

I.2.6 Histogramme

La distribution et la présentation des histogrammes en niveaux de gris ou en couleur dépendent de l'image.

Cela concerne la vitesse à laquelle chaque nuance de gris (couleur) se manifeste dans l'image exposée. Fournit un aperçu complet de la dispersion des couleurs en niveaux de gris.



Figure I.6: Contour d'une image [6].

On peut facilement observer la gamme dans laquelle les couleurs en niveaux de gris sont distribuées.

Dans les situations où l'image apparaît excessivement sombre[6].

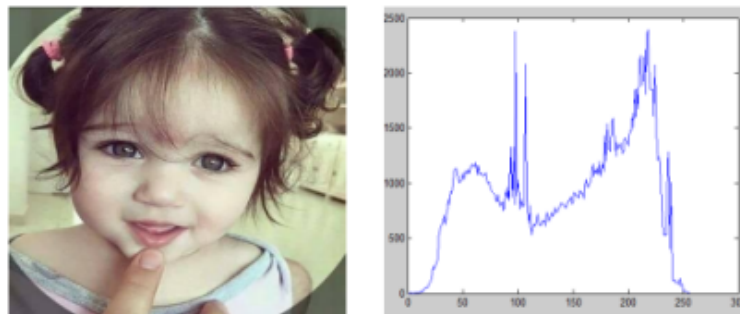


Figure I.7: Image avec histogramme [6].

I.3 Type des images

On distingue trois types des images:

I.3.1 Images en couleurs (Polychromes)

Ces images sont généralement codées en utilisant le codage des trois couleurs de base (Rouge, Vert, Bleu), puis on parle d'images RVB. (ceci correspond à l'encodage TV couleurs françaises). Ce sont donc des images trimodales.

Chaque couleur est codée comme Un octet, donc : composante rouge, intensité de 0 à 255
Image RVB composante verte, intensité Composante bleue de 0 à 255, intensité de 0 à 255.
Ensuite, nous pouvons encoder 16777216 couleurs différent[7].



Figure I.8: Images couleurs (polychrome)[7].

I.3.2 Image binaire (noir et blanc)

Chaque pixel est soit blanc soit noir. Puisqu'il y a uniquement deux valeurs pour chaque pixel, un seul bit est utilisé pour le coder[8].



Figure I.9: Image binaire (noir et blanc)[8]

I.3.3 Images à niveaux de gris (Monochromes)

Chaque pixel est un niveau de gris, allant de 0 (noir) à 255 (blanc). Cet intervalle de valeurs signifie que chaque pixel est codé sur huit bits (un octet). 256 niveaux de gris sont généralement suffisants pour la reconnaissance de la plupart des objets d'une scène [8].

I.4 Représentation numérique de l'image

I.4.1 Type de l'image numérique

Une image numérique peut être :



Figure I.10: image Monochromes (256 couleurs)[8]

I.4.1.1 Image vectoriel

Elle est composée de différents objets repérés par leurs coordonnées et comportant différents attributs (bordure, fond, forme, coordonnées). Leur avantage c'est qu'elles peuvent être facilement redimensionnées. Leur codage dépend directement du logiciel qui a permis de les créer[14].

I.4.1.2 Image matriciel

Une image matricielle, ou carte de points (de l'anglais bitmap est composée comme son nom l'indique d'une matrice (tableau) de points à plusieurs dimensions, chaque dimension représente une dimension spatiale (hauteur, largeur), ou autres (niveau de résolution). Dans le cas des images à deux dimensions, les points sont appelés pixels[14].

I.4.2 Les espaces colorimétriques

Une couleur est généralement représentée par trois composantes. Ces composantes définissent les espaces colorimétriques. On peut se référer à l'espace RVB, l'espace CIE XYZ ou Yxy, ou espace laboratoire.

Dépend de l'espace colorimétrique sélectionné Représente une image couleur, un nuage de couleur (c'est-à-dire un ensemble de couleur de l'image) n'aura pas la même distribution dans l'espace 3D. Les espaces colorimétriques classiques, tels que RVB, CIE XYZ, etc., sont dérivés de Une approche purement physique qui ne tient pas compte des données psychophysiques. Pour les autres espaces colorimétriques, tels que l'espace Lab, la méthode physique est Correction basée sur des données visuelles humaines.

I.4.2.1 L'espace RVB

L'espace d'acquisition et de sauvegarde des images est basé sur l'espace couleur RGB. L'acquisition utilise cet espace pour des raisons techniques évidentes dues aux capteurs de

	les avantages	les inconvénients
image vectoriel	<ul style="list-style-type: none"> • Résolution indépendante, peuvent être redimensionnées sans perte de qualité. • Fichiers plus légers, adaptés aux graphiques simples et aux illustrations. • Permettent de manipuler facilement les objets individuels, de modifier les couleurs et les traits. 	<ul style="list-style-type: none"> • Ne peuvent pas représenter des détails complexes et des variations de couleur subtiles. • Ne prennent pas en charge les effets de lumière, d'ombrage et de transparence. • Inadaptées aux images réalistes et photographiques.
Image matriciel	<ul style="list-style-type: none"> • Prise en charge des images complexes avec des variations de couleur et de texture détaillées. • Adaptées aux images réalistes, photographiques et basées sur des pixels. • Prise en charge des effets de lumière, d'ombrage et de transparence. 	<ul style="list-style-type: none"> • Les fichiers peuvent être volumineux, ce qui peut poser des problèmes de stockage et de transmission. • La redimension des images peut entraîner une perte de qualité. • Difficulté à représenter des lignes droites et des courbes en douceur.

Table I.1: Comparaison entre l'image vectorielle et matricielle [14].

type RGB.

Le triangle de Maxwell représenté dans l'espace RGB est le triangle reliant les trois couleurs pures Rouge, Vert et Bleu. Il nous permettra ainsi de suivre les différentes distorsions dues aux changements d'espaces couleur[9].

I.4.2.2 L'espace lab

Le système Lab est de CIE XYZ. il essaie de prendre en compte la réponse Logarithmique à l'œil, il a l'énorme avantage d'être uniforme. C'est très utile Par exemple, des mélanges de pigments pour le graphisme ou l'industrie textile[9].

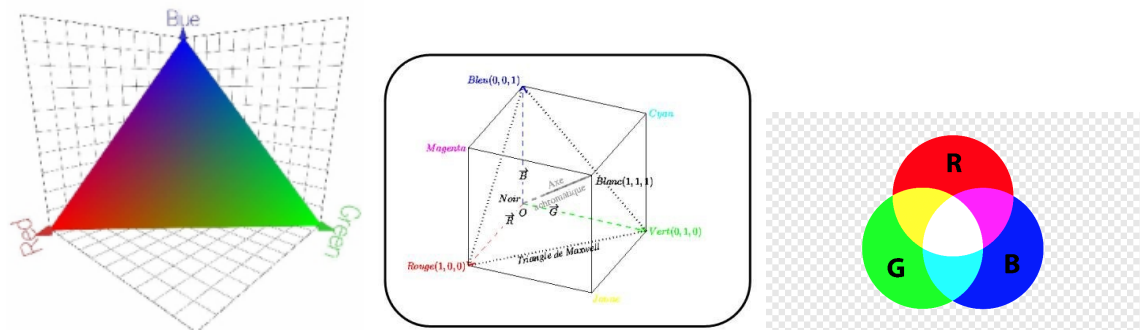


Figure I.11: la représentation de l'espace RGB[9].

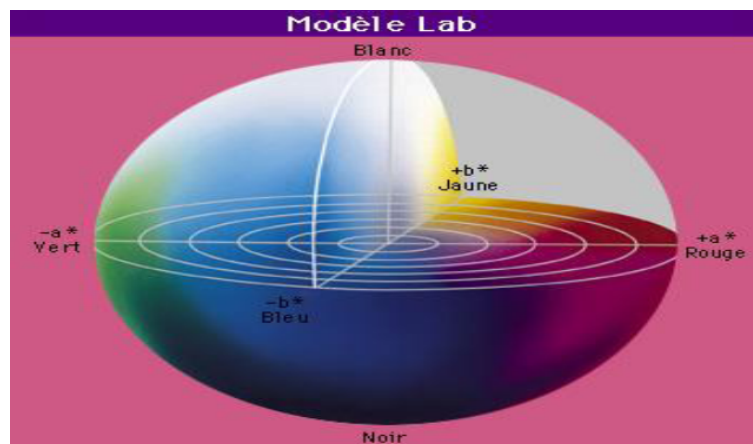


Figure I.12: La représentation de l'espace lab[9].

I.5 Traitement d'images numérique

Le traitement d'images est une discipline issue de deux domaines croisés Ce sont l'informatique et les mathématiques appliquées, ces dernières étudiant les caractéristiques et les propriétés des images numériques et leurs transformations pour que Soit améliorer leur qualité, soit en extraire des informations. Suivant,nous allons démontrer trois opérations de traitement d'image visant à améliorer la qualité ou extraire des informations de cette dernière[15].

I.5.1 Acquisition

L'acquisition d'images est l'un des maillons essentiels de toute chaîne de conception et de production d'images.Pour pouvoir manipuler une image sur un système informatique, il faut d'abord la convertir pour qu'elle devienne.

Le système est lisible et utilisable.Cette opération est effectuée par le système d'entrée peuvent être divisés en deux catégories : les appareils photo numériques et scanner[16]

I.5.2 Segmentation

La segmentation consiste à regrouper tous les pixels d'une image en différents groupes, en évaluant chaque groupe selon des critères d'homogénéité. L'objectif de cette technique de traitement d'image est de séparer de manière précise les différents objets présents dans l'image traitée, afin d'extraire les informations qu'elle contient. Il existe plusieurs méthodes et approches pour effectuer la segmentation d'image[10].

Les méthodes de segmentation

- Segmentation par région.
- Segmentation par contour.
- Segmentation par classification.



Figure I.13: la segmentation d'images (figure1 : Image d'origine, figure2 : l'image segmentée)[10].

I.5.3 Filtrage

Le principe du filtrage consiste à modifier les valeurs des pixels d'une image, généralement dans le but d'améliorer son apparence. En réalité, cela implique la création d'une nouvelle image en utilisant les valeurs de pixels de l'image d'origine. Plusieurs filtres sont appliqués, effectuant différents traitements en fonction des besoins de l'analyse. Un exemple de filtrage d'image est l'utilisation d'un filtre pour supprimer le bruit présent dans l'image en éliminant les pixels bruités[11].

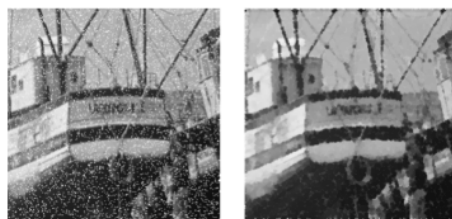


Figure I.14: Figure I.14: Exemple de filtrage d'une image bruitée (à gauche l'image d'origine, à droite l'image filtrée) [11].

I.6 Domaines D'application

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que :

I.6.1 Imagerie aérienne et spatiale

Dans laquelle les traitements concernent l'étude des images satellites, l'analyse des ressources terrestres, la cartographie automatique, les analyses météorologiques[10].

I.6.2 Technologies biomédicales

Nous trouvons des utilisations de cette technique dans l'échographie, la résonance magnétique nucléaire, ainsi que dans le domaine de la reconnaissance automatique des cellules ou des chromosomes[10].

I.6.3 La robotique

l'imagerie connaît un développement majeur dans plusieurs domaines, notamment l'assemblage de pièces mécaniques, les composants électroniques, le contrôle de qualité et la robotique mobile. Ces domaines tirent parti de l'imagerie pour effectuer des tâches telles que l'assemblage précis de pièces, la vérification de la qualité des produits et l'utilisation d'images pour la navigation et la perception dans les robots mobiles[10].

I.6.4 La télésurveillance

Exemple, radar automatique : recherche en temps réel d'un véhicule par reconnaissance de son immatriculation parmi un flot de véhicules circulant sur le boulevard périphérique par caméra fixe [10].

I.7 CONCLUSION

Dans ce chapitre, nous avons présenté une masse importante d'informations concernant le domaine de traitement d'image. Dans ce qui suit, nous allons entamer la détection d'objet ainsi que ses différentes techniques.

II.1 INTRODUCTION

La vision par ordinateur est une branche d'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle «voit». Elle peut servir à la reconnaissance de formes, qui consiste à reconnaître une forme dans une image après l'avoir enregistrée. La détection et le suivi d'objets est l'un des problèmes les plus étudiés ces dernières années. Ce sont des tâches importantes et difficiles dans de nombreuses applications de vision par ordinateurs tels que robots et vidéosurveillance. Dans ce chapitre, nous abordons les concepts de base en relation avec la détection des objets et les différentes techniques et algorithmes de Machine Learning et Deep Learning.

II.2 Définition de détection d'objets

La détection d'objets est un domaine de recherche dynamique qui vise à classifier et localiser des zones spécifiques d'une image ou d'une vidéo. Cette discipline se situe à l'intersection de deux autres domaines : la classification d'images et la localisation d'objets. Le principe de base de la détection d'objets consiste à identifier les régions de l'image susceptibles de contenir des objets, puis à extraire et classifier chacune de ces régions à l'aide d'un modèle de classification d'images. Les régions de l'image initiale qui obtiennent de bons résultats de classification sont conservées, tandis que les autres sont éliminées. Pour obtenir une méthode de détection d'objets efficace, il est essentiel de disposer d'un algorithme solide pour la détection des régions et d'un bon algorithme de classification d'images[17].

II.3 L'extraction des caractéristiques d'objet

L'objet peut être identifié par ses caractéristiques de forme, couleur et texture.

II.3.1 Forme

Les caractéristiques de forme sont importantes car elles fournissent une autre façon de décrire un objet, en utilisant ses caractéristiques les plus importantes et en réduisant la quantité d'informations stockées [18].

Histogramme de gradient orienté (HOG) L'idée importante derrière le descripteur HOG est que l'apparence et la forme locale des objets dans une image peuvent être décrites par la distribution d'intensité du gradient ou la direction des bords [?].

Dans notre projet, nous utilisons HOG comme caractéristiques pour la détection de forme (véhicule).

II.3.2 Couleur

Les caractéristiques de couleur nous permettent d'identifier la plupart des images et les objets incluent ceci. Par conséquent, ils sont définis comme des caractéristiques essentielles du contenu image [19].

L'histogramme des couleurs L'histogramme des couleurs est la technique la plus largement utilisée pour extraire la caractéristique couleur d'une image. Il représente l'image sous un autre angle. Il représente la distribution de fréquence des bacs de couleur dans une image. Il compte les pixels similaires et les stocke [20]. Ce travail utilise ce principe de description pour l'extraction des caractéristiques de couleur.

II.3.3 La texture

La texture est une caractéristique utilisée pour diviser une image en régions d'intérêt et pour classer ces régions. La texture fournit des informations sur la disposition spatiale des couleurs ou des intensités dans une image. Elle se caractérise par la distribution spatiale des niveaux d'intensité[21].

II.4 Apprentissage automatique(machine learning)

L'apprentissage automatique (Machine learning) est un domaine de recherche en informatique qui traite des méthodes d'identification et de mise en œuvre de systèmes et algorithmes par lesquels un ordinateur peut apprendre, ce domaine a souvent été associé

à l'intelligence artificielle et plus spécifiquement l'intelligence computationnelle [22]. Une définition qui s'applique à un programme informatique comme à un robot ou un être humain est celle proposée par Fabien Benureau [22]

L'apprentissage est une modification d'un comportement sur la base d'une expérience.

Le machine Learning peut servir à résoudre des problèmes :

- L'on ne sait pas résoudre (la prédiction d'achats).
- Que l'on sait résoudre, mais dont on ne sait formaliser en termes algorithmiques comment nous les résolvons (c'est le cas par exemple de la reconnaissance d'images ou de la compréhension du langage naturel).
- Que l'on sache résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques (c'est le cas par exemple de la prédiction d'interactions entre molécules de grande taille, pour lesquelles les simulations sont très lourdes).

Machine Learning englobe beaucoup de techniques où nous citons dans ce qui suit les plus utilisées.

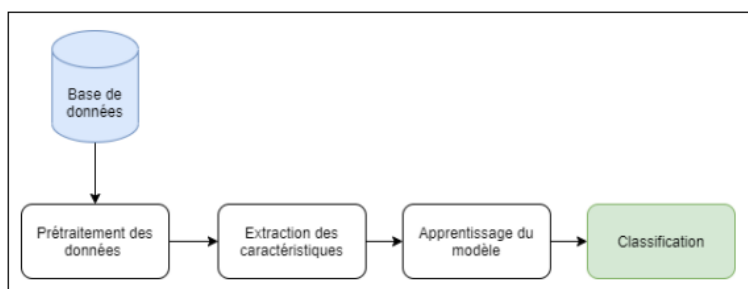


Figure II.1: Apprentissage Automatique ML[12].

II.4.1 Support Vector Machine (SVM)

C'est un algorithme d'apprentissage automatique supervisé, principalement utilisé pour la classification.

L'avantage de cet algorithme, par rapport aux autres algorithmes d'apprentissage automatique, est que non seulement il sépare les données en classes, mais il trouve aussi un hyperplan de séparation (l'analogie d'un plan dans un espace avec plus de trois dimensions) qui maximise la marge séparant chaque point de l'hyper-plan.

En outre, les machines vectorielles de support peuvent également traiter le cas des données qui ne sont pas linéairement séparables [23].

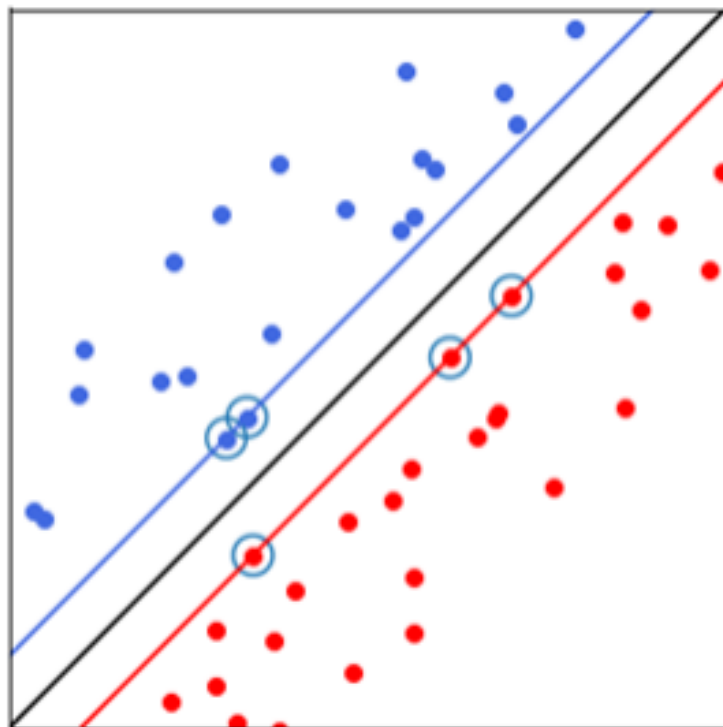


Figure II.2: Exemple illustrative d'un modèle SVM [12].

II.4.2 K-Nearest Neighbors (kNN)

L'algorithme des K plus proches voisins (kNN) est un algorithme d'apprentissage automatique supervisé qui repose sur des données d'entrée étiquetées qui permettent de définir une fonction par apprentissage, fonction qui va fournir une réponse appropriée lorsque de nouvelles données non étiquetées sont fournies [24].

II.5 Apprentissage profond (deep learning)

L'apprentissage profond (deep learning) est un ensemble de techniques d'apprentissage automatique qui ont fait de progrès significatifs en intelligence artificielle ces dernières années. en apprentissage automatique, un programme analyse un ensemble de données pour formuler des règles permettant de tirer des conclusions sur de nouvelles données[25]. Dans

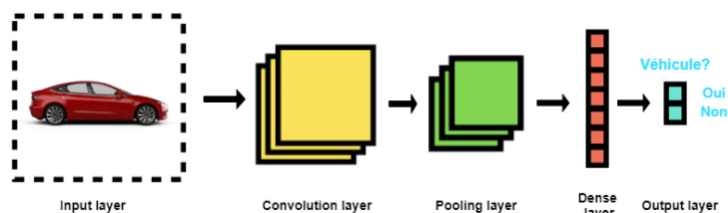


Figure II.3: Apprentissage profond DL[12].

ce qui suit, nous citons les algorithmes de Deep learning les plus utilisés.

II.5.1 Convolution Neural Network(CNN)

Est une classe de réseaux de neurones profonds, le plus souvent appliqué pour analyser l'imagerie visuelle. Maintenant, lorsque nous pensons à un réseau de neurones, nous pensons à des multiplications matricielles, mais ce n'est pas le cas avec le CNN. Il utilise une technique spéciale appelée Convolution. Or, en mathématiques, la convolution est une opération mathématique sur deux fonctions qui produit une troisième fonction qui exprime comment la forme de l'une est modifiée par l'autre[13].

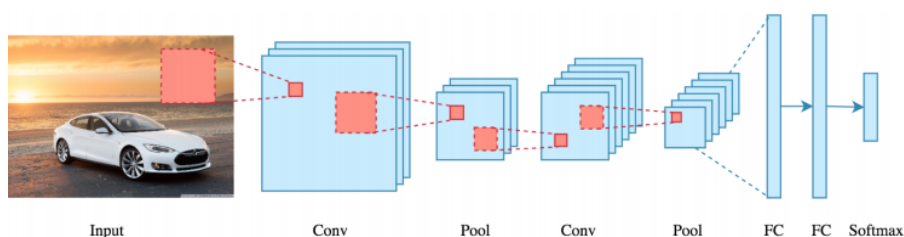


Figure II.4: Architecture générale du CNN [13].

II.5.2 You only look once : YOLO

La détection d'objets consiste en diverses approches telles que faster R-CNN rapide, RetinaNet . .etc. Bien que ces approches aient résolu les défis de la limitation des données et de la modélisation dans la détection d'objets, elles ne sont pas capables de détecter des objets en une seule exécution d'algorithme [26].

En vue que L'algorithme YOLO a gagné en popularité en raison de ses performances supérieures aux autres techniques de détection des objets, nous proposons de l'utiliser pour faire notre apprentissage profond. Conçu et continuellement amélioré par ses auteurs Joseph Redmon et Ali Farhadi de l'uni-versité de Washington [27].

You only look once(YOLO) est un algorithme de détection d'objets en temps réel qui identifie des objets spécifiques dans des vidéos, live feeds, ou des images. À la base, YOLO utilise un modèle CNN basé sur Googlenet (Szegedy et al., 2015) , qu'il nomme Darknet, avec modifications sur les couches d'entrée afin que s'effectuent le phases de détection et de classification à l'intérieur même du réseau, sans nécessiter l'utilisation d'un second réseau[28].

Le YOLO est actuellement en version 8 (yolov8) publié en 2023 [100] par Ultralytics .

II.6 Comparaison entre les versions YOLO

YOLO (You Only Look Once) est une famille d'architectures de réseaux neuronaux convolutifs utilisés pour la détection d'objets en temps réel. Chaque version de YOLO apporte des améliorations et des modifications par rapport à la version précédente. Voici les principales différences entre les différentes versions de YOLO :

	YOLO V1	YOLO V2	YOLO V3	YOLO V4	YOLO V5
Année	2015	2016	2018	2020	2020
Backbone	Darknet	Darknet	Darknet, MobileNetV1	CSPDarknet, CSPResNeXt	CSPDarknet
Couches	24	23	53	110-137	137-184
Détecteurs	1	5	9	11	3
Rendu en temps réel	Non	Oui	Oui	Oui	Oui
Précision	Moins précis que v2	Amélioration de la précision par rapport à v1	Meilleure précision par rapport à v2	Meilleure précision par rapport à v3	Variable en fonction des configurations
Objectifs	Détection d'objets	Détection d'objets, segmentation d'instance	Détection d'objets, segmentation d'instance, estimation de pose	Détection d'objets, segmentation d'instance, estimation de pose	Détection d'objets, segmentation d'instance

Table II.1: Les différences entr versions de YOLO (You Only Look Once) .

II.6.1 Les réseaux de capsules (Caps Nets)

Un réseau de capsules(caps net)est un système d'apprentissage automatique qui est un type de réseau de neurones artificiels qui peut être utilisé pour mieux modéliser les relations hiérarchiques. Cette approche vise à imiter plus étroitement l'organisation neuronale biologique.

L'idée est d'ajouter des structures appelées capsules à un réseau neuronal convolutif et de réutiliser les sorties de plusieurs de ces capsules pour former des représentations plus stables (par rapport à diverses perturbations) de capsules d'ordre supérieur. La sortie est un vecteur consistant en la probabilité d'une observation et une pose pour cette observa-

tion. Ce vecteur est similaire à ce qui est fait par exemple lors de la classification avec localisation dans les CNN [28].

II.6.2 Réseaux neuronaux récurrents(RNN)

Les RNN sont utilisés pour le traitement de séquences, tels que le langage naturel et la traduction automatique. Ils sont capables de prendre en compte les dépendances temporelles grâce à leur mécanisme de rétroaction [29].

II.6.3 Réseaux de neurones génératifs adverses (GAN)

Les GAN sont utilisés pour générer de nouvelles données réalistes, tels que des images ou des textes, en utilisant un générateur et un discriminateur qui s'affrontent dans un processus d'apprentissage[30].

II.7 déférence entre machine Learning et deep Learning

Dans cette partie du mémoire nous proposons notre point de vue comparatif entre les deux techniques machine learning et le deep learning. Le tableau suivant (Table II.2) illustre cette comparaison.

II.8 CONCLUSION

Dans ce chapitre nous avons exposé les techniques les plus connus de l'apprentissage automatique et profond. Dans le chapitre suivant nous procédons à l'implémentation de l'algorithme SVM (Machine Learning) et deux autres algorithmes YOLOv5 YOLOv8 (Deep Learning) afin de comparer et discuter leurs résultats.

	Machine Learning (apprentissage automatique)	Deep Learning (apprentissage profond)
Architecture des modèles	Utilise des algorithmes basés sur des caractéristiques extraites manuellement des données[31].	Utilise des réseaux de neurones profonds avec des architectures complexes et des millions de paramètres [32].
Représentation des données	Repose sur l'ingénierie des caractéristiques, les caractéristiques sont sélectionnées et extraites manuellement [31].	Apprend des caractéristiques discriminantes à partir des données brutes [32]..
Capacité d'apprentissage	Capacité à résoudre une grande variété de problèmes, mais peut être limitée par la complexité des modèles choisis [31].	Capable d'apprendre des représentations de données complexes et de résoudre des problèmes hautement non linéaires grâce à sa capacité à modéliser des relations hiérarchiques [32]..
Performance et complexité	Performances efficaces dans des tâches moins complexes [31]	Performances supérieures dans des tâches complexes, mais avec une complexité accrue [32]..
Interprétabilité	Les modèles sont souvent plus faciles à interpréter, car ils sont basés sur des algorithmes plus simples et transparents [31]	eut être plus difficile à interpréter en raison de la complexité des réseaux de neurones profonds et de leurs multiples couches de représentation [32].

Table II.2: les différences entre Machine learning et deep Learning

III.1 Introduction

Dans ce projet, j'ai implémenté trois algorithmes pour détecter la présence de véhicules dans le trafic routier. Les deux premiers algorithmes sont basés sur le Deep Learning, où j'ai utilisé les modèles YOLOV5 et YOLOV8 pour l'apprentissage approfondi. Le troisième algorithme est basé sur l'apprentissage automatique (Machine Learning), avec l'utilisation du modèle SVM.

Initialement, j'ai commencé la réalisation de cette application en utilisant le langage Python. Cependant, les performances de mon ordinateur ne m'ont pas permis d'obtenir des résultats satisfaisants en termes de taux d'apprentissage. Par conséquent, j'ai décidé de réfléchir à un environnement d'exécution virtuel, en optant pour Google Colab. Dans la suite, je présenterai l'environnement de programmation utilisé, les ressources et les bibliothèques mises en œuvre.

On a réalisé ces deux systèmes en utilisant le langage Python avec Jupyter. Toutefois, en ce qui concerne le système de Deep Learning, les performances de mon ordinateur ne nous ont pas permis d'obtenir des résultats satisfaisants en termes de taux d'apprentissage. Par conséquent, nous avons pris la décision de passer à un espace d'exécution virtuel, à savoir Google Colab.

Nous clôturons ce travail en présentant les résultats obtenus et en effectuant une compara-

ison entre les trois algorithmes implémentés.

III.2 Outil de développement

III.2.1 Jupyter

Jupyter [33] est une application web open-source qui permet de créer et de partager des documents qui contiennent du code en direct, des équations, des visualisations et du texte narratif. Les utilisations courantes incluent le nettoyage et la transformation de données, la simulation numérique, la modélisation statistique, la visualisation de données, l'apprentissage automatique et bien plus encore.

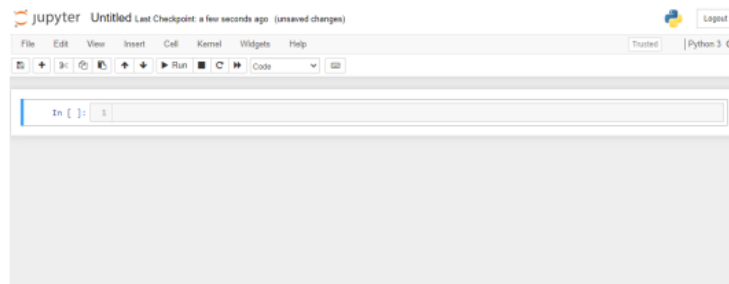


Figure III.1: Environnement de jupyter

III.2.2 Google Colab

Google Colab [34] est un service cloud de recherche, de développement et de formation en machine learning. Il offre un environnement Jupyter Notebook qui ne nécessite aucune configuration et qui s'exécute entièrement dans le cloud. Colab vous permet d'écrire et d'exécuter du code, de sauvegarder et de partager vos notebooks, ainsi que de profiter de ressources matérielles gratuites, notamment des processeurs graphiques (GPU) et des unités de traitement tensoriel (TPU).

Que nous offre Colab ?

Nous pouvons tirer profit les avantage suivants en utilisant Google Colab[35] :

- Écrire et exécuter du code en Python.

- Documentation du code pour soutenir les équations mathématiques.
- Facilite la Création, téléchargement et le partager des carnets et l'importation des ensembles de données (ex : a partir de Google).
- Intégrer Python, TensorFlow, Keras, et OpenCV.
- Service Cloud gratuit avec GPU gratuit.



Figure III.2: Environnement de Google Colab

III.2.3 Python

Python[36] est un langage de programmation interprété, orienté objet, de haut niveau, avec une sémantique dynamique. Sa syntaxe claire et expressive, ainsi que sa conception modulaire et orientée objet, en font un langage idéal pour les applications de script et de développement rapide.

III.2.4 Bibliothèques utilisées

III.2.4.1 OpenCV

Est une immense bibliothèque open source pour la vision par ordinateur, l'apprentissage automatique et le traitement d'images. OpenCV prend en charge une grande variété de langages de programmation comme Python, C++, Java, etc.

Il peut traiter des images et des vidéos pour identifier des objets, des visages, ou même l'écriture d'un humain. Lorsqu'il est intégré à diverses bibliothèques, comme Numpy qui est une bibliothèque hautement optimisée pour les opérations numériques, alors le nombre

d'armes augmente dans votre Arsenal, c'est-à-dire que toutes les opérations que l'on peut faire dans Numpy peuvent être combinées avec OpenCV [37].

III.2.4.2 SciPy

SciPy est une bibliothèque open-source pour le calcul scientifique en Python. Elle offre des fonctionnalités avancées pour la manipulation de tableaux multidimensionnels, l'optimisation numérique, l'algèbre linéaire, l'intégration numérique, l'interpolation, l'optimisation, le traitement du signal et bien d'autres domaines des mathématiques et de la science [38].

III.2.5 Matériel et limites

Nous avons utilisé pour l'implémentation de notre applicatiior un ordinateur **TOSHIBA satellite pro** ayant les caractéristiques suivantes :

- Système d'exploitation : Windows 10 .
- RAM : 4GB.
- CPU : Intel Core i5 3ème génération.

Nous tenons a mentionner que nous avons rencontré des limites lors de l'implémentation du Deep Learning :

- Problème de connexion .
- L'accès à google colab n'est pas disponible tout le temps. il nous limite dans le temps,si nous l'utilisons souvent il nous fait un ban 24h dans ce cas nous pouvons pas accéder au GPU.

III.3 Description de l'application

Le système englobe deux algorithmes pour détecter la présence des véhicules dans le trafic routier. Le premier algorithme est basé sur le Deep Learning ou nous proposons d'utiliser le modèle YOLOV5 et YOLOV8 pour faire l'apprentissage profond.Le deuxième algorithme

est basé sur l'apprentissage automatique (Machine Learning) ou nous avons adopté le modèle SVM.

III.3.1 Système basé sur deep Learning

Dans le système de détection des véhicules basé sur DL, nous formons un modèle de détection des véhicules personnalisé basé sur un modèle de détection des objets YOLOv5 et un modèle de détection des objets YOLOv8.

III.3.1.1 Algorithme 1(YOLOV5)

L'architecture de YOLO v5 est basée sur des couches de convolutions profondes, qui extraient des caractéristiques de l'image à différentes échelles et résolutions. Ces caractéristiques sont ensuite utilisées pour prédire les boîtes englobantes et les classes d'objets.

YOLOV5 utilise également des techniques d'augmentation de données pour améliorer les performances de détection et réduire le sur apprentissage.

Algorithme de YOLOv5:

- **Prétraitement:** cette étape englobe deux sous étapes la préparation de données et la Configuration du modèle.la première procède a annoter les images d'entraînement avec les boîtes englobantes et la classe d'objet (véhicule) à détecter. La deuxième est de sélectionner la taille appropriée du modèle YOLOv5 en fonction de nos besoins en termes de vitesse et de précision.
- **Entraînement du modèle:** Est le passage des images d'entraînement par le réseau de neurones YOLOv5, de calculer les prédictions des boîtes englobantes et des classes, et de comparer ces prédictions aux annotations réelles pour calculer la perte (loss). Le modèle est ensuite ajusté en utilisant l'algorithme de rétropropagation (backpropagation) afin d'améliorer progressivement les performances de détection.
- **Validation et ajustement :**Après chaque itération d'entraînement, il est essentiel de valider le modèle sur un ensemble de données de validation distinct pour évaluer ses performances. Si nécessaire, vous pouvez ajuster les hyperparamètres du modèle

(comme le taux d'apprentissage, la taille de la grille, etc.) pour obtenir de meilleurs résultats.

- **Évaluation du modèle** : Une fois que l'entraînement est terminé, vous pouvez évaluer les performances du modèle YOLOv5 sur un ensemble de données de test distinct. Cela permet de mesurer la précision, le rappel, la F-mesure et d'autres métriques pour évaluer la qualité de la détection d'objets.

L'architecture de YOLOv5

Est basée sur un réseau de neurones convolutifs profond avec une structure en "backbone" et une "head" de détection.

Les étapes de l'algorithme de DL basée sur YOLOV5 sont les suivantes :

- Colléction des données, et le choix des objets à détecter.
Dans ce cas nous détectons les véhicules.
- Configuration du YOLO avec la dataset : Maintenant que nous avons créer nos ensembles de données d'apprentissage et test, nous devons apporter des changements pour former notre modèle personnalisé de YOLO. Pour cela, nous choisissons le modèle de base le plus rapide de YOLOv5.

La figure(Figure III.3) suivante illustre les principales composantes de l'architecture YOLOv5 :

Alors l' algorithme se compose des etapes suivante :

- **Pretraitement** : redimensionnement de l'image pour quelle devienne .
- Apprentissage (YOLOv5)
- Evaluation du modèle sur les données de test.

III.3.1.2 Algorithme 2 (YOLOV8)

Les étapes impliquées dans un algorithme de détection d'objets utilisant YOLOv8 :

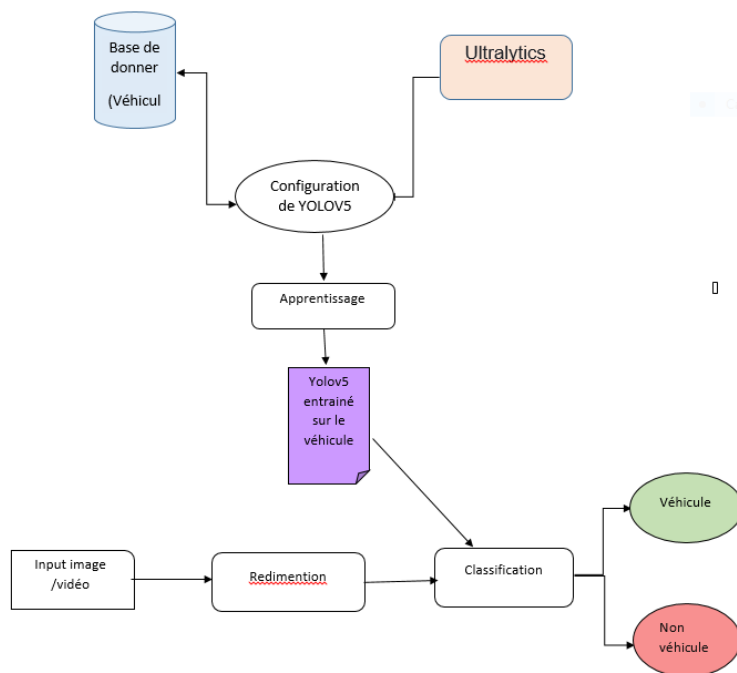


Figure III.3: L'architecture de modèle YOLOv5.

- **Collecte et préparation des données :** Rassemblez un ensemble de données d'entraînement contenant des images étiquetées avec des boîtes englobantes et des classes d'objets correspondantes. Les données peuvent également nécessiter des prétraitements tels que le redimensionnement, la normalisation et l'augmentation des données pour améliorer la variabilité et la qualité de l'entraînement.
- **Configuration du modèle :** Définissez l'architecture du modèle YOLOv8 en spécifiant les couches, les paramètres et les hyper-paramètres tels que la taille de l'image d'entrée, le nombre de classes, les paramètres de régularisation, etc.
- **Construction du modèle :** Créez le modèle YOLOv8 en empilant les couches convolutives, de regroupement (pooling), de normalisation et les couches spécifiques à YOLOv8, comme les prédicteurs d'ancres et les filtres de détection.
- **Initialisation des poids :** Initialisez les poids du modèle à l'aide de techniques telles que l'initialisation aléatoire ou le transfert de connaissances à partir de modèles pré-entraînés.
- **Entraînement du modèle :** Utilisez l'ensemble de données d'entraînement pour

entraîner le modèle YOLOv8 en utilisant des techniques d'apprentissage supervisé telles que la rétropropagation de l'erreur et l'optimisation du gradient. Cela implique le calcul de la perte (loss) entre les prédictions du modèle et les étiquettes réelles, et la mise à jour itérative des poids du modèle pour minimiser cette perte.

- **Validation et ajustement des hyperparamètres** : Évaluez les performances du modèle sur un ensemble de données de validation et ajustez les hyperparamètres tels que le taux d'apprentissage, la taille de lot (batch size), etc., pour améliorer les performances du modèle.
- **Évaluation du modèle** : Évaluez les performances du modèle sur un ensemble de données de test distinct pour mesurer la précision, le rappel, la précision moyenne (mAP) ou d'autres métriques d'évaluation spécifiques à la détection d'objets.
- **Inférence** : Utilisez le modèle entraîné pour effectuer des prédictions sur de nouvelles images ou vidéos en utilisant les méthodes de détection d'objets de YOLOv8, généralement basées sur la prédiction des boîtes englobantes et des classes d'objets. Les étapes de l'algorithme DL basée sur YOLOV8 sont les suivantes :

- Colléction des données, et le choix des objets à détecter. Dans ce cas nous détectons les véhicules.
- Configuration du YOLO avec la dataset : Maintenant que nous avons créer nos ensembles de données d'apprentissage et test, nous devons apporter des changements pour former notre modèle personnalisé de YOLO. Pour cela, nous choisissons le modèle de base le plus rapide de YOLOv8.

La figure(Figure III.3) suivante illustre les principales composantes de l'architecture YOLOv8 :

Alors notre algorithme se compose des etapes suivante :

- **Pretraitement** : redimensionnement de l'image pour quelle devienne .
- Apprentissage (YOLOv8)
- Evaluation du modèle sur les données de test.

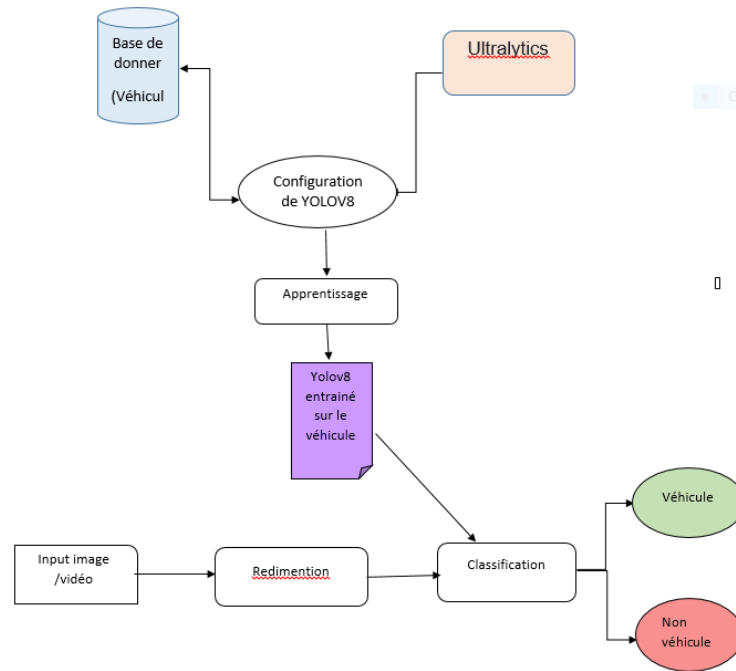


Figure III.4: L'architecture de modèle YOLOv8.

III.3.2 algorithme 3 (Apprentissage ML)

Dans ce système de détection des véhicules basé sur ML, nous avons adopté SVM avec un noyau linéaire pour l'apprentissage du modèle. Il est basé sur l'extraction et ensuite la combinaison de trois caractéristiques différentes :

HOG qui est la principale caractéristiques, l'**histogramme** de couleur et les caractéristiques **spatiales**. L'algorithme est basé sur les principales étapes suivantes :

- Redimensionnement des images de la base des images des véhicules à 64x64 pixels.
- Extraction des caractéristiques de HOG, histogramme de couleur et/ou caractéristiques spatiales ont été extraits pour chaque échantillon d'image.
- Combinaison des caractéristiques en un seul vecteur pour chaque image. Puis, la normalisation de ces vecteurs.
- Apprentissage du modèle avec un classifieur SVM avec un noyau linéaire.

Le schéma de description du système basé sur ML est illustré dans la figure III.5.

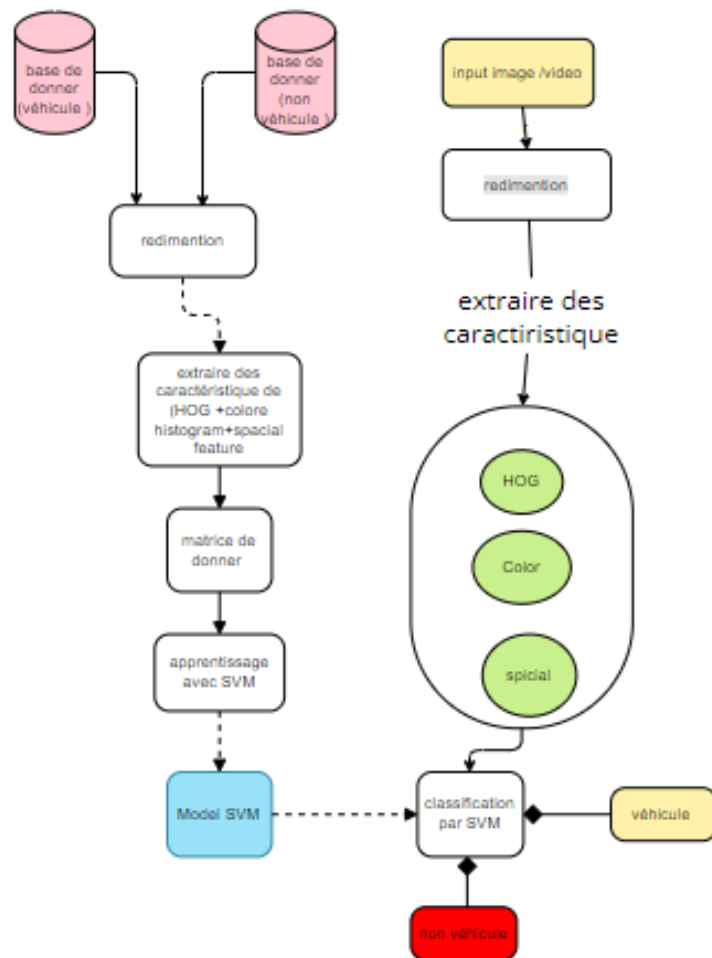


Figure III.5: Schéma de l'algorithme basé sur ML

III.3.2.1 Extraction des caractéristiques

Dans cette section, nous détaillons sur les trois types de caractéristiques que nous les extrairons depuis la base d'images (ou l'image d'entée).

Le descripteur final va permettre d'extraire 5168 caractéristiques.

Un exemple d'exécution d'un système de Machine Learning dans un environnement Jupyter Notebook.

La figure III.6 c'est un extraction des caractéristiques .

La figure III.7 c'est un Normalisation des caractéristiques .

HOG C'est la caractéristique principale dans notre algorithme basé sur ML. Pour une image d'entrée, nous travaillons avec le descripteur HOG qui possède les paramètres décrit

```

8792 Le nombre d'images positives (véhicules) et 8968 négative (non-véhicules) trouvé .
feature extraction ....
17760it [00:44, 395.08it/s]
Extraction des caractéristiques à partir de 17760 images dans 45.2 seconds

```

Figure III.6: Extraction des caractéristiques

```

Normalisation des caractéristiques

7034 images dans la base d'apprentissage des véhicules.
1758 images dans la base des tests des véhicules.
8792 totale images de véhicules.

7174 images dans la base d'apprentissage des non-véhicules.
1794 images dans la base des tests des non-véhicules.
8968 totale images de non-véhicules.

Apprentissage du classifieur...
Classifieur entraîné dans 6.2 s.

```

Figure III.7: Normalisation des caractéristiques

dans le tableau suivant :

Paramètre	Valeur
Taille de cellule	8x8
Taille de bloc	2x2
HOG classes (orientations)	20

Table III.1: Paramètres de HOG

Chaque pixel d'une cellule de 8x8 pixels contribue au vote pour l'une des 20 classes de l'histogramme, en fonction de l'orientation du gradient à ce point. Le poids du vote du pixel est déterminé par l'intensité du gradient en ce point. Afin de réduire les disparités causées par les variations d'illumination, nous regroupons plusieurs cellules dans un bloc de 2x2 cellules.

Lorsque nous travaillons avec une image d'entrée de taille 64x64, nous parcourons les blocs horizontalement et verticalement. Chaque bloc est composé de 4 cellules. En excluant le dernier bloc, le nombre total de blocs est égal à $(8 \text{ blocs} - 1) * (8 \text{ blocs} - 1)$. Pour chaque cellule, nous obtenons un vecteur de taille 20 en votant pour une classe d'histogramme en fonction de l'orientation du gradient de ses pixels. En multipliant le nombre de blocs par le nombre de cellules par bloc par la taille du vecteur de classe (20), nous obtenons un vecteur final de taille 3920. Ce vecteur capture les informations importantes extraites de



Figure III.8: l'application de HOG.

l'image pour une analyse plus poussée ou une classification ultérieure.

Histogramme de couleurs Dans cette approche, nous avons appliqué un histogramme de 16 classes à chacun des trois canaux de couleur (Bleu, Vert, Rouge) de l'image d'entrée, qui est représentée dans l'espace de couleur BGR. Par conséquent, le deuxième descripteur nous a permis d'extraire un total de 48 caractéristiques (16 classes multipliées par les 3 canaux), soit 16 caractéristiques pour chaque canal de couleur.

Caractéristiques spatiales En appliquant le regroupement spatial, nous réduisons la taille de l'image d'entrée de 64x64 à 20x20. Les valeurs des pixels de cette image sont utilisées comme caractéristiques spatiales pour chaque canal de couleur. Ainsi, nous obtenons un total de **1200** ($20 \times 20 \times 3 = 1200$) **caractéristiques spatiales**. Ces caractéristiques représentent des informations sur la répartition spatiale des couleurs dans l'image réduite.

III.3.2.2 Apprentissage

Nous utilisons l'algorithme supervisé Support Vector Machine (SVM) pour l'apprentissage du modèle de classification. Le noyau de SVM sera linéaire avec 1000 comme valeur de hyperparamètre C.

III.4 Présentation du Système

Notre système est composé de trois espaces : espace YOLOV5, YOLOV8 et Machine Learning .

Dans ce qui suit, nous allons donner un scénario d'utilisation des différentes modèle de notre système.

III.4.1 Intarface

La figure III.9 suivante présente l'interface principale de système de detection .

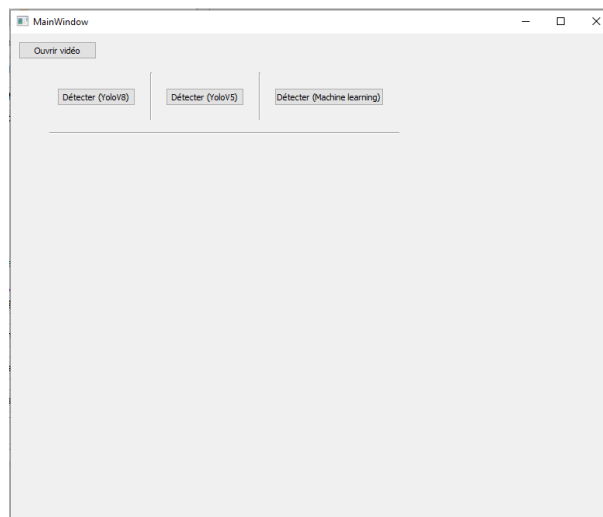


Figure III.9: L'intarface de l'application

III.4.2 l'espace de YOLOV5

La figure III.10 suivante présente l'espace de detection avac modèle de YOLOV5.

III.4.3 l'espace de YOLOV8

La figure III.11 suivante présente l'espace de detection avac modèle de YOLOV8.

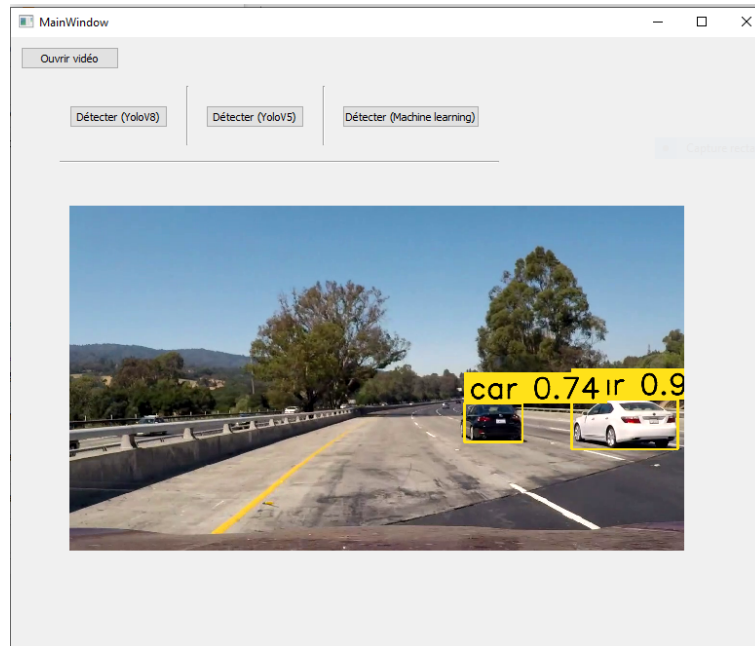


Figure III.10: détection avec YOLOV5 .

III.4.4 l'espace de Machine Learning

La figure III.12 suivante présente l'espace de detection avac machine Learning (SVM).

III.5 Expérimentation et résultats

Pour l'expérimentation de notre système, nous avons comparé les performances des deux systèmes implémentés.

La clé d'une comparaison équitable des algorithmes de ML/DL est de s'assurer que chaque algorithme est évalué de la même manière sur les mêmes données.

Le critère de comparaison utilisé ici est le taux de reconnaissance (accuracy) calculé par la formule suivante :

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

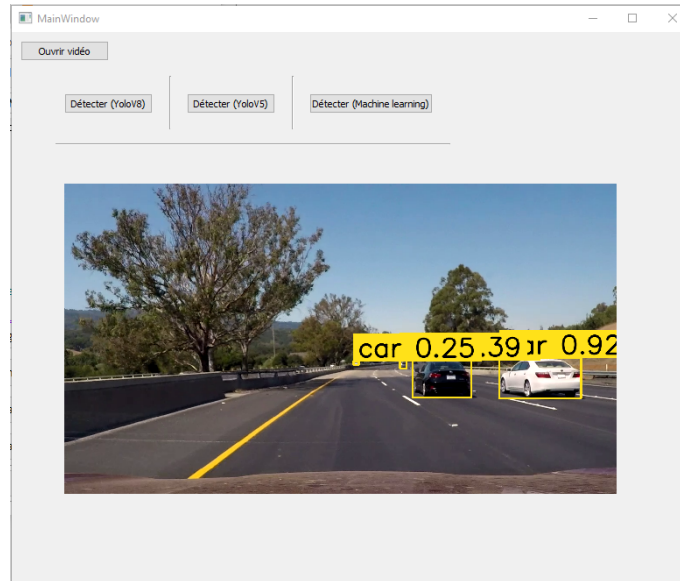


Figure III.11: détection avec YOLOV8.

III.6 Bases de Données

Le modèle sera entraîné sur tous les images de la base de données Udacity[39] .

Elle contient 8799 images de véhicules vus sous différents angles et 8971 images de non-véhicules (route, autoroutes, garde-corps, etc.). Ces exemples d'images proviennent d'une combinaison de la base de données d'images de véhicules GTI [?] et KITTI [40].

III.7 Expérimentation

Après avoir mélanger les données, nous les divisons en :

- 80% données d'apprentissage (trainSets)
- 20% données de test (testSets).

Après l'entraînement du modèle sur les données d'apprentissage (trainSets), on exécute le modèle sur les données de test (testSets)

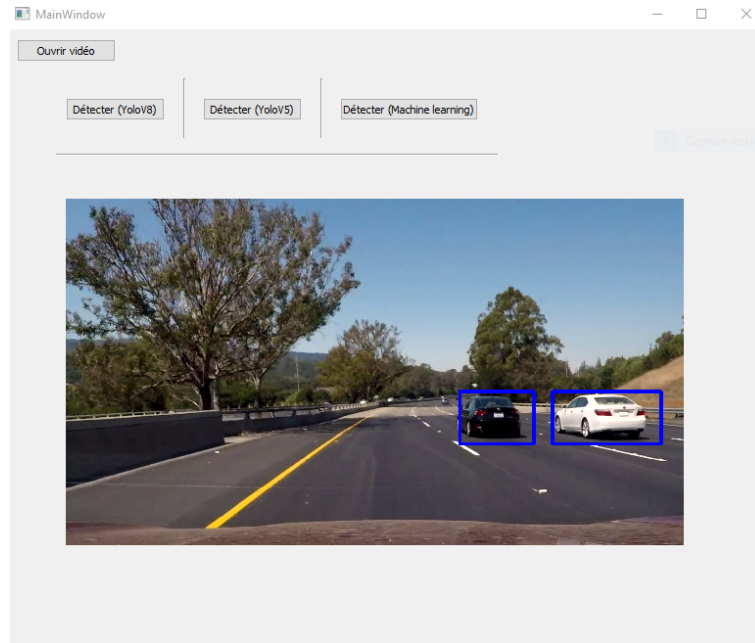


Figure III.12: détection avec ML

III.8 Résultats

III.8.1 Résultats de Machine Learning

Les résultats de l'évaluation du système ML proposé sont présentés dans le tableau III.3 les critères utilisés pour la comparaison comprennent la taille du vecteur de caractéristiques, le temps d'extraction des caractéristiques, le temps d'apprentissage et le taux de reconnaissance (Accuracy). Les performances de notre algorithme de classification SVM peuvent être résumées dans la matrice de confusion, qui est illustrée dans le tableau III.2.

	Non-véhicule	véhicule
Non-véhicule	1745	13
véhicule	31	1763

Table III.2: Matrice de confusion du système ML

Selon ces résultats, nous pouvons conclure les remarques suivantes :

- La combinaison des trois caractéristiques de HOG, histogramme de couleur, et les caractéristiques spatiales permet d'augmenter la performance de notre système ML, avec un taux de 98,76%.

	Taille du vecteur des carac.	Temps d'extraction des carac.	Temps d'apprentissage	Accuracy
HOG	3920	18.5s	10.2s	96.56%
Color	48	57.5s	0.6s	86.20%
Spacial	1200	17.2s	39.1s	90.48%
HOG + Color	3968	52.2s	8.8s	97.83%
HOG + Spacial	5120	19.4s	10.8s	98.25%
Color + Spacial	1248	51.8s	87.4s	95.15%
HOG Color + Spacial	5168	79.3s	12.2s	98.76%

Table III.3: Résultats du notre système ML

- Deux caractéristiques sur trois sont les plus fortes : HOG et spacial. La combinaison de ces deux donne un taux de 98.28% proche de l'optimale.
- L'histogramme de couleur n'augmente pas trop la performance, contrairement à HOG qui est le plus important (taux de 96.56% si nous l'utilisons seul).
- L'extraction des caractéristiques de l'histogramme de couleur est la plus lente.

Avec le système Deep Learning. Nous avons beaucoup de résultats à mettre dans ce mémoire.

Cependant, nous avons trouvé un taux de reconnaissance égale à 98.81%.

III.8.2 Résultats de Deep Learning

III.8.2.1 Résultats de YOLOV5

Les résultats de l'évaluation du système DL avec modèle de YOLOV5 sont présentés dans le tableau III.4.

	prétraiter	largeur	hauteur	fps	frames	accuracy
video 1	3.6ms	3840	2160	25	538	80%
video 2	37.8ms	1920	1080	23	332	50%

Table III.4: Résultats de YOLOV5

III.8.2.2 Résultats de YOLOV8

Les résultats de l'évaluation du système DL avec modèle de YOLOV8 sont présentés dans le tableau III.5.

	prétraiter	largeur	hauteur	fps	frames	accuracy
video 1	12,6ms	3840	2160	25	538	80%
video 2	37.8ms	1920	1080	23	332	50%

Table III.5: Résultats de YOLOV8

Une comparaison entre nos deux systèmes est-il raisonnable ? Malgré que le taux de reconnaissance dans les deux systèmes est presque le même, la comparaison entre ces deux systèmes n'est pas raisonnable. Le système à base DL est clairement surperformé car notre modèle Custom YOLOv3 est déjà préentraîné sur la base de données COCO [41] (parmi eux, il y a 12786 images contenant des véhicules).

III.9 Conclusion

Dans ce chapitre, nous avons décrit les trois conceptions que nous avons proposées pour la détection des véhicules. Pour les deux architectures de Deep Learning nous avons personnalisé le modèle de YOLOv5 et YOLOV8. Mais concernant le machine learning, nous avons combiné trois types de caractéristiques, pour la détection des véhicules.

Ensuite, nous avons implémenté nos deux systèmes pour faire l'expérimentation. À partir de cette dernière, on peut conclure que le modèle YOLOv5 ou YOLOv8 est performant par rapport au modèle ML, car il est déjà préentraîné sur un grand nombre de véhicules.

Dans le système basé sur ML, on peut aussi conclure que la combinaison des trois caractéristiques améliore la performance du modèle SVM et surtout la combinaison de HOG et les caractéristiques spatiales, avec moins d'influence de l'histogramme de couleurs.

CONCLUSION GENERAL

Dans ce projet de fin d'études, nous nous sommes intéressés à la détection des véhicules dans le trafic routier et avons mis en œuvre deux systèmes : le premier basé sur l'apprentissage automatique (Machine Learning) et le deuxième basé sur l'apprentissage profond (Deep Learning). Les expérimentations réalisées nous ont permis de tirer les conclusions suivantes:

- La combinaison de différents types de caractéristiques, notamment le HOG (Histogram of Oriented Gradients) et les caractéristiques spatiales, permet d'améliorer la détection sous l'apprentissage automatique.
- Les modèles entraînés avec le Deep Learning présentent de meilleures performances en raison de l'utilisation de volumes de données massifs lors de l'entraînement.

En ce qui concerne les perspectives de recherches futures, nous envisageons :

- D'utiliser d'autres types de caractéristiques pour améliorer la détection avec l'apprentissage automatique.
- De détecter d'autres objets dans le trafic routier.

Les connaissances acquises au cours de ce projet de fin d'études se situent dans deux dimensions :

- **Sur le plan théorique :** Nous avons acquis une maîtrise des concepts de base dans les domaines de l'apprentissage automatique et du traitement d'images, qui étaient nouveaux pour nous et n'ont pas été étudiés au cours de nos trois années de licence.

- **Sur le plan pratique :** En plus d’approfondir nos connaissances en langage Python, nous avons appris à utiliser certaines fonctions des bibliothèques OpenCV, NumPy et scikit-learn. Nous avons également acquis une expérience dans l’utilisation de Google Colab pour l’implémentation du système de Deep Learning, en raison de sa performance lors de la phase d’apprentissage. L’implémentation de nos systèmes constitue une application concrète de plusieurs connaissances en intelligence artificielle (apprentissage automatique, Deep Learning), en traitement d’images, en algorithmique et en paradigmes des langages de programmation, pour n’en citer que quelques-uns.

REFERENCES

- [1] T. SADOUN and S. BEKKOUCHE, “Filtrage d’images numériques niveau de gris,” 2016.
- [2] F. BAHADJ, M. BELLAOUI, *et al.*, *Extraction de la Température de Surface de la Terre à Partir d’une Image Satellitaire*. PhD thesis, Université Ahmed Draia-ADRAR, 2015.
- [3] M. J. d. L. Magalhães, *Augmented reality over maps*. PhD thesis, 2020.
- [4] J. Weber, *Segmentation morphologique interactive pour la fouille de séquences vidéo*. PhD thesis, Université de Strasbourg, 2011.
- [5] H. Yacine and M. Silia, *Véhicule intelligent pour la détection des plaques d’immatriculation suspectes*. PhD thesis, Université Mouloud Mammeri, 2019.
- [6] M. R. B. E. L. Taha and M. A. L. Aymen, *Détection et lecteur de Qr code avec OpenCV*. PhD thesis, Faculté des Sciences et Technologies, 2021.
- [7] O. Lezoray, *Segmentation d’images par morphologie mathématique et classification de données par réseaux de neurones: Application à la classification de cellules en cytologie des séreuses*. PhD thesis, Université de Caen Basse-Normandie, 2000.
- [8] A. Zerougui and N. Sari, “Traitement d’images monochromes,” 2017.
- [9] J.-L. Le Quellec, J. Harman, C. Defrasne, and F. Duquesnoy, “Dstretch® et l’amélioration des images numériques: applications à l’archéologie des images rupestres,” *Les Cahiers de l’AARS*, vol. 16, pp. 177–198, 2013.
- [10] L. BERREZOUG and A. BENYAGOUB, *Utilisation des Algorithmes de L’Intelligence Artificielle appliquer à l’Aide à la Conduite*. PhD thesis, Directeur: M. ABDELLAOUI Ghouti, 2021.

- [11] R. AHMED, “Stéganographie d’images à l’aide de l’apprentissage automatique,”
- [12] B. Khadidja, *Interface Design for Human Pose Estimation*. PhD thesis, Université mohamed boudiaf des sciences et de la technologie d’oran, 2015.
- [13] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Mathematics and computers in simulation*, vol. 177, pp. 232–243, 2020.
- [14] A. Marion, *Introduction aux techniques de traitement d’images*. Eyrolles, 1987.
- [15] S. Bahroun, *Contributions à la construction de résumés Vidéos: application à la recherche d’objets génériques et à la reconnaissance faciale*. PhD thesis, faculté des sciences de tunis, université de tunis el manar, 2023.
- [16] M. Nadia, *ANALYSE DES IMAGES MEDICALES PAR DES TECHNIQUES HYBRIDES*. PhD thesis, Université mohamed boudiaf des sciences et de la technologie d’oran, 2014.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [18] G. Farin, B. Hamann, and H. Hagen, *Hierarchical and geometrical methods in scientific visualization*. Springer Science & Business Media, 2003.
- [19] A. J. Afifi and W. M. Ashour, “Image retrieval based on content using color feature,” *International Scholarly Research Notices*, vol. 2012, 2012.
- [20] D. Srivastava, R. Wadhvani, and M. Gyanchandani, “A review: color feature extraction methods for content based image retrieval,” *International Journal of Computational Engineering & Management*, vol. 18, no. 3, pp. 9–13, 2015.
- [21] L. Mărginean, P. A. Ștefan, A. Lebovici, I. Opincariu, C. Csutak, R. A. Lupean, P. A. Coroian, and B. A. Suciu, “Ct in the differentiation of gliomas from brain metastases: The radiomics analysis of the peritumoral zone,” *Brain Sciences*, vol. 12, no. 1, p. 109, 2022.
- [22] G. Zacccone and M. R. Karim, “Ahmed menshawy deep learning with tensorflow,” *Birmingham: Packt Publishing*, 2017.
- [23] I. BAHRI, M. LAIB, M. El, *et al.*, *COMBINAISON SVM ET APPRENTISSAGE PROFOND POUR LA RECONNAISSANCE DE CARACTERES ARABE*. PhD thesis, University Ahmed DRAIA of Adrar, 2022.

- [24] M. HAMMOUDI and R. A. KECHRA, *Systèmes d'Identification Automatique des Véhicules*. PhD thesis, Université Ibn Khaldoun-Tiaret-, 2022.
- [25] M. Boughaba, B. Boukhris, and M. Meflah, “L'apprentissage profond (deep learning) pour la classification et la recherche d'images par le contenu,” 2017.
- [26] G. Karimi, “Introduction to yolo algorithm for object detection,” *Section. io*. [https://www. section. io/engineering education/introduction-to-yolo-algorithm-for-objectdetection/](https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-objectdetection/)(accessed May 7, 2021), 2021.
- [27] J. Terven and D. Cordova-Esparza, “A comprehensive review of yolo: From yolov1 to yolov8 and beyond,” *arXiv preprint arXiv:2304.00501*, 2023.
- [28] F. S. do Couto, V. L. Batalha, J. S. Valadas, J. Data-Franca, J. A. Ribeiro, and L. V. Lopes, “Escitalopram improves memory deficits induced by maternal separation in the rat,” *European journal of pharmacology*, vol. 695, no. 1-3, pp. 71–75, 2012.
- [29] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.
- [30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [31] T. M. Mitchell, “Does machine learning really work?,” *AI magazine*, vol. 18, no. 3, pp. 11–11, 1997.
- [32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [33] L. A. Barba, L. J. Barker, D. S. Blank, J. Brown, A. B. Downey, T. George, L. J. Heagy, K. T. Mandli, J. K. Moore, D. Lippert, *et al.*, “Teaching and learning with jupyter,” *Recuperado: https://jupyter4edu.github.io/jupyter-edu-book*, pp. 1–77, 2019.
- [34] D. K. Barupal, “Response: Commentary: Data processing thresholds for abundance and sparsity and missed biological insights in an untargeted chemical analysis of blood specimens for exposomics,” *Frontiers in Public Health*, vol. 10, p. 1003148, 2022.
- [35] I. BOULEFRAKH, “La segmentation d'organes à partir d'images médicales,” 2022.
- [36] G. Van Rossum, B. Warsaw, and N. Coghlan, “Pep 8–style guide for python code,” *Python. org*, vol. 1565, p. 28, 2001.

- [37] W. Nitinuntatip, K. Wattanasuk, A. Dersingh, J. Rojanarowan, W. Plodpradista, J. Kongthon, V. Patanavijit, A. Bhattarai, and M. E. Ali, “Object recognition system for visual impairment,” 2022.
- [38] M. Visani, *Contributions à l’analyse d’images par catégorisation pour la description et la reconnaissance*. PhD thesis, Université de La Rochelle, 2014.
- [39] H. N. M. Nguyen, X. H. Phan, and Q. L. Tran, “Deep learning based detection of vehicles, lane and street sign for behavior cloning in autonomous car,” tech. rep., VNU-UET, 2018.
- [40] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.
- [41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.