

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE DE  
LA MAÎTRISE EN MATHÉMATIQUES ET  
INFORMATIQUE APPLIQUÉES

PAR

PAPE DEMBA BEYE

FAISABILITÉ TECHNIQUE DES SYSTÈMES AVANÇÉS D'AIDE À LA  
CONDUITE (ADAS) POUR LA SÉCURITÉ ROUTIÈRE

Décembre 2021

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## **Remerciement**

En premier je remercie Dieu de m'avoir donné la santé et le courage afin de surmonter les obstacles pour réaliser ce mémoire. Je remercie aussi mes parents pour leur soutien moral et financier.

Je tiens à remercier mon directeur de recherche, le professeur Fathallah Nouboud, qui m'a aidé durant toutes les étapes de ce projet de recherche. Merci de m'avoir soutenu, encadré et conseillé tout au long de mes études. Merci de m'avoir accorder de la confiance en m'offrant une grande autonomie dans mon travail. Vos conseils m'ont beaucoup aidé à réaliser ce feux e mémoire.

J'adresse également mes sincères remerciements à tous les membres du département de Mathématiques et Informatique de l'Université du Québec à Trois-Rivières (UQTR) qui ont toujours été disponibles pour m'aider à réaliser mon cursus universitaire au programme de maîtrise en mathématiques et informatique appliquées.

## **Dédicace**

Je dédicace ce mémoire à ma mère qui a été ma première éducatrice, à mon père qui a été une figure importante dans ma vie, à mes frères et sœurs pour leurs encouragements.

## Résumé

Durant les dernières années, le développement du domaine de la vision par ordinateur a poussé les industries automobiles à créer des systèmes d'aide à la conduite qui permettent de réduire les accidents de la route d'une manière significative. Dans ce mémoire, nous proposons un système de détection des objets de la circulation routière et d'estimer leur distance par rapport à la caméra. Notre système localise et reconnaît des objets (voitures, camions, piétons, etc.) situés en face du véhicule. Afin de déterminer avec précision la nature de chaque objet, un algorithme d'apprentissage profond a été utilisé, il avertit aussi le conducteur de l'existance des feux de la circulation et les panneaux STOP. Par la suite, il estime la distance entre la caméra intégrée et l'objet détecté. La combinaison de ces différentes informations permet aux systèmes d'aide à la conduite une meilleure perception de l'entourage.

Les mots-clés sont: la vision par ordinateur, la détection des objets, l'estimation de distance et les systèmes d'aide à la conduite.

## Listes des figures

<b>Figure 1. 1.</b>	
Représentation d'image numérique [1]. .....	4
<b>Figure 1. 2.</b>	
Le groupe de pixels forme la lettre A [2]. .....	5
<b>Figure 1. 3.</b>	
Représentation d'un histogramme d'une image sous Matlab avec $H(x)$ .....	7
<b>Figure 1. 4.</b>	
Filtrage d'une image par un noyau de convolution [11].....	10
<b>Figure 1. 5.</b>	
L'application des filtrages linéaire [5]. .....	11
<b>Figure 1. 6.</b>	
L'application des filtrages non linéaire [5]. .....	12
<b>Figure 1. 7.</b>	
La segmentation d'une image [6]. .....	12
<b>Figure 1. 8.</b>	
La segmentation basée sur les pixels [2]. .....	13
<b>Figure 1. 9.</b>	
Détection des contours sur Lena [5]. .....	14
<b>Figure 1. 10.</b>	
Illustration qui montre l'architecture de Deep Learning [10]. .....	15
<b>Figure 1. 11.</b>	
Architecture et composition d'un réseau des neurones convolutifs [7]. .....	16
<b>Figure 1. 12.</b>	
Illustration de l'opération de convolution entre une image et un filtre.....	18
<b>Figure 1. 13.</b>	
Illustration de l'opération de pooling .....	17
<b>Figure 1. 14.</b>	
Illustrations de la différence entre la classification et la detection.....	22
<b>Figure 1. 15.</b>	
Architecture de modèle Faster R-CNN [24]. .....	26
<b>Figure 1. 16.</b>	
L'illustration de l'architecture de Mask R-CNN [25]. .....	27
<b>Figure 1. 17.</b>	
Modèle YOLO [26]. .....	28
<b>Figure 2. 1.</b>	
Les catégories d'un ADAS [35]. .....	33
<b>Figure 2. 2.</b>	
Image RVB d'une caméra monoculaire [38]. .....	36
<b>Figure 2. 3.</b>	
Une image de caméra stéréoscopique [35]. .....	37
<b>Figure 2. 4.</b>	
Une capture d'une image à partir d'une caméra thermique passive.....	38
<b>Figure 2. 5.</b>	
La différence entre la caméra thermique, la caméra active et la caméra monoculaire [40]. .....	38
<b>Figure 2. 6.</b>	
Une représentation des applications d'ADAS [62]. .....	41
<b>Figure 2. 7.</b>	
Le système de détection des piétons par une caméra .....	42
<b>Figure 2. 8.</b>	
L'affichage de guidage de stationnement sur .....	43

<b>Figure 2. 9.</b>	
Système de détection et de la reconnaissance des panneaux de signalisationé.....	42
<b>Figure 2. 10.</b>	
Alerte de somnolence lorsque le seuil passe la 4e foisdetections.....	43
<b>Figure 3. 1.</b>	
Architecture et composition de SSD [53]. .....	53
<b>Figure 3. 2.</b>	
L'architecture de réseau de base (VGG-16) avec une détecteur SSD [27]. .....	53
<b>Figure 3. 3.</b>	
La visualisation des cartes des caractéristiques CNN et du champ réceptif [54]. .....	54
<b>Figure 3. 4.</b>	
L'utilisation d'un filtre de 3*3 pour la localisation et la classification [27]. .....	54
<b>Figure 3. 5.</b>	
Utilisation des cartes d'entités de différentes couches pour la détection [27]. .....	55
<b>Figure 3. 6.</b>	
Utilisation de plusieurs boites par défaut dans une seule cellule [27]. .....	56
<b>Figure 3. 7.</b>	
Diagramme d'explication d'IoU (l'indice de Jaccard) [55]. .....	57
<b>Figure 3. 8.</b>	
La conception en générale du notre système. .....	58
<b>Figure 3. 9.</b>	
Diagramme de réseau de base de modèle SSD300.....	58
<b>Figure 3. 10.</b>	
Addition des couches de localisation et classification au réseau de base. .....	59
<b>Figure 3. 11.</b>	
Le diagramme de la phase d'apprentissage de système. .....	60
<b>Figure 3. 12.</b>	
La suppression de non maximum [54]. .....	61
<b>Figure 3. 13.</b>	
Le diagramme représente les étapes de la construction. .....	62
<b>Figure 3. 14.</b>	
Le diagramme représente la lecture du vidéo et le prétraitement. .....	63
<b>Figure 3. 15.</b>	
Le diagramme représente le processus de détection des objets et l'estimation de la distance.....	64
<b>Figure 4. 1.</b>	
La création de la copie. .....	68
<b>Figure 4. 2.</b>	
La sélection des couches et l'extraction des filtres et des biaises. .....	70
<b>Figure 4. 3.</b>	
Remplacer les filtres et les tenseurs dans le modèle de la destination.....	70
<b>Figure 4. 4.</b>	
Les couches de classification dans le modèle des poids SSD300 pré entrainé sur MS-COCO....	71
<b>Figure 4. 5.</b>	
Les couches de classification de notre modèle SSD300. .....	71
<b>Figure 4. 6.</b>	
L'interface principale de système. .....	72
<b>Figure 4. 7.</b>	
Les taches de système .....	72
<b>Figure 4. 8.</b>	
Le choix d'une image. .....	73
<b>Figure 4. 9.</b>	
Le choix d'un enregistrement vidéo. .....	74

<b>Figure 4. 10.</b>	
La fonction d'estimation de la distance. ....	75
<b>Figure 4. 11.</b>	
L'affichage de notre système.....	75
<b>Figure 4. 12.</b>	
La détection des objets et l'estimation de distance dans la première situation. ....	76
<b>Figure 4. 13.</b>	
La détection des objets dans la deuxième situation.....	76

## Table des Matières

Remerciement .....	ii
Dédicace .....	ii
Résumée.....	iii
Listes des figures .....	iv
Tableau de Matière.....	vii
Introduction générale .....	- 1 -
CHAPITRE 1 : Le traitement d'image et la vision par ordinateur .....	3
1.1. Introduction.....	3
1.2. Le traitement d'images .....	4
1.2.1. Définition de l'image numérique .....	4
1.2.2. Les caractéristiques des images numériques .....	4
1.2.2.1. Le pixel .....	4
1.2.2.2. La résolution .....	5
1.2.2.3. La dimension .....	5
1.2.2.4. La profondeur .....	6
1.2.2.5. Le poids de l'image .....	6
1.2.2.6. La texture .....	6
1.2.2.7. Le bruit.....	6
1.2.2.8. La luminance .....	6
1.2.2.8. Histogramme.....	7
1.2.2.9. Le contraste.....	7
1.2.3. Les formats standards d'image .....	8
1.2.3.1. L'image matricielle.....	8
1.2.3.2. L'image vectorielle .....	8
1.2.4. Les types des images .....	8
1.2.4.1. L'image binaire.....	8

1.2.4.2. L'image en niveaux de gris.....	9
1.2.4.3. L'image couleur.....	9
1.2.4.4. L'image a valeur réelle .....	9
1.2.5. Les principales techniques de traitement des images .....	9
1.2.5.1. Acquisition.....	9
1.2.5.2. Le filtrage.....	10
1.2.5.2.1. Filtrage linéaire.....	10
1.2.5.2.1. Filtrage non linéaire .....	11
I.2.5.3. La segmentation .....	12
I.2.5.3.1. Segmentation basée sur les pixels .....	12
Figure 1. 8. La segmentation basée sur les pixels [2].....	13
I.2.5.3.3. Segmentation basée sur les contours .....	13
1.3. La vision par ordinateur.....	14
1.3.1. Définition.....	14
1.3.2. L'apprentissage automatique .....	14
1.3.3. L'apprentissage profond .....	15
1.3.3.1. Les réseaux de neurones convolutifs .....	16
1.3.3.1.1. Les couches d'un Réseau Neurones Convolutives .....	1817
1.3.3.1.1.1. La couche de convolution (CONV).....	17
1.3.3.1.1.2. La couche de correction (Relu).....	18
1.3.3.1.1.3.. La couche de pooling (POOL).....	18
1.3.3.1.1.4. La couche entièrement connectée (FC) .....	19
1.3.3.1.1.5. La couche de perte (LOSS, Softmax) .....	19
1.3.3.1.2. Les architectures de CNN .....	19
1.3.3.1.2.1. LeNet .....	19
1.3.3.1.2.2. ZFNet.....	19
1.3.3.1.2.4. GoogleNet.....	20

1.3.3.1.2.4. VGGNet .....	20
1.3.3.1.2.5. ResNet .....	20
1.3.4. La classification des images .....	20
1.3.5. La détection des objets .....	21
1.3.5.1. Définition .....	21
1.3.5.2. Les descripteurs de caractéristiques .....	252
1.3.5.2.1. Les caractéristiques d'histogramme des gradients orientés .....	252
1.3.5.2.2. Les caractéristiques de Transformation d'une invariante à l'échelle .....	23
1.3.5.2.3. Les caractéristiques de robuste accéléré .....	23
1.3.5.2.4. Les caractéristiques de pseudo-Haar .....	23
1.3.5.3. Les algorithmes traditionnels de la détection des objets .....	23
1.3.5.3.1.1. La sélection des régions .....	23
1.3.5.3.1.2. L'extraction des caractéristiques .....	24
1.3.5.3.1.3. La classification .....	24
1.3.5.3.2. Les exemples des algorithmes traditionnels .....	27
1.3.5.3.2.1. La méthode de Viola et Jones .....	24
1.3.5.3.2.2. La méthode des séparateurs à vaste marge avec HOG .....	25
1.3.5.4. Les groupes des méthodes de détection des objets basée sur CNN .....	25
1.3.5.4.1. Le détecteur à deux étages .....	25
1.3.5.4.1.1. Réseau convolutif régional plus rapide (Faster R-CNN) .....	25
1.3.5.4.1.2. Les Masques de Réseau convolutif régional (Mask R-CNN) .....	26
1.3.5.4.2. Le détecteur à un étage .....	27
1.3.5.4.2.1. You Only Look Once (YOLO) .....	27
1.3.5.4.2.2. Le Détecteur monocoup (SSD) .....	28
1.3.6. Les bases de données d'images utilisées .....	28
1.3.6.1. The PASCAL Visual Object Commun .....	29
1.3.6.2. ImageNet .....	29
1.3.6.3. Common Object in Context .....	29

1.3.6.4. Google's Open Images .....	29
1.4. Les domaines d'application de la vision par ordinateur et le traitement d'image.....	30
1.4.1. La militaire.....	30
1.4.2. Des soins de santé.....	30
1.4.3. Les Drones .....	30
1.4.4. Les véhicules autonomes .....	31
1.5. Conclusion .....	31
CHAPITRE 2 : Le système avancé d'aide à la conduite .....	32
2.1. Introduction.....	32
2.2. Définitions .....	32
2.3. Les types de capteurs utilisés.....	34
2.3.1. Le capteur de vision.....	34
2.3.2. Le capteur de LIDAR .....	35
2.3.3. Le capteur de RADAR.....	35
2.3.4. Les capteurs ultrasoniques .....	35
2.4. Les ADAS basés sur la vision .....	35
2.4.1. Les types des caméras utilisées dans l'ADAS basé sur la vision.....	36
2.4.1.1. Les caméras monoculaires .....	36
2.4.1.2. Les caméras stéréos .....	37
2.4.1.3. Les caméras thermiques (infrarouges).....	37
2.4.2. Le fonctionnement de l'ADAS basé sur la vision.....	38
2.4.2.1. L'acquisition des images .....	39
2.4.2.2. L'extraction et la compréhension des informations.....	39
2.4.2.3. La décision du système.....	40
2.4.3. Des exemples d'ADAS basés sur la vision.....	40
2.4.3.1. Le système de protection des piétons .....	41
2.4.3.2. Le système assistant de stationnement .....	42
2.4.3.3. La détection et la reconnaissance des panneaux de signalisation.....	43

2.4.3.4. La détection de fatigue.....	44
2.4.3.4. Le système de la régulation de la distance et de la vitesse adaptatif .....	45
2.4.3.5. Le système de la détection des feux de circulation.....	46
2.5. Les avantages des ADAS.....	46
2.5.1. Le potentiel de réduction des accidents .....	47
2.5.2. Coût réduit.....	47
2.5.3. La connexion entre les ADAS et le système télématique.....	47
2.5.4. L'utilisation des technologies FCW et AEB.....	47
2.6. Les défis majeurs des ADAS.....	47
2.6.1. Les changements météorologiques.....	47
2.6.2. La consommation d'énergie.....	47
2.6.3. La sécurité.....	48
2.6.4. Les contraintes géospatiales.....	48
2.7. Conclusion .....	48
Chapitre 3 : La conception de système .....	49
3.1. Introduction.....	49
3.2. État de l'art.....	49
3.2.1. La détection.....	49
3.2.2. L'estimation de la distance.....	51
3.3. La description de l'algorithme SSD .....	52
3.3.1. L'extraction des cartes de convolution .....	52
3.3.2. La localisation et la classification des objets .....	54
3.3.3. Les Boites de délimitation par défaut .....	55
3.3.4. Les boites vérité et l'intersection sur l'union .....	56
3.4. La conception de notre système utilisé pour la détection.....	57
3.4.1. La détection et la reconnaissance des objets de circulation routière .....	58
3.4.1.1. L'architecture de réseau de base.....	58
3.4.1.2. Les couches du localisation et classification de modèle SSD .....	58
3.4.1.3. La phase d'apprentissage par transfert de notre modèle.....	59

□ Rechercher des indices des objets de la circulation routière et créer des vecteurs .....	60
□ Le chargement de modèle SSD préentraîné et la création d'une copie .....	60
□ La sélection des couches de classification.....	60
□ Sous-échantillonner les poids des couches de classification .....	61
3.4.1.4. Suppression de non maximum.....	61
3.4.2. L'estimation de la distance entre la caméra et l'objet détecté .....	62
3.4.3. La phase d'utilisation de modèle SSD300 et l'estimation de la distance.....	62
3.4.3.1. La construction de modèle.....	62
3.4.3.2. Acquisition d'image et le prétraitement .....	62
3.4.3.3. La détection des objets et l'estimation de la distance .....	63
3.4.3.4. L'affichage du système.....	64
3.5. Conclusion .....	65
Chapitre 4 : L'implémentation et les résultats obtenus.....	66
4.1. Introduction.....	66
4.2. Les outils.....	66
4.2.1. Le matériel .....	66
4.2.2. Environnements et outils de développement utilisés.....	66
□ <b>Python</b> .....	66
□ <b>OpenCV</b> .....	67
□ <b>NumPy</b> .....	67
□ <b>TensorFlow</b> .....	67
□ <b>Keras</b> .....	67
□ <b>PyQt5</b> .....	68
□ <b>Sublime Text</b> .....	68
4.3. L'implémentation des composantes de notre système.....	68
4.3.1.1. La création du modèle des poids SSD300(8 classes) .....	68
□ Faire une copie de modèle SSD300 pré-entraîné et la charger.....	68
□ La sélection des couches et sous-échantillonnage des filtres et des biais.....	70

4.3.1.2. La détection des objets de la circulation routière .....	71
4.3.2. Le développement de la partie d'estimation entre la caméra et l'objet détecté.....	74
4.4. Les résultats obtenus et la discussion .....	75
□ La première situation .....	75
□ La deuxième situation.....	76
4.1 Conclusion : .....	77
Conclusion générale.....	78
Bibliographie .....	

## Introduction générale

Le développement technologique de ces dernières années a entraîné un large accès aux systèmes numériques de notre vie quotidienne. Parmi ces composantes majeures, une grande importance a été accordée au traitement de l'image. La représentation et le traitement des images numériques sont actuellement l'objet de plusieurs recherches très actives. Le traitement des images est un domaine très vaste et connaît un développement fulgurant depuis beaucoup de décennies.

L'un des types de l'intelligence artificielle les plus importants et les plus convaincants est la vision par ordinateur. Cette dernière est le domaine de l'informatique qui se concentre sur la reproduction des parties de la complexité du système de vision humain et qui permet aux ordinateurs de reconnaître et de traiter les objets dans les vidéos et les images de la même façon que l'humain le fait. Jusqu'à maintenant, la vision par ordinateur ne fonctionnait que de façon réduite. Grâce aux recherches sur l'intelligence artificielle et aux innovations en matière d'apprentissage profond, le domaine a connu un grand bond vers l'avant dernièrement et a pu dépasser de loin l'humain dans beaucoup de tâches dans le domaine de la détection et de la reconnaissance des objets.

L'industrie automobile a connu un des changements les plus fulgurants de son histoire. Les systèmes avancés d'aide à la conduite et les véhicules autonomes sont l'une des technologies d'invention qui sont à l'origine de ce changement dans les véhicules, les systèmes avancés d'aide à la conduite sont un ensemble de systèmes comportant des capteurs tels que l'image, le Lidar, le radar et des processeurs informatiques pour permettre aux utilisateurs d'accéder à une expérience de conduite sécuritaire, intelligente et confortable. Beaucoup d'accidents sont causés par les erreurs humaines, raison pour laquelle les systèmes avancés d'aide à la conduite aident à automatiser et améliorer les véhicules pour sécuriser et améliorer la conduite. Ils sont conçus pour prévenir le conducteur d'un éventuel danger et l'aider à garder le contrôle du véhicule afin d'éviter les accidents et diminuer sa gravité si possible.

Dans notre projet, nous avons développé un programme informatique basé sur les techniques de vision par ordinateur et de traitement d'image. Le programme a pour but de détecter les objets de la circulation routière (exemple piéton, camion, voiture, etc.). Il permet d'estimer la distance entre la caméra, de localiser l'objet et reconnaître sa catégorie et cet objet. Pour réaliser ce projet, nous ferons appel à un capteur (une caméra) placé sur le tableau de bord

d'une voiture. Les résultats ainsi produits par la caméra seront traités par le programme qui va générer des informations (distance, messages d'alarmes sur la nature de l'obstacle, coordonnées, etc.) qui vont être analysées par le système embarqué sur le véhicule.

Ce projet de mémoire sera déployé sur quatre chapitres. Le chapitre 1 traitera en particulier deux parties. Dans la première, nous traiterons la notion et les techniques principales de traitement d'image et essentiellement la segmentation et l'acquisition. La seconde partie sera consacrée à la vision par ordinateur ainsi que de l'apprentissage profond. Dans le chapitre 2, nous présentons la notion de système d'aide à la conduite et les différents capteurs utilisée dans le système, ensuite nous détaillerons les systèmes d'aide à la conduite fondés sur la vision (capteurs de vision), et nous ferons ressortir les défis majeurs rencontrés dans ce système ainsi que ses avantages.

Dans le chapitre 3, nous mettrons en place une méthode qui va nous permettre de détecter de manière automatique les différents objets (piétons, voiture, etc.), elle se basera sur des techniques telles que la vision par ordinateur et sur une méthode qui permet d'estimer la distance entre l'objet détecté et une caméra, ainsi que sur les architectures d'apprentissage profond. Le chapitre IV, sera dédié au développement du système et des résultats attendus lors de sa mise en application sur des enregistrements vidéo, des images, et des vidéos prises par le tableau de bord des véhicules. Nous mettrons également en évidence les avantages et les faiblesses du système que nous allons développer. Par la suite nous terminerons par une conclusion générale.

## CHAPITRE 1

### Le traitement d'image et la vision par ordinateur

#### 1.1. Introduction

Le traitement d'image est un domaine très vaste qui a connu un essor important depuis quelques années. C'est l'application d'un ensemble de techniques sur des images numériques pour les améliorer ou en tirer des informations. Les images et vidéos peuvent être recueillies sur certaines plateformes telles que (Google, Facebook, etc.) ce qui aide au développement de la vision par ordinateur.

Grâce au traitement d'images, l'ordinateur peut comprendre et analyser le monde qui nous entoure. Des milliards de dollars sont investis par les pays développés pour augmenter les connaissances de cette technologie, car plus le taux d'erreurs est bas et plus les gains sont importants, ce qui donne de meilleurs résultats. Cette technologie pourra être utilisée dans beaucoup de domaines (le domaine de la circulation, de la santé, la surveillance, les véhicules autonomes, etc.).

Nous définirons dans la première partie de ce chapitre l'image numérique et ces différentes caractéristiques, ensuite les techniques principales de traitement d'images.

Dans la seconde partie, nous définirons la vision par ordinateur, l'apprentissage automatique, et l'apprentissage profond. Dans la présentation de l'apprentissage profond, nous décrivons principalement les réseaux de neurone convolutif CNN. Puis, nous définirons la classification des images, et la détection des objets.

Par la suite, nous présentons les bases de données les plus utilisées comme une référence dans le développement et la recherche des techniques de la vision par ordinateur.

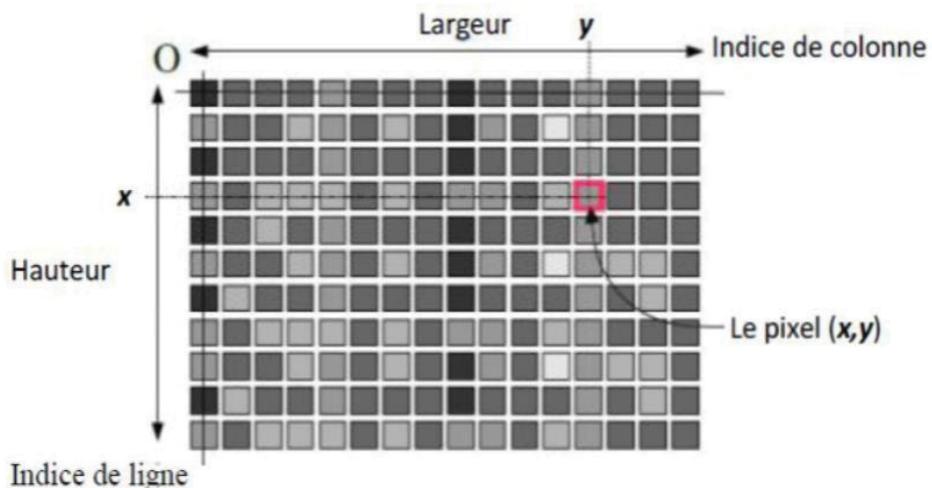
Dans la dernière partie, nous montrons quelques domaines où on utilise la technologie de la vision par ordinateur et les techniques de traitement d'images.

## 1.2. Le traitement d'images

### 1.2.1. Définition de l'image numérique

L'image numérique est une image dont la surface est séparée en éléments de taille fixe nommée cellules ou pixels, ayant chacune des caractéristiques à un niveau de gris ou de couleurs. La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques  $f(x,y)$ , comme la montre la figure I.1. où :  $x$ ,  $y$  sont les coordonnées cartésiennes d'un point de l'image et  $f(x, y)$  est son niveau d'intensité.

La valeur de chaque point exprime la mesure d'intensité lumineuse perçue par le Capteur [1].

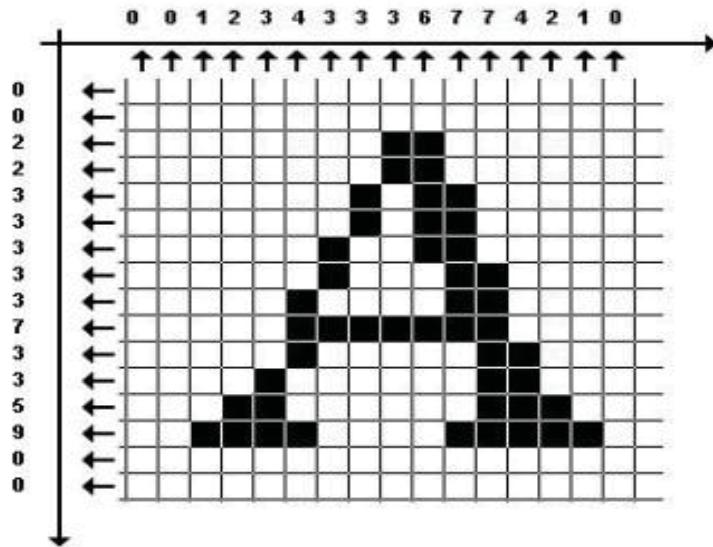


**Figure 1.1.** Représentation d'image numérique [1].

### 1.2.2. Les caractéristiques des images numériques

#### 1.2.2.1. Le pixel

Pour former une image numérique, il faut constituer un ensemble de points appelés pixels. On appelle pixel le plus petit élément constitutif d'une image numérique, et dans chaque pixel contient une couleur. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image [1].



**Figure 1.2.** Le groupe de pixels forme la lettre A [2].

### 1.2.2.2. La résolution

On appelle résolution le nombre de pixels par unité de surface, elle est exprimée le plus souvent en points par pouce (PPP, en anglais DPI pour Dots Per Inch), un pouce représente 2.54 cm [3].

La résolution définit la qualité et la précision d'une image. Plus la résolution est grande (plus il y a de pixels dans une surface de 1 pouce), plus l'image est précise dans les détails et plus la qualité de l'image augmente.

**Remarque [3] :**

$$1 \text{ pouce} = 2,54 \text{ cm.}$$

$$1 \text{ pouce} = 25,40 \text{ mm} = 100 \text{ pixels.}$$

$$1 \text{ inch} = 2,54 \text{ cm} = 1 \text{ pouce.}$$

### 1.2.2.3. La dimension

La dimension est la hauteur et la longueur d'une image numérique, elle se mesure en pixels et peu se présente sous la forme d'une matrice contenant des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de colonnes de cette matrice multiplié par le nombre de lignes nous donne le nombre total de pixels d'une image [2].

#### 1.2.2.4. La profondeur

La profondeur de l'image représente le nombre de bits par pixel, cette valeur reflète le nombre de gris ou de niveaux de couleur d'une image, par exemple [4]:

- 32 bits/pixel = 1,07 milliard de couleurs
- 24 bits = 16,7 millions de couleurs
- 16 bits = 65 536 couleurs □ 8 bits = 256 couleurs

#### 1.2.2.5. Le poids de l'image

On peut déterminer le poids d'une image en fonction de ces deux paramètres : la profondeur et la dimension. Le poids de l'image est calculé en multipliant sa dimension par sa profondeur.

Par exemple, pour une image 640x480 en vraies couleurs (True colors), nous avons les données ci-dessous.

- Le nombre de pixels (dimension) :  $640 \times 480 = 307200$
- Le poids de chaque pixel (profondeur) : 24 bits = 3 octets
- Le poids de l'image est ainsi égal à :  $307200 \times 3 = 921600$  octets [4].

#### 1.2.2.6. La texture

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont, par exemple, un motif basique qui se répète.

La texture est composée de Texel, l'équivalent des pixels [3].

#### 1.2.2.7. Le bruit

Un bruit ou un parasite dans une image est un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Le bruit numérique est une notion générale à tout type d'image numérique, et ce quel que soit le type du capteur à l'origine de son acquisition (appareil photo numérique, scanner, caméra thermique. etc.) [3].

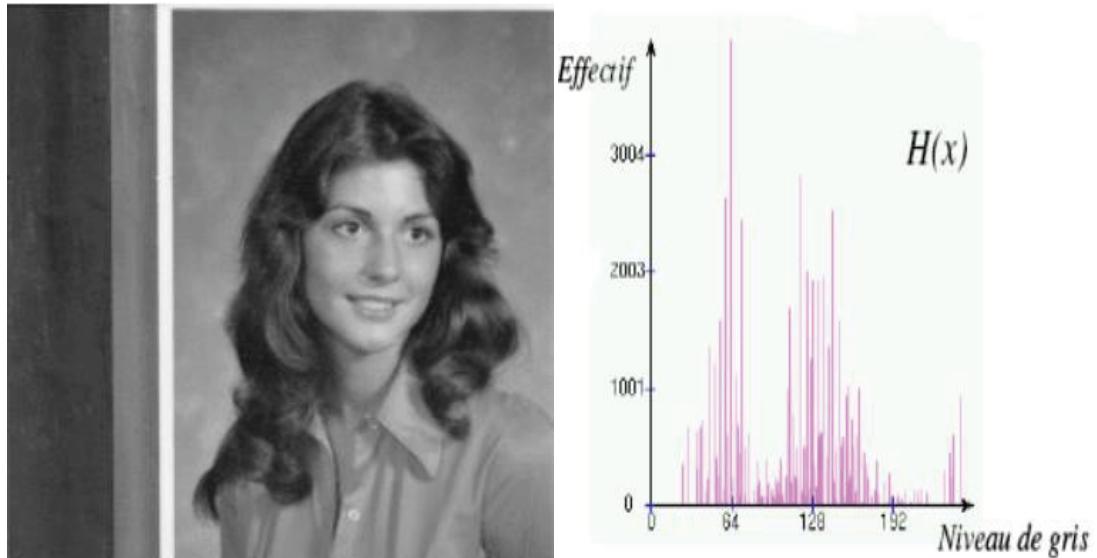
#### 1.2.2.8. La luminance

La luminance représente le degré de luminosité des points de l'image. Elle est définie aussi comme étant un quotient de l'intensité lumineuse dans une surface par l'aire apparente de cette

surface. Pour l'observateur lointain, le substitut du mot luminance est le mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance est caractérisée par des images lumineuses et brillantes avec un bon contraste. Il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir. Ces images entraînent des pertes de détails dans les zones sombres ou lumineuses [3].

#### 1.2.2.8. Histogramme

L'histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite) [2].



**Figure 1.3.** Représentation d'un histogramme d'une image sous Matlab avec  $H(x)$  est le nombre de pixels dont le niveau de gris est égal à  $x$ .

#### 1.2.2.9. Le contraste

Le contraste est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images [1].

### **1.2.3. Les formats standards d'image**

#### **1.2.3.1. L'image matricielle**

Une image matricielle (ou bitmap) est formée d'un tableau de points ou pixels, chaque point contient des informations de position et de couleur.

Plus la densité des points est élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée. Ce type d'image est adapté à l'affichage sur écran mais peu adapté pour l'impression car bien souvent la résolution est faible (couramment de 72 à 150 ppp pour les images sur Internet) [1].

Les formats standards des images matricielles sont BMP (Windows Bitmap), PCX (PiCture eXchange), GIF (Graphic Interchange Format) et JPG ou JPEG (Joint Photographique Expert Group) [1].

#### **1.2.3.2. L'image vectorielle**

Le principe des images vectorielles est de représenter les données de l'image à l'aide de formules mathématiques. Cela permet d'agrandir l'image indéfiniment sans perte de qualité et d'obtenir un faible encombrement. Des formats standards des images vectorielles sont DXF (Data eXchange Format) et CGM (Computer Graphics Metafile) [24].

Par exemple, pour décrire un cercle dans une image, il suffit de noter la position de son centre et la valeur de son rayon plutôt que l'ensemble des points de son contour.

Ce type est généralement obtenu à partir d'une image de synthèse créée par logiciel (par exemple : Autocad) et non pas à partir d'un objet réel. Ce type est donc particulièrement adapté pour le travail de redimensionnement d'images, la cartographie ou l'infographie [3].

### **1.2.4. Les types des images**

#### **1.2.4.1. L'image binaire**

Une image binaire (ou image noir et blanc) est une image  $M \times N$  où chaque point peut prendre uniquement la valeur 0 ou 1. Les pixels sont noirs (0) ou blancs (1). Le niveau de gris est codé sur un bit (Binary digit). Dans ce cas avec  $Ng = 2$  et la relation sur les niveaux de gris devient :  $p(i, j) = 0$  ou  $p(i, j) = 1$  [2].

#### **1.2.4.2. L'image en niveaux de gris**

Une image en niveaux de gris permet un dégradé de gris entre le noir et le blanc. En général, on code le niveau de gris sur un octet (8 bits) soit 256 nuances de dégradé. L'expression de la valeur du niveau de gris avec Niveau de gris = 256 devient :  $p(i, j) \in [0, 255]$  [2].

#### **1.2.4.3. L'image couleur**

Une image couleur est la composition de trois images (ou plus) en niveaux de gris sur trois composantes (ou plus). On définit donc trois plans de niveaux de gris, un rouge, un vert et un bleu. La couleur finale est obtenue par synthèse additive de ces trois composantes (ou plus) [2].

#### **1.2.4.4. L'image à valeur réelle**

Pour certains calculs sur les images, le résultat peut ne pas être entier, il est donc préférable de définir l'image de départ et l'image de résultat comme des images à valeurs réelles. En général, une image à valeur réelle est telle que le niveau de gris est un réel compris entre 0.0 et 1.0.

On a dans ce cas pour une image à niveaux de gris:  $p(i, j) \in [0.0, 1.0]$ . Pour une image couleur, la relation devient  $p_R(i, j) \in [0.0, 1.0]$ ,  $p_V(i, j) \in [0.0, 1.0]$ ,  $p_B(i, j) \in [0.0, 1.0]$  [2].

### **1.2.5. Les principales techniques de traitement des images**

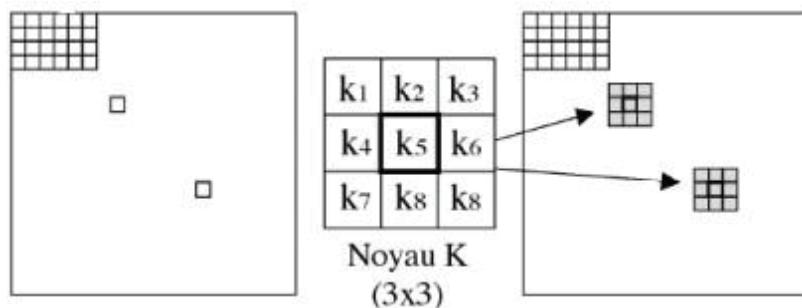
Le traitement d'images est l'ensemble des méthodes et techniques opérant sur l'image, dont le but est d'améliorer son aspect visuel ou d'en extraire des informations jugées pertinentes. Il se définit comme un ensemble de tâches destinées à extraire de l'image des informations qualitatives et quantitatives [4].

#### **1.2.5.1. Acquisition**

Afin de manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation (échantillonnage, quantification). On utilise plus couramment des caméras vidéo, des appareils photo numériques. En médecine, on utilise des imageurs d'écho doppler, échographie, scintigraphie, etc. [4]

### 1.2.5.2. Le filtrage

Le filtrage est une opération qui consiste à réduire le bruit contenu dans une image au moyen d'algorithmes provenant des mathématiques par l'utilisation de méthodes d'interpolation ou de la morphologie mathématique [26]. Il vise à modifier le contenu d'un pixel en prenant en compte une information locale, c'est-à-dire une information extraite du voisinage plus ou moins étendu du pixel. D'une façon générale, le filtrage est obtenu par convolution de l'image avec un noyau défini. Ce noyau peut être interprété comme une petite image ou vignette contenant un gabarit de transformation (linéaire ou non linéaire) et que l'on applique sur chacun des pixels de l'image à filtrer pour créer une nouvelle image [2].



**Figure 1.4.** Filtrage d'une image par un noyau de convolution [11].

#### 1.2.5.2.1. Filtrage linéaire

Ces opérateurs sont caractérisés par leur réponse impulsionnelle  $h(x,y)$  (ou  $h(i,j)$  dans le cas discret, la relation entrée-sortie étant donnée par :

$$S[i,j] = \text{Somme}_{u,v} ( E[i,j] * h[i-u,j-v] )$$

Pour  $u, v$  variant de moins l'infini à plus l'infini.

Ici  $h$  est un support borné. Un filtre linéaire donné sera le plus souvent caractérisé par son kernel, c'est-à-dire la matrice  $[h(i, j)]$  [5].

$$\begin{array}{ccccc}
 h[0,0] & & h[1,0] & h[2,0] & \\
 & & h[0,1] & h[0,2] & \\
 & & h[1,1] & h[1,2] & \\
 & & h[2,1] & h[2,2] & 
 \end{array}$$

Les figures ci-dessous représentent les résultats d'un filtrage linéaire. La figure I.5.a représente l'image originale, la figure I.5.b représente l'application de filtre moyenneur  $7*7$  et la figure I.5.c représente l'application de filtre gaussien  $7*7$ .



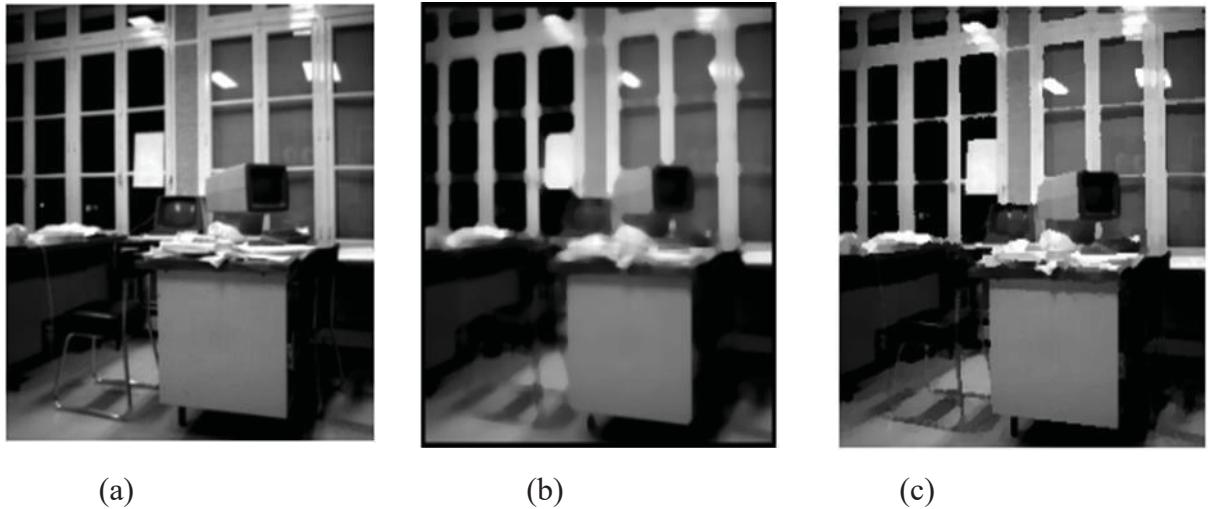
a b c

**Figure 1.5.** L'application des filtrages non linéaire [5].

#### 1.2.5.2.1. Filtrage non linéaire

Le filtrage non linéaire est une opération qui remplace la valeur de chaque pixel par une combinaison non-linéaire des valeurs de ses pixels voisins, ce type de filtre pallie les inconvénients majeurs des filtres linéaires dont la présence des valeurs aberrantes même après filtrage et la mauvaise conservation des transitions [5].

Il existe des méthodes qui appliquent le filtrage non linéaire comme le filtre médian et le filtre de nagao. Dans la figure 1.6.b, on a utilisé un filtre médian de taille  $7*7$  et dans la figure 1.6.c, un filtre nagao de taille  $9*9$  sur une image de niveau de gris.

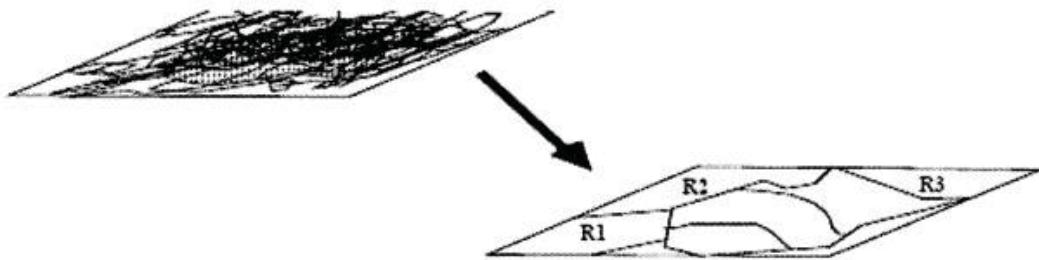


**Figure 1.6.** L'application des filtrages non linéaire [5].

### I.2.5.3. La segmentation d'une image

On utilise la segmentation pour obtenir une partition de l'image en différentes régions d'intérêt. La segmentation est un traitement qui consiste à créer une partition de l'image considérée, en sous-ensembles appelés régions. Une région est un ensemble connexe de pixels ayant des propriétés communes (intensité, texture) qui les différencient des pixels des régions voisines [2].

Il existe plusieurs types de segmentations regroupés en trois catégories : segmentation basée sur les pixels, segmentation basée sur les régions et segmentation basée sur les contours [2].

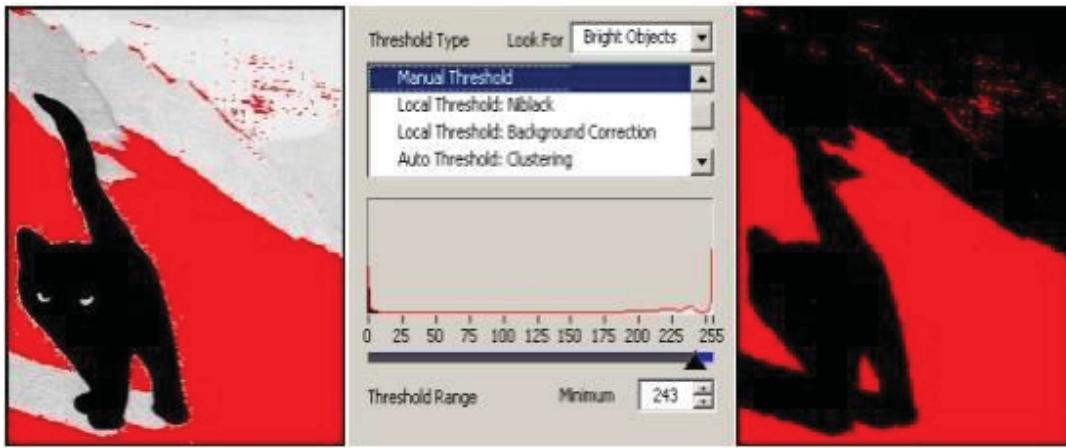


**Figure 1.7.** La segmentation d'une image [6].

#### I.2.5.3.1. Segmentation basée sur les pixels

Le principe est de regrouper les pixels selon leurs attributs sans tenir compte de leur localisation au sein de l'image. Cela permet de construire des classes de pixels. Les pixels adjacents, appartenant à une même classe, forment alors des régions. Il existe des méthodes qui utilisent cette

technique comme les méthodes de seuillage et les méthodes de classification (clustering). La figure 2.6, montre les résultats de la segmentation basée sur les pixels.



**Figure 1.8.** La segmentation basée sur les pixels [2].

#### I.2.5.3.2. Segmentation basée sur les régions

La segmentation basée sur les régions consiste à partitionner l'image traitée en régions homogènes. Chaque objet de l'image pouvant être ainsi constitué d'un ensemble de régions. Dans le but de produire des régions volumineuses et afin d'éviter une division parcellaire des régions, un critère de proximité géographique peut être ajouté au critère d'homogénéité. Au final, chaque pixel de l'image reçoit une étiquette lui indiquant son appartenance à telle ou telle région.

On distingue deux familles d'algorithmes pour l'approche en région.

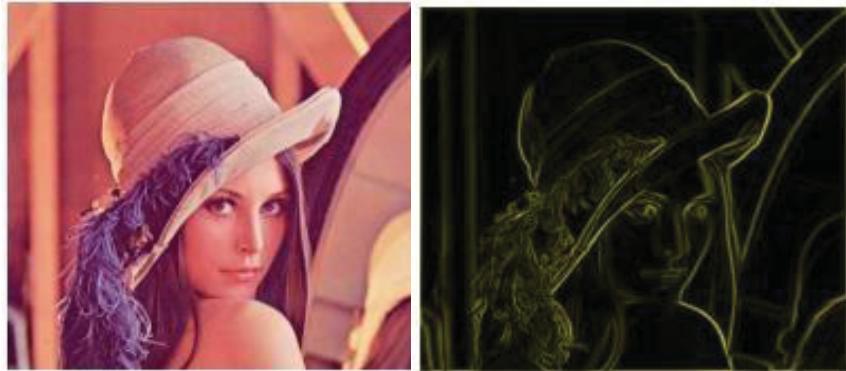
- Les méthodes de croissance de régions qui agrègent les pixels voisins (méthodes ascendantes) selon le critère d'homogénéité (intensité et vecteur d'attributs).
- Les méthodes qui fusionnent ou divisent les régions en fonction du critère choisi (méthodes dites descendantes) [2].

#### I.2.5.3.3. Segmentation basée sur le contour

La segmentation est basée sur le contour de l'objet dans l'image. La plupart des algorithmes qui lui sont associés sont locaux, c'est-à-dire qu'ils fonctionnent au niveau du pixel.

Des filtres détecteurs de contours sont appliqués à l'image et donnent généralement un résultat difficile à exploiter sauf si les images sont très contrastées.

Les contours extraits sont la plupart du temps découpés et peu précis, il faut alors utiliser des techniques de reconstruction de contours par interpolation ou connaître a priori la forme de l'objet recherché [5].



**Figure 1.9.** Détection des contours sur Lena [5].

### 1.3. La vision par ordinateur

#### 1.3.1. Définition

La vision par ordinateur (en anglais « computer vision ») est un domaine de l'informatique qui vise à permettre aux ordinateurs de voir, d'identifier et de traiter les images de la même façon que la vision humaine, puis de fournir des résultats appropriés. C'est la théorie qui sous-tend la capacité des systèmes d'intelligence artificielle à voir et à comprendre leur environnement. Ce domaine interdisciplinaire simule et automatise ces éléments des systèmes de vision humaine en utilisant des capteurs (par exemple la caméra), des ordinateurs et des algorithmes d'apprentissage automatique [7]. L'objectif de la vision par ordinateur est de comprendre ce qui est vu, et extraire des informations complexes sous une forme qui peut être utilisées dans d'autres processus. Les spécialités les plus connues et les plus utilisées dans ce domaine sont la classification des images et la détection des objets.

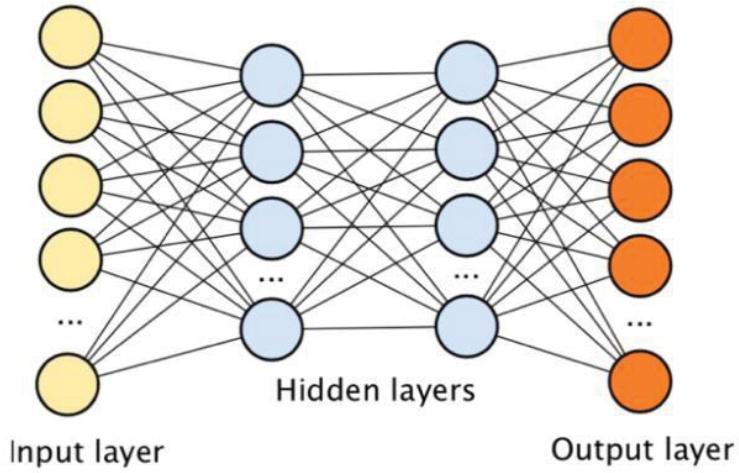
#### 1.3.2. L'apprentissage automatique

L'apprentissage automatique (en anglais Machine Learning) est un champ d'études de l'intelligence artificielle qui se fonde sur des approches statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des

tâches sans être explicitement programmés pour chacune. L'apprentissage automatique comporte généralement deux phases. La première consiste à calculer un modèle à partir de données, appelées l'apprentissage. La seconde phase correspond à la mise en production, le modèle étant déterminé, de nouvelles données qui peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée (prédiction) [8]. Dans le domaine de la classification des images, on utilise les résultats de descripteurs caractéristiques comme une base de données qui permet au modèle de faire l'apprentissage pour classifier les contenus des images (par exemple reconnaître la présence d'un chat dans une image).

### 1.3.3. L'apprentissage profond

L'apprentissage profond (en anglais « Deep Learning ») est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. On pourrait dire qu'il s'agit d'un réseau neuronal complexe contenant de nombreuses couches cachées [9].



**Figure 1.10.** Illustration montre l'architecture de *Deep Learning* [10].

Un CNN prend généralement un tenseur d'ordre 3 comme entrée, par exemple, une image avec  $H$  lignes,  $W$  colonnes, et 3 canaux (canaux de couleur R, G, B). Des tenseurs d'ordre supérieur d'ordre supérieur peuvent toutefois être traitées par les CNN de manière similaire. L'entrée passe ensuite séquentiellement par une série de traitements. Une étape de traitement est généralement une couche, qui peut être une couche de convolution, une couche de mise en

commun, une couche de normalisation, une couche entièrement connectée, une couche de perte, etc.

Pour l'instant, donnons d'abord une description abstraite de la structure du CNN

$$x_1 \rightarrow w_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_{L-1} \rightarrow w_{L-1} \rightarrow x_L \rightarrow w_L \rightarrow z$$

L'équation ci-dessus illustre la façon dont un CNN se déplace couche par couche dans un passage en avant.

L'entrée est  $x_1$ , généralement une image (tenseur d'ordre 3). Elle passe par le traitement dans la première couche, qui est la première boîte. Nous désignons les paramètres impliqués dans le traitement de la première couche collectivement comme un tenseur  $w_1$ .

A la sortie de la première couche est  $x$  qui sert également d'entrée au traitement de la deuxième couche. Ce traitement se poursuit jusqu'à ce que toutes les couches du CNN soient terminées. Une couche supplémentaire est toutefois ajoutée pour la propagation d'erreurs à rebours, une méthode qui apprend de bonnes valeurs de paramètres dans le CNN. Supposons que le problème à résoudre est un problème de classification d'images avec  $C$  classes. Une stratégie couramment utilisée consiste à sortir  $x$  sous la forme d'un vecteur de dimension  $C$ , dont la  $i$ -ième entrée code la prédiction (posteriori) entrée encode la prédiction (probabilité postérieure de  $x_1$  provient de la  $i$ -ème classe).

Pour faire de  $x_L$  une fonction de masse de probabilité, nous pouvons définir le traitement dans la  $(L - 1)$ -ème couche comme une transformation softmax de  $x_{L-1}$ .

Dans d'autres applications, la sortie  $x_L$  peut avoir d'autres formes et interprétations.

La dernière couche est une couche de perte. Supposons que  $t$  soit la valeur cible correspondante (vérité du terrain) correspondante pour l'entrée  $x_1$  alors une fonction de coût ou de perte peut être utilisée pour mesurer l'écart entre la prédiction CNN  $x_L$  et la cible  $t$ .

Par exemple, une fonction de perte simple pourrait être.

$$z = \frac{1}{2} \|t - x^L\|^2,$$

Il existe différents algorithmes de Deep Learning.

- Les réseaux de neurones profonds : Ces réseaux sont similaires aux réseaux RNA mais avec plus de couches cachées.
- Les réseaux de neurones récurrents.
- Les réseaux de neurones convolutifs.

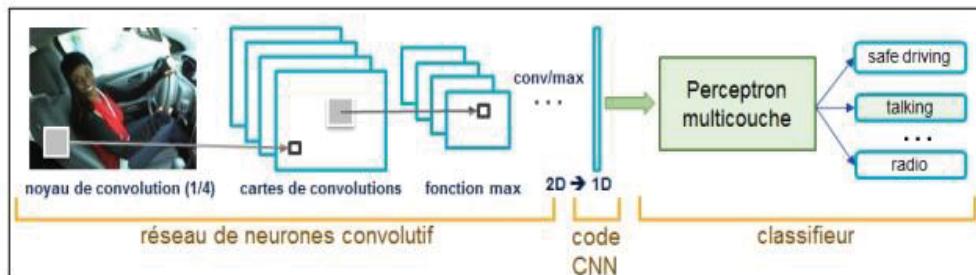
Dans ce mémoire nous nous sommes intéressés à l'étude des réseaux de neurones convolutifs, car cette technique de l'apprentissage profond qui est la plus utilisée dans les domaines de la classification d'images et la détection des objets [8].

### 1.3.3.1. Les réseaux de neurones convolutifs

Les réseaux de neurones à convolutions (en anglais Convolutional Neural Network (CNN)) peuvent être vus comme des réseaux de neurones multicouches particulièrement adaptés au traitement des signaux 2D. Ces réseaux ont été inspirés par les travaux de Hubel et Wiesel sur le cortex visuel chez les mammifères. Il était créé la première fois par Yann Lecun et al en 1989 [11].

Dans une architecture standard d'un CNN, on distingue deux parties [12]. La première est convective. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Figure 1.11).

Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.



**Figure 1.11.** Architecture et composition d'un réseau de neurones convolutifs [7].

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

### 1.3.3.1.1. Les couches d'un Réseau Neurones Convolutives

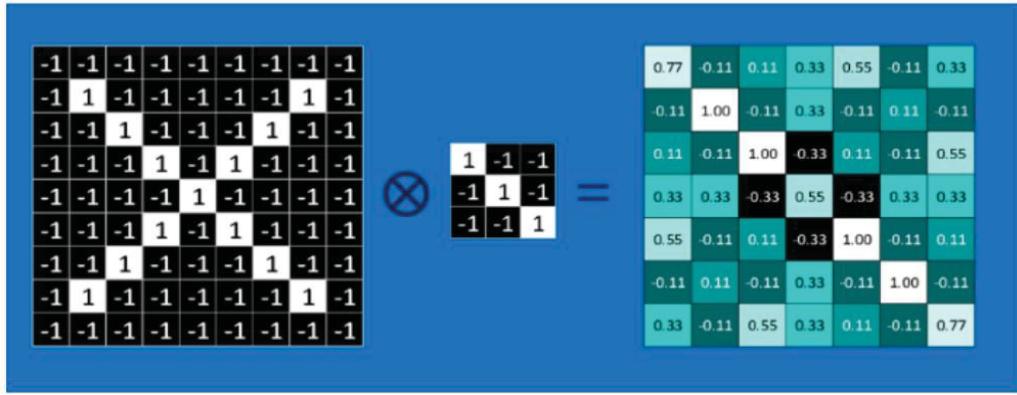
Une architecture CNN est formée par un empilement de couches de traitement indépendantes comme expliqué dans cette sous-section [13] [14] [15]

#### 1.3.3.1.1.1. La couche de convolution (CONV)

La couche de convolution est le bloc de construction de base d'un CNN, elle traite les données d'un champ récepteur. Trois paramètres permettent de dimensionner le volume de la couche de convolution. La profondeur, le pas et la marge.

- Profondeur de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas : contrôle du chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie est grand.
- La marge (à 0) ou « zéro padding » : Cette marge permet de contrôler la dimension.

En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée. L'opération de convolution est illustrée dans la (figure 1.12).



**Figure 1.12.** Illustration de l'opération de convolution entre une image et un filtre [13].

### 1.3.3.1.1.2. La couche de correction (Relu)

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement, une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. Cette fonction force les neurones à retourner des valeurs positives.

Une couche Relue ne modifie pas la taille de l'entrée, c'est-à-dire  $x$  et  $y$  partagent la même taille. En fait, l'Unité Linéaire Rectifiée (d'où le nom ReLU) peut être considérée comme une troncature effectuée individuellement pour chaque élément de l'entrée :

$$\text{a)} \quad y_{i,j,d} = \max\{0, x_{i,j,d}^l\},$$

Avec  $0 \leq i < Hl = Hl+1$ ,  $0 \leq j < Wl = Wl+1$ , et  $0 \leq d < Dl = Dl+1$

Il n'y a pas de paramètre à l'intérieur d'une couche ReLU, donc pas besoin de paramètre d'apprentissage dans cette couche.

$$\text{b)} \quad \frac{dy_{i,j,d}}{dx_{i,j,d}^l} = \llbracket x_{i,j,d}^l > 0 \rrbracket,$$

Cette formule est la fonction indicatrice, étant 1 si son argument est vrai, et 0 sinon.

Par conséquent, nous avons

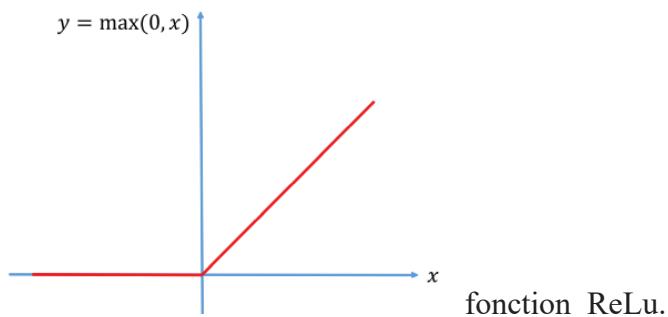
$$\left[ \frac{\partial z}{\partial \mathbf{x}^l} \right]_{i,j,d} = \begin{cases} \left[ \frac{\partial z}{\partial \mathbf{y}} \right]_{i,j,d} & \text{if } \mathbf{x}_{i,j,d}^l > 0 \\ 0 & \text{otherwise} \end{cases} .$$

Notez que  $y$  est un alias pour  $x_{L+1}$

Strictement parlant, la fonction  $\max(0, x)$  n'est pas dérivable en  $x = 0$ , donc l'équation b est un peu problématique en théorie. En pratique, ce n'est pas un problème et ReLU est sûr à utiliser. Le but de ReLU est d'augmenter la non-linéarité du CNN. Depuis l'information sémantique dans une image (par exemple, une personne et un chien Husky assis à côté les uns aux autres sur un banc dans un jardin) est évidemment une cartographie hautement non linéaire des valeurs de pixel dans l'entrée, nous voulons le mappage de l'entrée CNN à sa sortie également être hautement non linéaire. La fonction ReLU, bien que simple, est une fonction non linéaire

Si on traite  $x_{L, i, j, d}$  comme l'un des HL WL DL caractéristiques extraites par les couches CNN à  $L-1$ , qui peut être positif, nul ou négatif. Par exemple,  $x_{L, i, j, d}$  peut être positif si une région à l'intérieur de l'image d'entrée à certains motifs (comme la tête d'un chien ou une tête de chat ou d'autres motifs similaires); et  $x_{L, i, j, d}$  est négatif ou 0 lorsque cette région ne présente pas ces modèles. La couche ReLU définira toutes les valeurs négatives à 0, ce qui signifie que  $y_{L, i, j, d}$  ne seront activés que pour les images possédant ces modèles à cette région particulière. Intuitivement, cette propriété est utile pour reconnaître des motifs et des objets complexes.

Par exemple, ce n'est qu'une preuve faible pour soutenir "l'image d'entrée contient un chat" si une caractéristique est activée et le motif de cette fonction ressemble à une tête de chat. Cependant, si nous trouvons de nombreuses fonctionnalités activées après la couche ReLU dont les motifs cibles correspondent à la tête, au torse, à la fourrure, aux pattes, etc. du chat, nous avons une plus grande confiance (au niveau 1 + 1) pour dire qu'un chat existe probablement dans l'image d'entrée.

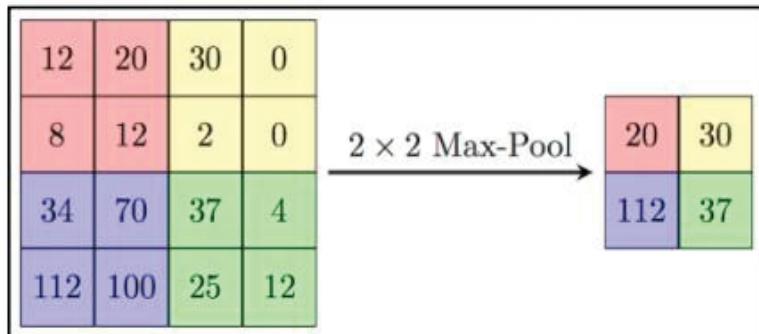


### 1.3.3.1.1.3. La couche de pooling (POOL)

Un autre concept des CNN est le pooling, ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels, chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler le surapprentissage.

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2x2 (largeur/hauteur) et comme valeur de sortie de la valeur maximale en entrée (voir figure 1.13). On parle dans ce cas de « Max-Pool 2x2 ».



**Figure 1.13.** Illustration de l'opération de pooling « POOL 2×2 » [15].

### 1.3.3.1.1.4. La couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente.

### **1.3.3.1.1.5. La couche de perte (LOSS, Softmax)**

La couche de perte spécifie comment l’entraînement du réseau pénalise l’écart entre la sortie prévue (désirée) et réelle (obtenue). Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « Softmax » qui est la plus utilisée permet de calculer la distribution de probabilités sur les classes de sortie.

### **1.3.3.1.2. Les architectures de CNN**

Les architectures de CNN plus populaires sont décrites ci-dessous [9][15].

#### **1.3.3.1.2.1. LeNet**

LeNet était le réseau neuronal convolutif archétype développé par Yann LeCun et al, en 1990, puis amélioré en 1998. L’architecture LeNet la plus efficace et la plus connue est celle qui était habituée à lire les codes postaux, les chiffres, etc.

#### **1.3.3.1.2.2. AlexNet**

La première architecture CNN bien connue est AlexNet, elle popularise le réseau neuronal convolutif pour la vision par ordinateur. Elle a été développée par Alex et al. Plus tard, en 2012, AlexNet a été présenté au défi ILSVRC (en anglais The ImageNet Large Scale Visual Recognition Challenge) et a obtenu de bien meilleurs résultats que le deuxième finaliste. Il a atteint un taux d’erreur de 16 % dans le top 5, alors que le deuxième finaliste avait un taux d’erreur de 26 %.

#### **1.3.3.1.2.3. ZFNet**

Après AlexNet, ce réseau neuronal convolutif de Matthew Zeiler et Rob Fergus a été le gagnant de l’ILSVRC 2013. Il a été baptisé ZFNet (abréviation de Zeiler & Fergus Net). Il a été amélioré sur AlexNet en ajustant les hyperparamètres de l’architecture, principalement en augmentant les dimensions des couches de convolutions centrales et en réduisant la taille des foulées et des filtres sur la couche primaire.

#### **1.3.3.1.2.4. GoogleNet**

Cette architecture a été développée par Szegedy et al, de Google et a été la gagnante du ILSVRC 2014. Elle a atteint un taux d’erreur de 6,67 % dans le top 5, ce qui est très proche de la performance au niveau humain. GoogLeNet disposait de 22 couches ; cependant, avec moins

d'hyperparamètres par rapport à AlexNet (il ne comptait que 4 millions d'hyperparamètres, contre 60 millions pour AlexNet).

#### 1.3.3.1.2.4. VGGNet

Ce réseau finaliste d'ILSVRC 2014 était développé par Karen Simonyan et Andrew Zisserman et désigné sous le nom de VGGNet. Sa principale réalisation a été de montrer que la profondeur du système pouvait être un facteur essentiel de bonne performance. Le dernier réseau le plus performant comportait 16 couches CONV/FC au total (VGG16) et il présentait une architecture uniforme qui n'effectuait que 3x3 convolutions et 2x2 mises en commun du début à la fin.

#### 1.3.3.1.2.5. ResNet

Kaiming He et al, ont développé le réseau résiduel (ResNet). Cette architecture CNN présente des connexions de saut uniques et l'utilisation essentielle de la normalisation par lots. Le principal inconvénient de ce réseau est qu'il est très coûteux à évaluer en raison de la grande diversité des paramètres. Toutefois, jusqu'à présent, ResNet est considéré comme le modèle de réseau neuronal convolutif de pointe et constitue l'option par défaut pour l'utilisation des ConvNets dans la pratique. Il a été le gagnant de l'ILSVRC 2015.

### 1.3.4. La classification des images

La classification des images est une technique utilisée pour classer ou prédire la classe d'un objet spécifique dans une image. Dans cette technique, les entrées sont généralement une image d'un objet spécifique et les sorties sont les classes prédites qui définissent et correspondent aux objets d'entrée.

Un exemple d'un problème de classification est lorsque l'on donne une image d'un chien ou autre chose, nous voulons savoir quel contenu dominant est là. Ainsi, un système de classification devrait toujours classifier cette image comme « chien », peu importe où le chien est dans l'image tant que le chien est le contenu dominant dans l'image. Si le chien n'est plus le contenu dominant, le système devrait changer l'étiquette de l'image au contenu dominant suivant [16].

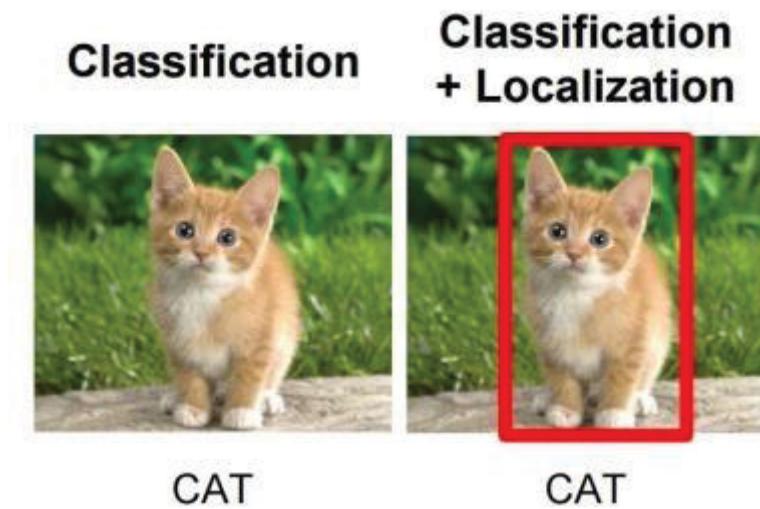
### 1.3.5. La détection des objets

Dans cette partie, nous allons définir ce domaine puis nous allons décrire les descripteurs caractéristiques car ils sont utilisés avec les algorithmes de l'apprentissage automatique pour la détection des objets (les algorithmes traditionnels), enfin nous allons montrer le pipeline des algorithmes traditionnels et les groupes des méthodes de détection des objets basée sur CNN.

#### 1.3.5.1. Définition

La détection d'objets est une tâche de vision par ordinateur qui implique à la fois la localisation d'un ou plusieurs objets dans une image et la classification de chaque objet dans l'image. Pour ce faire, on dessine une boîte de délimitation autour de l'objet identifié avec sa classe prédictive. Cela signifie que le système ne se contente pas de prédire la classe de l'image comme dans les tâches de classification d'images. Il prédit également les coordonnées de la boîte englobante qui correspond à l'objet détecté.

Il s'agit d'une tâche de vision par ordinateur difficile car elle nécessite à la fois une localisation réussie de l'objet afin de localiser et de dessiner une boîte de délimitation autour de chaque objet dans une image, et une classification de l'objet pour prédire la classe correcte de l'objet qui a été localisé [7].



**Figure 1.14** Illustration de la différence entre la classification et la détection [7].

### **1.3.5.2. Les descripteurs de caractéristiques**

Une caractéristique est un élément d'information qui est pertinent pour résoudre la tâche de calcul liée à une certaine application. Ils peuvent être des structures spécifiques dans l'image telle que des points, des bords ou des objets. Un descripteur de caractéristique est un algorithme qui prend une image et produit des descripteurs de caractéristiques/vecteurs de caractéristiques. Les descripteurs de caractéristiques encodent des informations intéressantes en une série de chiffres et agissent comme une sorte « d'empreinte numérique » qui peut être utilisée pour différencier une caractéristique d'une autre. Idéalement, ces informations devraient être invariantes lors de la transformation de l'image, de sorte que nous puissions retrouver l'élément même si l'image est transformée d'une manière ou d'une autre [17], les algorithmes le plus populaires décrits ci-dessous

#### **1.3.5.2.1. Les caractéristiques d'histogramme des gradients orientés**

L'histogramme des gradients orientés (en anglais « the Histogram of Oriented Gradients (HOG feature) ») décrit en 1996, par Robert K. McConnell du Wayland Research inc, s'est généralisé lorsque Navneet Dalal et Bill Triggs utilisent ce descripteur en son projet « la détection de piéton » dans une image statique en 2005. C'est un descripteur de caractéristiques utilisé en vision par ordinateur et en traitement d'images pour la détection d'objets. Cette technique compte les occurrences de l'orientation des gradients dans des parties localisées d'une image [17].

#### **1.3.5.2.2. Caractéristiques de transformation d'une invariante à l'échelle**

La transformation d'une caractéristique invariante à l'échelle (en anglais « ScaleInvariant Feature Transform (SIFT feature) ») a été présentée pour la première fois en 2004, par D. Lowe, de l'Université de Colombie-Britannique. SIFT est l'invariance à l'échelle et à la rotation de l'image, elle est utilisée pour suivre des images, détecter et identifier des objets (qui peuvent aussi être partiellement cachés) [18].

#### **1.3.5.2.3. Caractéristiques de robuste accéléré**

Une fonction robuste accélérée "Caractéristiques robustes accélérées" (caractéristiques SURF)) est un algorithme et un descripteur de fonctionnalité de description, introduits pour la première fois par des chercheurs de l'EPF de Zurich et de l'Université catholique de Louvain. Il est

utilisé dans le domaine de la vision par ordinateur, pour des activités de détection d'objets ou de reconstruction 3D. Il s'inspire en partie du descripteur SIFT, qui dépasse en vitesse et, selon ses auteurs, plus robuste pour différentes transformations d'images [19].

#### **1.3.5.2.4. Caractéristiques de pseudo-Haar**

Les caractéristiques pseudo-Haar (en anglais « Haar-like features ») sont décrites pour la première fois dans un article en 2001 de la revue scientifique (International Journal of Computer Vision (IJCV)) par Paul Viola et Michael Jones, dans lequel ils utilisent une nouvelle méthode de reconnaissance faciale. Avec des images intégrales. Il est possible de calculer très rapidement les caractéristiques pseudo-Haar à l'aide des images intégrales. Une image intégrale est une table de correspondance 2D, construite à partir de l'image d'origine, et de même taille qu'elle. Elle contient en chacun de ses points la somme des pixels situés au-dessus et à gauche du pixel courant [20].

### **1.3.5.3. Les algorithmes traditionnels de la détection des objets**

#### **1.3.5.3.1. Le pipeline des algorithmes traditionnels de la détection d'objet**

##### **1.3.5.3.1.1. La sélection des régions**

Pour trouver des objets dans une image, les méthodes traditionnelles balayent l'image entière en appliquant des fenêtres coulissantes de différentes tailles et échelles et en générant des coupures d'image plus petites qui sont ensuite analysées individuellement pour déterminer s'il y a un objet à l'intérieur de la fenêtre coulissante. En raison du grand nombre de candidats analysés, ce processus est coûteux en temps de calcul [21].

##### **1.3.5.3.1.2. L'extraction des caractéristiques**

Afin d'analyser chaque recadrage d'image généré au cours du processus de fenêtre coulissante, nous avons besoin de fonctionnalités visuelles qui nous fournissent des informations d'image significatives. Par exemple, les fonctions HOG utilisées dans la détection humaine et les fonctions de type Haar utilisées dans la reconnaissance faciale. Cependant, la plupart des descripteurs de caractéristiques sont conçus pour détecter un type d'objet spécifique et leurs performances peuvent être affectées par les conditions d'éclairage [21].

### **1.3.5.3.1.3. La classification**

Une fois que le vecteur de descripteur de caractéristiques pour chaque fenêtre déroulante est obtenu, l'étape suivante consiste à classer les éléments d'image dans une classe d'objet cible et un arrière-plan [21].

### **1.3.5.3.2. Les exemples des algorithmes traditionnels**

#### **1.3.5.3.2.1. La méthode de Viola et Jones**

La méthode de Viola et Jones a été proposée en 2001 par les chercheurs Paul Viola et Michael Jones. Initialement proposée pour la reconnaissance faciale dans une image numérique ou une séquence vidéo et plus tard utilisée pour reconnaître d'autres objets tels que des automobiles dans un processus d'apprentissage (elle utilise les caractéristiques de pseudo-Haar), pour entraîner le classificateur.

AGHaar (un algorithme d'apprentissage automatique). Une fois appris, ce classifieur permet de détecter la présence éventuelle de l'objet dans une image en le recherchant de manière exhaustive dans toutes les positions possibles et dans toutes les tailles possibles [8].

#### **1.3.5.3.2.2. La méthode des séparateurs à vaste marge avec HOG**

Cette méthode est plus largement utilisée dans le domaine de la détection de véhicules et de la détection de piétons [22] [23]. Elle nécessite la préparation de la base d'images en une collection d'échantillons positifs et négatifs, les échantillons positifs étant les objets d'intérêt. Et les échantillons négatifs sont les images qui n'ont pas d'objet d'intérêt. Le HOG est utilisé comme vecteur de caractéristiques pour entraîner le modèle de séparateurs à large marge (SVM). Le descripteur HOG utilise une fenêtre de détection déroulante qui se déplace autour de l'image. À chaque position de la fenêtre de détection, ce descripteur est calculé pour la fenêtre de détection. Le SVM entraîné est ensuite affiché, par exemple en classant l'objet en « personne » ou « non-personne ».

### **1.3.5.4. Les groupes des méthodes de détection des objets basée sur CNN**

#### **1.3.5.4.1. Le détecteur à deux étages**

Les structures de détection à deux étages impliquent la surveillance d'un processus à deux étages. Tout d'abord, l'algorithme se concentre sur la génération d'une région d'intérêt (c'est une zone

suggérée de l'image d'origine) ou de suggestions, puis classe chaque région d'intérêt en classes d'objets prédéfinies [21]. Les exemples suivants appartiennent à ce groupe.

#### 1.3.5.4.1.1. Réseau convolutif régional plus rapide (Faster R-CNN)

L'idée principale d'un réseau de convolution régional plus rapide (en anglais « Faster Region Neural Network ») [24] est de remplacer l'algorithme pour suggérer des régions d'intérêt par un réseau de neurones (en suggérant des régions d'intérêt avec un réseau de convolution de neurones). Il a notamment présenté le Réseau de Propositions Régionales (RPN). Une fois que nous avons nos suggestions régionales, nous les alimentons directement dans un RCNN essentiellement rapide. Nous ajoutons une couche de regroupement, des couches entièrement connectées et enfin une couche de classification softmax et un redresseur de boîte de délimitation (voir figure 1.16). Dans un sens, Faster R-CNN = RPN + Fast R-CNN.

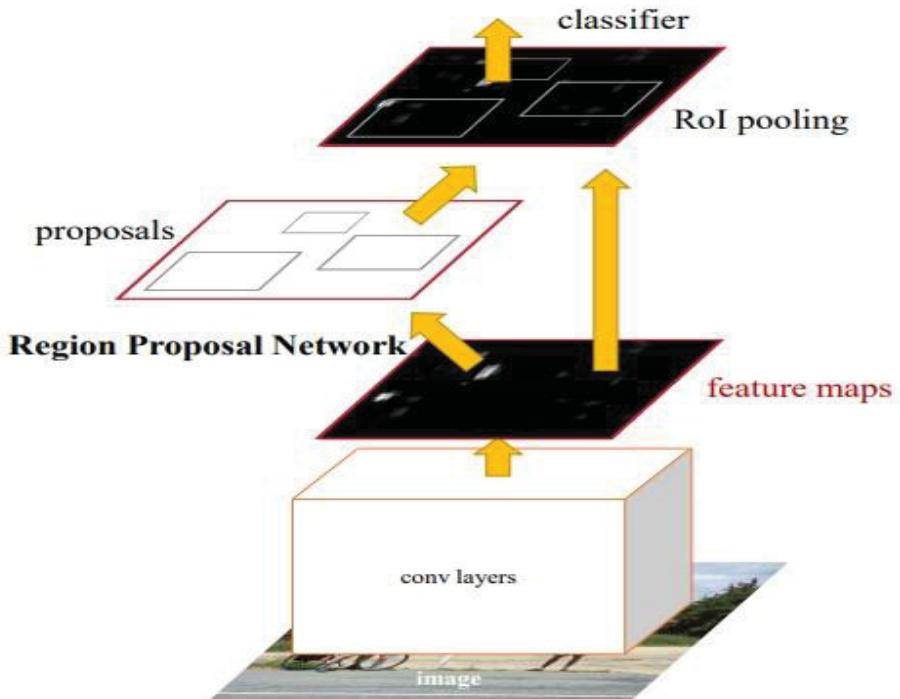
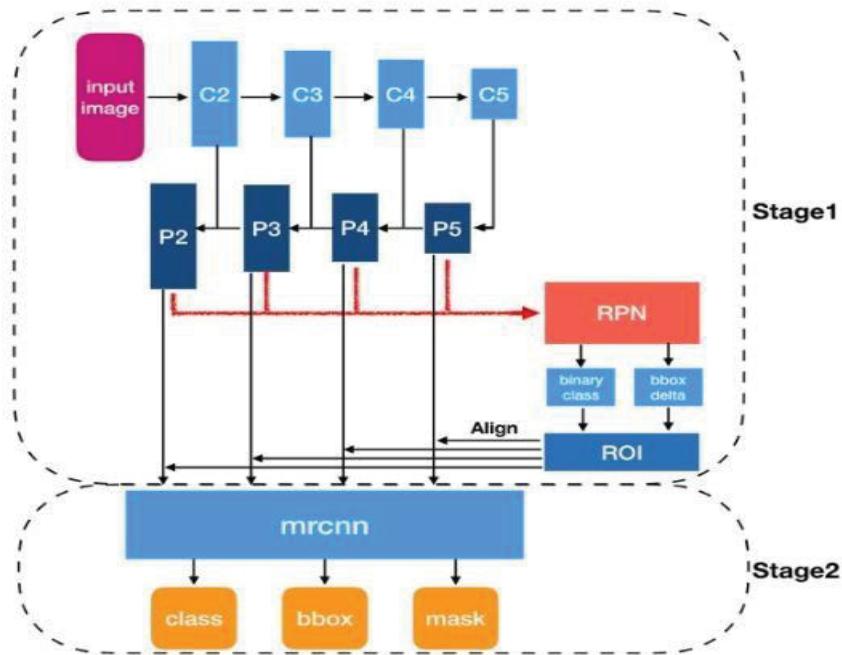


Figure 1.15. Architecture de modèle Faster R-CNN[24].

#### 1.3.5.4.1.2. Les Masques de Réseau convolutif régional (Mask R-CNN)

Le masque RCNN (en anglais « Mask Region Neural Network ») [25] est un détecteur à deux étages. Le premier étage scanne l'image et génère des propositions (zones qui contiennent probablement un objet). La deuxième étape classe les propositions et génère des cadres de délimitation et des masques. Les deux phases sont liées à la structure de base. La structure de base est un réseau neuronal convolutif standard (généralement ResNet50 ou ResNet101) qui agit comme un extracteur de fonction. Les premiers niveaux détectent les éléments de bas niveau (bords et coins) et les niveaux suivants détectent ensuite les éléments de niveau supérieur (voiture, personne, ciel) (voir figure 1.16).



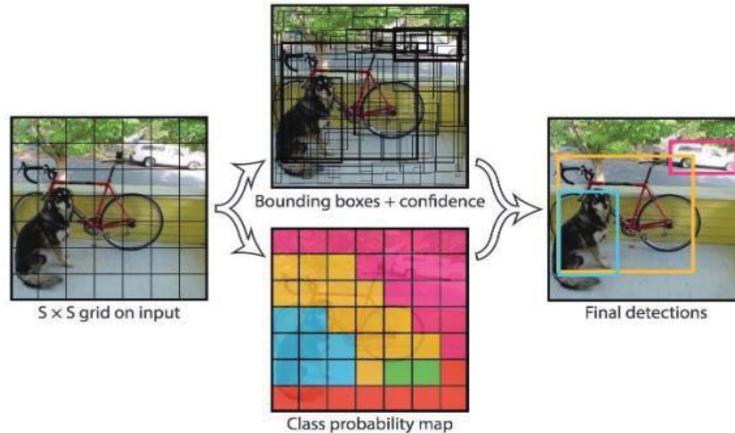
**Figure 1.16.** Architecture de Mask R-CNN [25].

#### 1.3.5.4.2. Le détecteur à un étage

Le détecteur à un étage prédit directement la classe et l'emplacement des objets à l'aide d'un seul réseau de neurones convolutifs (le réseau est capable de trouver tous les objets d'une image en un seul passage à travers le réseau) [21]. L'objectif principal des détecteurs à un étage est d'améliorer la vitesse de détection, mais leur précision est inférieure à celle des détecteurs à deux étages. Les exemples suivants appartiennent à ce groupe.

#### 1.3.5.4.2.1. You Only Look Once (YOLO)

Le modèle YOLO [49] prédit directement les boîtes englobantes et les probabilités de chaque classe avec un seul réseau dans une seule évaluation. La simplicité du modèle YOLO permet des prédictions en temps réel.



**Figure 1. 17.** Modèle YOLO[26].

Initialement, le modèle prend une image en entrée, la divise dans une grille  $S \times S$ . Chaque cellule de cette grille a un nombre  $B$  de zones de limitation avec un score de confiance. Ce score de confiance est simplement la multiplication de la probabilité de détecter une classe d'objets par l'indice IoU (abréviation de « Intersection Junction Overlap ») entre les zones de contraintes prédites (les cadres de délimitation  $B$  utilisés dans chaque cellule) et les images réelles (marquées).

#### 1.3.5.4.2.2. Le DéTECTEUR monocoup (SSD)

Créé en décembre 2016 par Wei Liu et son équipe de recherche, SSD (l'abréviation de « Single Shot Multibox Détection ») [27] est un algorithme de détection d'objets. Le nom de SSD a pour signification que les tâches de localisation et de classification des objets sont effectuées avec en un seul passage dans le réseau, ceci est réalisé grâce à une technique de régression Multi Box. Multi Box est une méthode qui propose des coordonnées de boîte de délimitation indépendantes des classes rapides, avec plusieurs boîtes de délimitation.

#### 1.3.6. Les bases de données d'images utilisées

Les algorithmes d'apprentissage automatique et d'apprentissage profond que nous avons définis nécessitent une base d'images pour trouver le meilleur modèle capable de détecter des objets ou de classer des images avec une grande précision utilisés par des chercheurs d'entreprises bien

connues telles que google, Facebook, Microsoft, etc. , Ils sont créés à partir de bases de données d'images et sont considérés comme des références pour permettre aux chercheurs en intelligence artificielle de mener plus facilement des recherches sans perdre de temps à collecter des images, ci-dessous nous décrivons quelques concepts de base connus dans la recherche en vision par ordinateur.

#### **1.3.6.1. The PASCAL Visual Object Commun**

PASCAL VOC (abréviation de "The PASCAL Visual Object Common") est un concours lancé en 2005 et présentant une base de données d'images avec 20 classes. Lors du dernier concours en 2012, la base de formation et de validation d'images était constituée de 27 50 objets capteurs dans 11 530 images avec 20 classes différentes. Pour la segmentation, la base d'images de validation comprend 6 929 objets, qui sont segmentés en 11 530 images [28].

#### **1.3.6.2. ImageNet**

La base d'images d'ImageNet est représentée par le concours ILSVRC (abréviation de "The ImageNet Large Scale Visual Recognition Challenge") qui a débuté en 2010. À la suite de ce concours, plusieurs modèles ont été sélectionnés pour la détection et la classification d'objets, dont AlexNet. Pour la détection d'objets, ImageNet comprend 65 567 images pour la formation et 20 121 images pour la validation pour 200 classes différentes. Il est organisé hiérarchiquement, selon WordNet, ce qui signifie que chaque noeud de la hiérarchie est représenté par des centaines et des milliers d'images [28].

#### **1.3.6.3. Common Object in Context**

COCO (abréviation de « Common Object in Context ») est une base de données d'images, introduite par Microsoft en 2015, contenant une segmentation d'instances de 80 catégories d'objets dans leur contexte naturel. Les balises COCO incluent également des balises et des points clés ont été ajoutés en 2016. L'ensemble de données COCO comprend 2,5 millions d'instances étiquetées sur 382 000 images [28].

#### **1.3.6.4. Google's Open Images**

Cet ensemble de données contient une collection d'environ 9 millions d'images qui ont été annotées avec des étiquettes au niveau de l'image et des boîtes de délimitation des objets.

Le jeu de formation de la version 4 contient 14,6 millions de boîtes englobantes pour 600 classes d'objets sur 1,74 million d'images, ce qui en fait le plus grand jeu de données existant avec des annotations de localisation d'objets [28].

## **1.4. Les domaines d'application de la vision par ordinateur et le traitement d'images**

La vision par ordinateur et le traitement d'images sont répandus dans le monde entier et ont connu un développement extraordinaire dans divers domaines tels que le militaire, la santé, l'industrie, etc. Dans cette partie, nous avons parlé de certains domaines que nous devrions prendre comme exemples.

### **1.4.1. Le militaire**

Dans le domaine de l'armée moderne, la vision par ordinateur et le traitement d'images sont une technologie de base importante qui aide les systèmes de sécurité à détecter les troupes ennemis ou les saboteurs et à améliorer les capacités de ciblage des systèmes de missiles guidés.

Les concepts militaires tels que la connaissance de la situation s'appuient fortement sur des capteurs d'images pour fournir des informations sur le champ de bataille qui sont utilisées pour prendre des décisions tactiques [29].

### **1.4.2. Les soins de santé**

Le but de la vision par ordinateur dans les soins de santé est de fournir un diagnostic plus rapide et plus précis qu'un médecin ne pourrait faire. Actuellement, les cas d'utilisation les plus courants de la vision par ordinateur et du traitement d'images sont la radiologie et l'imagerie médicale. Les solutions basées sur la vision par ordinateur sont de plus en plus soutenues par les cliniciens en raison du diagnostic de maladies et d'affections à l'aide de divers scanners, tels que les appareils de radiographie et d'IRM [30].

### **1.4.3. Les Drones**

La vision par ordinateur joue un rôle important dans la reconnaissance de différents types d'objets sur les vols aériens. Un puissant système de traitement d'images embarqué et un réseau de neurones de drones sont utilisés pour détecter, classer et suivre les objets en vol. Le réseau neuronal du drone permet de détecter différents types d'objets tels que des véhicules, des contreforts, des

bâtiments, des arbres, des objets sur ou près de la surface de l'eau et sur différents terrains. La vision par ordinateur, utilisée dans les drones, peut également reconnaître des êtres vivants tels que les humains, les baleines, les animaux terrestres et d'autres mammifères marins avec une grande précision [31].

#### **1.4.4. Les véhicules autonomes**

Un domaine qui a captivé l'imagination du public est celui des voitures sans conducteur, qui reposent fortement sur la vision par ordinateur et l'apprentissage en profond. Bien qu'elle ne soit pas encore capable de remplacer complètement le conducteur humain, la technologie des véhicules autonomes a fait des progrès significatifs ces dernières années. L'intelligence artificielle analyse les données obtenues auprès de millions d'automobilistes, apprenant du comportement des conducteurs pour automatiser les marquages routiers, estimer la courbure de la route, détecter les dangers et interpréter les panneaux de signalisation. [32].

### **1.5. Conclusion**

Le but de ce chapitre est de donner un bref aperçu de l'imagerie numérique et des techniques d'imagerie. Nous présentons brièvement la technologie de vision industrielle et le rôle important de l'intelligence artificielle dans cette technologie. La vision par ordinateur avec la disponibilité d'images et de vidéos sur Internet a permis aux industries automobiles de créer des systèmes d'aide à la conduite.

## CHAPITRE 2

### Le système avancé d'aide à la conduite

#### 2.1. Introduction

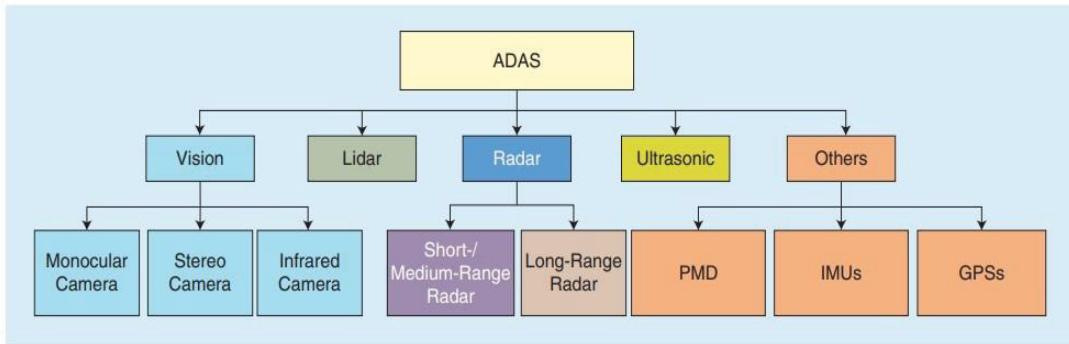
Le nombre d'accidents de la circulation est un problème très préoccupant dans le monde entier. Selon l'Organisation mondiale de la santé, 1,2 million de personnes meurent chaque année et plus de 50 millions de personnes sont blessées dans des accidents de la route [33]. En 2018, l'Algérie a enregistré 3 310 décès et 23 570 blessés dans 2302 accidents de la circulation. Des études récentes de la Direction générale de la sûreté nationale (DGSN) montrent que le facteur humain continue d'être la principale cause de ces accidents (conducteurs et piétons), suivi de l'état des véhicules et des routes et de l'environnement, a précisé le responsable, ajoutant que les excès de vitesse, les dépassements dangereux, l'imprudence des piétons, les manœuvres dangereuses et le non-respect de la signalisation sont les principales causes de ces accidents. La sécurité est devenue un enjeu majeur dans l'industrie automobile au cours des dernières décennies. Des organisations comme "European New Car Assessment" fournissent aux clients des informations sur les systèmes de sécurité de différentes marques et constructeurs automobiles. Ces cotes sont principalement basées sur les systèmes de sécurité passive qui protègent les occupants du véhicule contre les blessures après un accident, tels que les ceintures de sécurité, les airbags et les tableaux de bord rembourrés. Cependant, les constructeurs automobiles ont fait de gros efforts en matière de sécurité active ces dernières années et s'intéressent de plus en plus au développement de systèmes de sécurité active, également appelés systèmes avancés d'aide à la conduite [34].

Dans ce chapitre, nous présentons les systèmes avancés d'aide à la conduite et définissons les technologies utilisées, notamment celles basées sur la vision par ordinateur. Nous soulignerons également les différents avantages de l'utilisation de ce système, ainsi que les différents problèmes que nous pouvons rencontrer lors de sa mise en œuvre.

#### 2.2. Définitions

Les systèmes avancés d'assistance à la conduite ADAS sont des systèmes embarqués intelligents conçus pour être intégrés dans les véhicules et il y a eu de grands progrès au fil des années. Depuis quelques années, ADAS rend la conduite en milieu urbain et périurbain plus facile et plus efficace en exploitant les informations fournies par les capteurs installés dans le véhicule.

pour percevoir l'environnement autour du véhicule. Les ADAS sont classés selon les types de capteurs utilisés, la plupart d'entre eux sont principalement basés sur la vision (caméra), la détection, la portée de la lumière (lidar) et la détection et la portée radio (radar). D'autres technologies telles que PMD (abréviation de Photonics Mixer Device) et GPS (abréviation de Global Positioning System) et IMU (abréviation d'Inertial Measurement Unit) sont également utilisées pour améliorer la fonctionnalité d'autres technologies [35]. La figure 2.1 montre les types de technologies utilisées dans les applications ADAS.



**Figure 2. 1.** Les catégories d'un ADAS [35].

La plupart des applications ADAS exécutent les mêmes processus pour aider les conducteurs. L'ADAS utilise les capteurs mentionnés ci-dessus pour capturer des informations sur l'environnement du véhicule, puis il reconnaît les informations capturées (telles que des images, des signaux, des lumières, etc.). Deuxièmement, ADAS améliore les informations capturées (telles que les images, les signaux, etc.) en utilisant des algorithmes de prétraitement et de traitement [35]. Si le système a récemment compris les informations capturées et détermine que le véhicule ou le conducteur se trouve dans une situation critique (par exemple, freinage d'urgence automatique) ou avertit le conducteur d'éviter un accident, il y a dans les systèmes de sécurité active des dispositifs dans la plupart des systèmes. Le système de sécurité active peut être défini comme un ensemble de véhicules, de personnes et d'éléments environnementaux qui, par leur présence ou leur fonctionnement, peuvent minimiser ou prévenir la gravité de l'accident, contrairement aux systèmes de sécurité passive qui entrent en jeu pendant l'accident [36].

Les ADAS peuvent être regroupés en deux catégories. Ceux pour la sécurité du véhicule et de ses occupants et ceux pour la sécurité des usagers de la route. Dans la première catégorie, on peut citer par exemple l'Adaptive Cruise Control (ACC) qui permet de réguler la vitesse du véhicule pour

maintenir une distance de sécurité raisonnable par rapport au véhicule devant, ou encore Lane Departure Warning (LDW) qui reconnaît la voie et s'active lorsqu'un conducteur est sur le point de sortir d'une voie sans utiliser l'indicateur de direction. Dans la deuxième catégorie, on trouve par exemple des systèmes de détection de piétons qui permettent de reconnaître les piétons et d'avertir le conducteur afin qu'il puisse prendre une décision pour éviter la collision avant que le système n'intervienne et utilise le freinage d'urgence. Une étude d'ABI Research publiée en 2015 montre que les constructeurs automobiles Mercedes-Benz, Volvo et BMW dominent le marché des systèmes anticollision pour la protection des piétons [36]. On trouve également des systèmes de détection de somnolence qui alertent le conducteur dès que des signes caractéristiques de fatigue ou d'inattention sont détectés [8]. Quels que soient les systèmes proposés, les constructeurs automobiles cherchent à proposer des ADAS qui contribuent à renforcer la sécurité routière en aidant les conducteurs.

## 2.3. Les types de capteurs utilisés

Les ADAS sont classés en fonction du type de capteur utilisé. Les capteurs les plus utilisés sont les capteurs de vision, LIDAR, RADAR et à ultrasons. Nous décrivons ces capteurs ci-dessous, mais les autres capteurs sont utilisés pour améliorer les fonctions du système, comme le GPS qui est utilisé pour connaître la position du véhicule dans l'environnement, l'IMU est utilisé pour leur fournir une localisation fiable et un discernement de mouvement. Les applications de navigation et de stabilisation ADAS et PMD sont utilisés pour effectuer une détection optique rapide et une démodulation simultanée de signaux lumineux incohérents [35].

### 2.3.1. Le capteur de vision

La caméra est le capteur de vision le plus utilisé dans l'ADAS basé sur la vision, elle capture des images qui fournissent des informations visuelles au système pour comprendre et analyser la situation du véhicule dans un environnement (comme connaître la présence d'objets autour du véhicule), également il peut être utilisé pour surveiller l'intérieur du véhicule. Les avantages de l'utilisation de la caméra sont qu'elle est facile à intégrer dans le véhicule, qu'elle est peu coûteuse et, surtout, qu'elle peut fournir de nombreuses informations précises sur le système par rapport au système d'autres capteurs [35].

### **2.3.2. Le capteur de LIDAR**

La télédétection par laser (en anglais Light Détection And Ranging (LIDAR)) a une technique très simple pour connaître l'environnement du véhicule, c'est que le lidar tire un faisceau laser puis mesure le temps qu'il faut au faisceau laser pour faire rebondir ce capteur, ces systèmes peuvent produire des images 3D haute résolution et générer une image 3D à 360 ° de l'environnement avec des informations de profondeur précises. Les avantages du Lidar sont qu'il est utile pour détecter des objets et, en plus, les véhicules qui ont LIDAR peuvent voir une portée allant jusqu'à 60 mètres. Mais ces types de capteurs sont lourds, encombrants et très coûteux, et ont un autre inconvénient que les conditions météorologiques (par exemple la pluie, le brouillard etc.) peuvent affecter la précision du système [35].

### **2.3.3. Le capteur de RADAR**

Le capteur de détection et de portée radio (RADAR) fonctionne bien dans les systèmes d'estimation de distance de sécurité car il fournit des données précises. Le rôle du radar est l'émission de micro-ondes, ainsi le système mesure le décalage de la fréquence d'onde (par exemple acoustique ou électromagnétique) observé entre le RADAR (l'émission) et l'objet (la réception), ce décalage représente la distance entre l'émetteur et le récepteur et varie dans le temps. Comparé au LIDAR, le RADAR peut détecter des objets à longue distance et n'est pas influencé par dans conditions météorologiques (pluie, brouillard, etc.) et il peu coûteux. Selon la façon dont ils fonctionnent pour calculer la distance, les radars peuvent être classés en courte portée (0,2 - 30 mètres), moyenne portée (30 - 80 mètres) ou longue portée (80 200 mètres) [35].

### **2.3.4. Les capteurs ultrasoniques**

Ces capteurs utilisent des ondes sonores pour mesurer la distance à un objet. Ces capteurs sont principalement utilisés pour détecter des objets très proches du véhicule [35].

## **2.4. Les ADAS basés sur la vision**

Les constructeurs automobiles cherchent à développer des ADAS plus sûrs basés sur la vision par ordinateur (caméra). En 2015, Tesla a développé l'Autopilot System, une application ADAS basée sur la vision qui peut détecter les voies sur la route et offre un régulateur de vitesse

adaptatif, un stationnement automatique et d'autres fonctionnalités. Le conducteur est responsable de toutes ces fonctions et la voiture doit être constamment surveillée [37].

Dans cette section, nous avons décrit les types de caméras utilisées dans les ADAS basées sur la vision, puis nous avons défini les opérations ADAS basées sur la vision et l'utilisation de la vision par ordinateur dans cette opération, et enfin nous avons quelques exemples de ce système.

#### **2.4.1. Les types des caméras utilisées dans l'ADAS basé sur la vision**

Il existe 3 types de caméras qui sont les plus utilisées dans les ADAS basés sur la vision. Les caméras monoculaires, les caméras stéréo et les caméras thermiques. Nous décrirons ces types ci-dessous.

##### **2.4.1.1. Les caméras monoculaires**

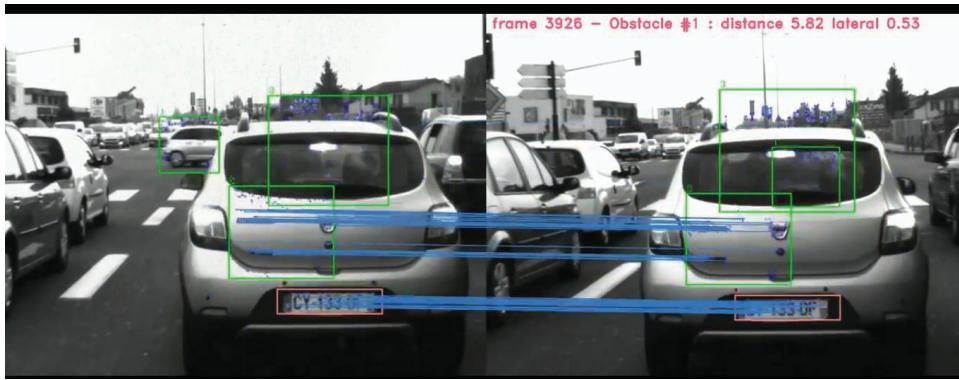
Le système de caméra monoculaire a un seul objectif, ce qui signifie qu'il ne fournit qu'une image à la fois, et les besoins en traitement d'images sont faibles par rapport aux autres types de caméras. Ce type de caméras peut être utilisé dans diverses applications telles que la détection d'obstacles, la détection de piétons (voir figure 2.2), la signalisation routière est également utilisée pour surveiller le conducteur à l'intérieur du véhicule en tant que système de détection de fatigue. Les images des caméras monoculaires manquent d'informations sur la profondeur et ne sont donc pas des capteurs fiables pour l'estimation de la distance.



**Figure 2. 2.** Image RVB d'une caméra monoculaire) [38].

#### 2.4.1.2. Les caméras stéréos

Ce type d'appareil photo se compose de deux ou plusieurs objectifs, l'écran ressemble à une matrice d'images. Les caméras stéréo sont très utiles pour extraire des informations tridimensionnelles à partir de deux images bidimensionnelles ou plus en faisant correspondre des paires stéréo et en utilisant une carte de disparité pour estimer la profondeur relative d'une scène [35]. Les caméras stéréo sont utilisées dans une variété d'applications, telles que la reconnaissance des panneaux de signalisation, des voies, des piétons et l'estimation de la distance, avec beaucoup plus de précision que les caméras monoculaires. La plupart des véhicules utilisant l'ADAS sont équipés de caméras stéréo, les caméras sont situées à l'intérieur du véhicule, derrière le rétroviseur, légèrement incliné vers le bas et face à la route.



**Figure 2. 3.** Une image de caméra stéréoscopique [35].

#### 2.4.1.3. Les caméras thermiques (infrarouges)

Il existe deux types de caméras thermiques, une caméra thermique active avec une source de lumière proche infrarouge (avec une longueur d'onde de 750 nm à 1 00 nm) qui peut être directement intégrée dans les véhicules pour éclairer la scène (invisible à l'œil humain) et un appareil photo numérique standard pour capturer la lumière réfléchie. La figure 2.5 montre la différence entre les images de la caméra normale et de la caméra thermique nocturne [39]. Une caméra thermique passive utilise des capteurs infrarouges, où chaque pixel du capteur infrarouge peut être considéré comme un capteur de température capable de détecter le rayonnement thermique émis par chaque matériau, ne nécessitant pas d'éclairage de scène spécial et il est utilisé plus fréquemment dans les

applications ADAS par rapport aux caméras infrarouges actives (voir Figure 2.4) [40]. Les caméras thermiques restent une solution de vision nocturne permettant au conducteur de voir plus clairement dans des conditions de faible luminosité.



**Figure 2. 4.** Une capture d'une image à partir d'une vidéo de caméra thermique passive.



**Figure 2. 5.** La différence entre caméra thermique, caméra active et caméra monoculaire [40].

#### 2.4.2. Le fonctionnement de l'ADAS basé sur la vision :

Pour que le système puisse prendre une décision concernant l'environnement d'un véhicule, il a besoin d'informations visuelles vues à travers les caméras. Par conséquent, il capture dans un premier temps une image (ou vidéo), puis améliore la qualité de l'image, puis utilise des algorithmes qui lui permettent de reconnaître l'environnement autour du véhicule (ex: détection de voie, détection de véhicule, etc.) et le cas échéant, prennent les décisions ou actions nécessaires (avertissement, freinage d'urgence, signal sonore, etc.).

#### 2.4.2.1. L'acquisition des images

C'est le processus consistant à prendre des photos à partir d'une vidéo. Chaque image est représentée par une matrice de données (pixels) et chaque pixel est souvent représenté par trois couleurs (rouge, vert, bleu (RVB)). Chaque image contient trois canaux d'informations (voir chapitre 1). La fréquence d'images dans l'ADAS visuel varie entre 5 images par seconde et 60 images par seconde (tout dépend de l'application), par exemple les systèmes d'estimation de distance (voir Figure 2.2) et la vitesse nécessitent une fréquence d'images plus élevée en raison des changements rapides de parcours des véhicules dans la rue [35].

#### 2.4.2.2. L'extraction et la compréhension des informations

Dans un premier temps, l'image obtenue après acquisition peut contenir du bruit et ce bruit peut affecter l'efficacité des algorithmes. Par conséquent, le pré-traitement vise à améliorer la qualité de l'image pour extraire des informations de l'environnement du véhicule. Il existe plusieurs techniques de prétraitement d'images, telles que les méthodes d'amélioration du contraste. L'amélioration du contraste se fait en changeant les valeurs initiales pour utiliser toutes les valeurs possibles, augmentant le contraste entre les lentilles et l'environnement environnant. Par exemple, il existe des méthodes de conversion d'espace couleur, cela consiste à convertir l'espace couleur (espace RVB, HSV, CMY, etc) d'une image à l'autre. Son principal avantage est qu'il permet de séparer les effets négatifs (une ombre, un éclairage irrégulier, etc) [35]. Il existe des systèmes qui utilisent des techniques de traitement d'images et d'autres systèmes utilisent la technologie de vision par ordinateur pour extraire des informations et les comprendre, il existe deux catégories de ces systèmes.

- Les systèmes utilisent des descripteurs de caractéristiques (tels que HOG, Haar like, SIFT, etc.) pour extraire des informations (caractéristiques) des objets présents dans l'image capturée à partir d'une vidéo (c'est-à-dire localiser l'objet dans l'image). Ils utilisent ensuite des algorithmes d'apprentissage automatique pour classer des objets localisés (c'est-à-dire

reconnaître la classe d'objets tels que les piétons, les camions, les panneaux de signalisation, etc.).

- Les systèmes utilisent des algorithmes (tels que YOLO, MaskRCNN, Faster RCNN, etc.) basés sur des architectures CNN (telles que AlexNet, ResNet, VGG, etc.). Ces algorithmes utilisent les architectures CNN pour extraire les propriétés de l'objet, pour localiser l'objet dans l'image et aussi pour classer les objets localisés. Les deux catégories de systèmes détectent des objets autour du véhicule, ils peuvent également surveiller des objets ou des conducteurs de véhicules pour détecter ces actions (comme le passage de véhicules, le maintien d'une distance et d'une vitesse de sécurité, etc.). La différence entre les deux catégories est qu'avec de nombreux objets autour du véhicule, la première catégorie de systèmes prend beaucoup de temps pour détecter et surveiller tous les objets, tandis que les autres systèmes détectent et surveillent tous les objets en temps réel.

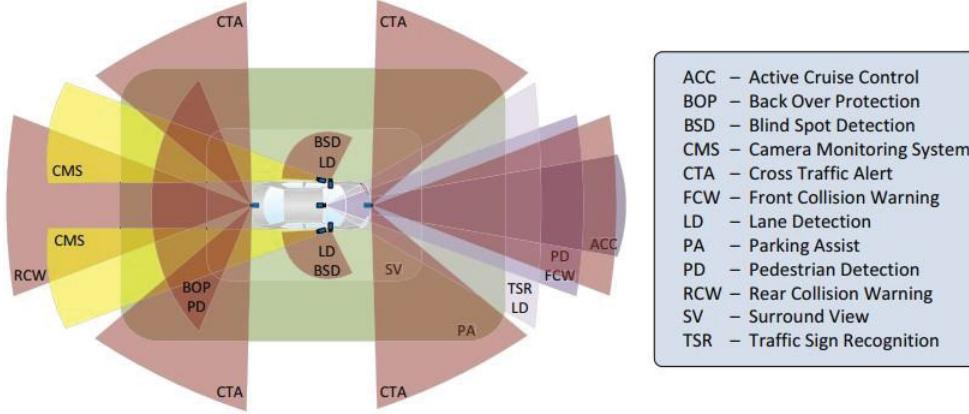
#### **2.4.2.3. La décision du système**

Dans la phase précédente, le système comprend et analyse les images de la caméra représentant l'environnement du véhicule. Ensuite, dans cette phase, le système s'occupe d'interpréter les résultats des phases précédentes afin de prendre des décisions. Dans cette phase, la plupart des ADAS basés sur la vision utilisent la technologie d'avertissement de collision avant et de freinage d'urgence autonome (AEB) pour aider à la prise de décision ou au départ. La technologie d'avertissement de collision avant (FCW) est un dispositif de sécurité active qui avertit le conducteur en envoyant un signal acoustique et/ou tactile ou visuel pour avertir le conducteur d'une situation de collision potentielle. La technologie AEB est un système de sécurité active intelligent utilisé dans les ADAS basés sur la vision pour aider les conducteurs à éviter une collision de véhicules avec les usagers de la route (autres véhicules, piétons, etc.).

Le principal défi de cette phase est que le système peut produire une sortie incorrecte avec une certaine certitude, c'est-à-dire que les résultats des phases précédentes sont corrects, mais la sortie du système est incorrecte.

#### **2.4.3. Des exemples d'ADAS basés sur la vision**

Il existe des ADAS qui utilisent une seule caméra ou une caméra couplée à des capteurs RADAR, LIDAR, etc. (voir la figure 2.6). La caméra peut être installée à l'intérieur ou à l'extérieur du véhicule. Voici quelques ADAS qui utilisent un type de caméra spécifique (nous avons montré les types de caméras utilisées dans la partie précédente).



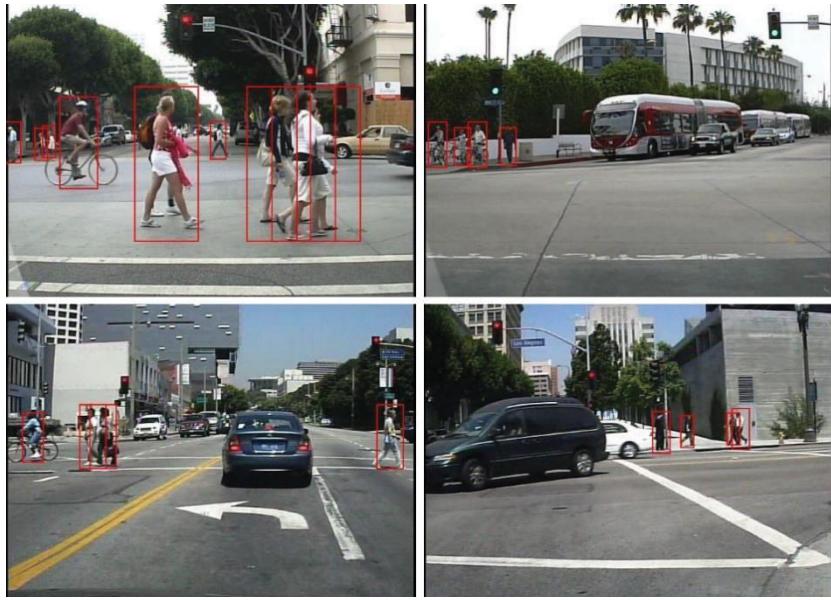
**Figure 2. 6.** Représentation des applications d'ADAS [62].

#### 2.4.3.1. Le système de protection des piétons

Les accidents de piétons représentent une grande partie des accidents mortels dans le monde, c'est pourquoi les industries et les chercheurs tentent de proposer des ADAS basés sur la vision par ordinateur pour réduire le nombre d'accidents, par exemple, le système d'avertissement d'assistance Mercedes-Benz et le système d'évitement des piétons de Toyota Système. [36]. Le principe du système de protection des piétons est d'utiliser des séquences d'images enregistrées par la caméra lorsqu'un véhicule est installé pour détecter les piétons puis avertir le conducteur pour éviter les collisions. La figure 2.7 montre le système de détection des piétons avec une caméra. Il utilise des techniques de vision par ordinateur pour détecter les piétons en tant que descripteurs, par exemple les descripteur HOG, Haarlike et Sift et la segmentation par exemple algorithme AGHaar ou algorithme SVM. Pour classer les piétons et les non-piétons et détecter leurs mouvements. Les algorithmes les plus utilisés pour le suivi des piétons sont Kalman, Meanshift ou Alphabeta Tracker [1].

Dans des travaux plus récents, les chercheurs ont mis en œuvre l'architecture CNN (comme AlexNet et ResNet) pour créer ce système, car cette architecture permet la détection et la

classification des piétons en temps réel [1]. Mais la difficulté est de distinguer les humains et les objets voisins. C'est l'un des principaux problèmes affectant les résultats de ce système.



**Figure 2.7.** Le système de détection des piétons par une caméra monoculaire [48].

#### 2.4.3.2. Le système assistant de stationnement

Le système d'aide au stationnement est un ADAS basé sur la vision qui aide le conducteur à décider de stationner son véhicule. Le système consiste à analyser la taille des positions de station disponibles, ceci peut être réalisé en traitant les données d'images capturées par l'avant, l'arrière et les côtés. Il est également possible de capturer des images autour du véhicule à l'aide d'une caméra grand-angle ou des quatre caméras "fisheye". C.Wang et al [43] ont introduit une méthode de stationnement automatique basée sur un système de vision à 360 degrés, couvert par quatre caméras fisheye embarquées et préinstallées autour d'un véhicule. Le système analyse les stationnements à l'aide d'un algorithme basé sur la transformation du radon (une méthode de traitement d'images est utilisée pour détecter les objets). De plus, le système planifie l'itinéraire de stationnement à l'aide d'une planification d'itinéraire en double boucle (une méthode de vision par ordinateur pour guider le conducteur à garer son véhicule sur le parking). K. Eyal et al [44] ont proposé un système d'aide au stationnement, ce système se compose de trois caméras monoculaires standards connectées à un ordinateur, qui sont fixées au véhicule. Dans un premier temps, recherchez un stationnement gratuit puis informez le conducteur, il utilise l'algorithme de transformation de Hough pour identifier les limites de stationnement disponibles, et utilise la méthode de filtrage de descripteur local pour filtrer la zone d'ombre présente à cet endroit lorsque l'emplacement est sélectionné par le conducteur. La

position relative entre le véhicule et le parking est supervisée. Il utilise la méthode Lucas Kanade d'algorithme de flux optique pour surveiller les objets autour du véhicule. Des instructions de stationnement visuelles et vocales sont présentées au conducteur. Le système avertit le conducteur des objets surveillés par l'algorithme de flux optique. La figure 2.8 montre l'écran de guidage au stationnement sur l'ordinateur, l'image en haut à gauche est la vidéo de la caméra avant. L'image en haut à droite est la caméra vidéo sur la droite. L'image ci-dessous à gauche montre le mouvement de la voiture détecté par le système. L'image ci-dessous à droite montre les instructions que le système donne au contrôleur).



**Figure 2. 8.** Affichage de guidage de stationnement sur ordinateur [64].

#### 2.4.3.3. La détection et la reconnaissance des panneaux de signalisation

La détection et la reconnaissance des panneaux de signalisation sont parmi les composants les plus importants de l'ADAS basé sur la vision, car les panneaux régulent la circulation, indiquent les conditions routières et sont omniprésents. Les panneaux se caractérisent par leurs formes (triangle, cercle ou octogonal), leurs couleurs (rouge, blanc) et aussi par leur contenu. La figure 1.8 montre un exemple de système de détection et de reconnaissance de panneaux de signalisation. Les chercheurs ont proposé des approches pour développer ce système, y compris les approches de Ben Romdhane et al [21] qui utilisent l'espace colorimétrique HueSaturationValue (TSV) pour générer des régions candidates. Les fonctions HOG (voir chapitre 1) sont utilisées comme descripteurs de fonction et la méthode de la voiture porteuse multimédia est utilisée pour identifier la catégorie des panneaux de signalisation.

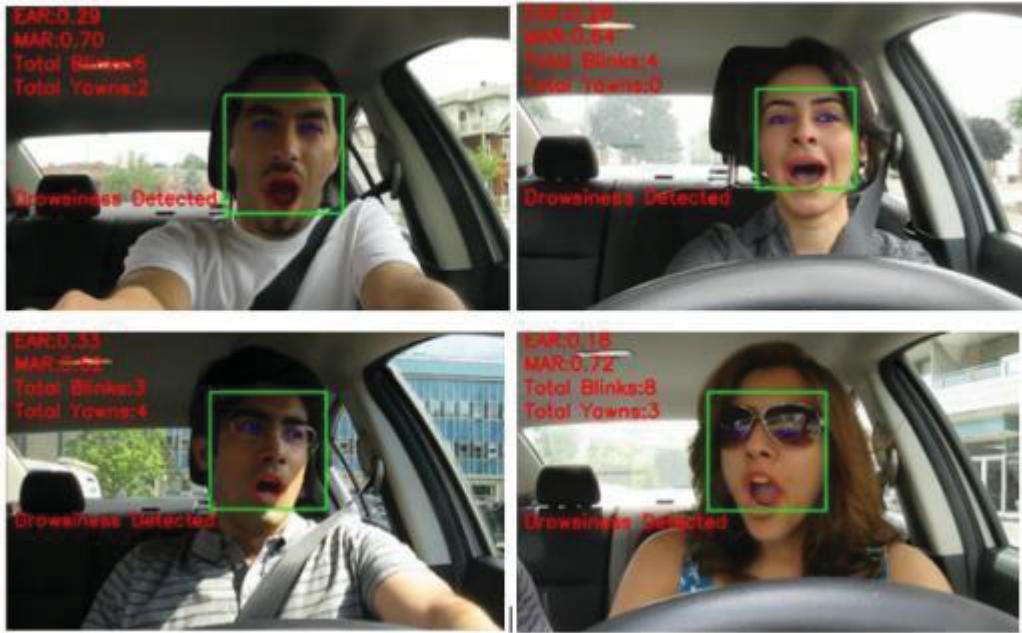
Zhang et al [21] utilisent une version modifiée de YOLO (voir chapitre 1) pour détecter les panneaux de signalisation. Les chercheurs ont ajusté la taille des filtres et le nombre de couches pour trouver le meilleur équilibre entre vitesse et précision.



**Figure 2.9.** Système de détection et de reconnaissance des panneaux de signalisation [21].

#### 2.4.3.4. La détection de fatigue

La somnolence au volant est une cause majeure d'accidents de la route et expose les conducteurs à un risque de collision très grave. Les méthodes de détection de la fatigue du conducteur sont basées sur la description de certaines expressions faciales. Il existe des systèmes qui utilisent des mesures appliquées à ces caractéristiques prélevées sur les yeux et la bouche. De plus, l'utilisation d'algorithmes d'apprentissage (méthode viola and Jonas, algorithme Multi TaskCNN, etc.) permet de classer les couplages de mesures dans deux états possibles : fatigué ou normal. Ces valeurs permettent de distinguer les signes de fatigue (bâillement, yeux fatigués, etc.) des autres expressions faciales (sourire, parler, etc.) [8]. Si le conducteur est fatigué, le système l'alerte pour prendre une décision (par exemple, arrêter le véhicule). La figure ci-dessous montre le résultat du système de détection de fatigue.



**Figure 2. 10.** Alerte de somnolence lorsque le seuil passe la 4e fois Détections [8].

#### 2.4.3.4. Le système de la régulation de la distance et de la vitesse adaptatif

Le système d'estimation de la distance et de la vitesse de sécurité est l'un des composants les plus importants de l'ADAS basé sur la vision, car les collisions arrière constituent la majorité de tous les accidents (29,5 % aux États-Unis et 29 % en Allemagne). L'inattention (sommeil, utilisation d'appareils de communication, etc.) représente 91 % des accidents liés au conducteur. Si le conducteur remarque plus tôt le danger, les collisions arrière des années 1960 et les collisions des années 1990 peuvent être évitées si elles sont remarquées plus tôt, après une seconde [38]. Ce système consiste à détecter et surveiller les véhicules, à estimer la distance à chaque véhicule et à estimer la vitesse du véhicule, après quoi il fournit au conducteur la vitesse de sécurité pour éviter la collision. Dans un premier temps, la plupart des systèmes améliorent la séquence vidéo (par exemple, augmenter le contraste et l'éclairage, utiliser des filtres médians, etc.).

Pour la détection et le suivi d'objets, il existe des systèmes qui utilisent des algorithmes descriptifs des caractéristiques (comme SURF, Haarlike) ou des techniques de traitement d'images (comme le filtre gaussien dérivé, le filtre de Kalman, ...etc.) et des algorithmes d'apprentissage automatique (B. Adboost, méthode SVM, hair classifier) [56], d'autres systèmes utilisent également des algorithmes utilisant l'architecture CNN (eg YOLO, R CNN, fast RCNN) [5]. Pour estimer la

distance, Muhammad Abdul Haseeb et al. [38] ont proposé une nouvelle méthode, cette méthode basée sur un réseau neuronal multicouche caché appelé DisNet, qui est utilisée pour apprendre et prédire la distance entre l'objet et la caméra, Hasan et al. [5] ont proposé une méthode pour estimer la distance entre les véhicules en utilisant la vision stéréo, mais elle ne peut mesurer la distance entre les véhicules que dans un rayon de 20 m.

Pour l'estimation de la vitesse, Yu et al. [5] a utilisé la méthode d'étalonnage de la caméra pour obtenir la relation entre la caméra et le monde réel, ensuite il utilise la méthode d'ajustement de l'image pour calculer le changement de position de la caméra du véhicule, Wang [6] a utilisé la connaissance a priori de la taille du véhicule pour suivre les objets, puis a calculé la différence entre les images de la vidéo pour calculer la vitesse du véhicule.

#### **2.4.3.5. Le système de la détection des feux de circulation**

Le système de reconnaissance des feux de circulation est devenu très important pour les ADAS basés sur la vision. Il fournit des informations essentielles au conducteur aux intersections et aux passages pour piétons et peut réduire le nombre d'accidents dus à la distraction aux feux de circulation. Il détecte le feu tricolore en milieu urbain puis reconnaît son état pour alerter le conducteur. Les chercheurs ont proposé des méthodes et des algorithmes pour créer ce système. TaeHyun H et al [47] ont proposé en 2006 une approche de détection de feux tricolores qui consiste à établir des seuils de couleur en haut de l'image entière au moyen d'une convolution gaussienne, pour détecter l'émission de lumière des feux de circulation. GwangGook. LEE et al [48] ont proposé une méthode qui détecte le feu de circulation et reconnaît le feu qui s'allume (rouge, vert ou orange). Dans un premier temps, on applique l'algorithme d'amélioration de la discrimination des couleurs rouge et vertes (un algorithme de prétraitement) pour modifier l'intensité des couleurs rouge et vertes, ensuite on applique la méthode de segmentation en régions et le filtre de région pour localiser le feu de circulation. Enfin, on utilise les architectures de LeNet (voir chapitre 1) pour classer le sémaaphore et AlexNet (voir chapitre 1) pour classer et éclairer.

### **2.5. Les avantages des ADAS**

### **2.5.1. Le potentiel de réduction des accidents**

Des études montrent qu'ADAS réduit le nombre de décès de piétons et de cyclistes jusqu'à 30 % ou plus dans certaines circonstances. Une étude menée par les institutions d'assurance-accidents en Allemagne depuis 2013 estime que l'introduction complète de caméras de recul avec freinage automatique pourrait éviter la mort de piétons tués par des camions en Allemagne [36].

### **2.5.2. Coût réduit :**

Les ADAS sont généralement peu coûteux (des centaines à des milliers de dollars), en plus de la facilité de son installation [36].

### **2.5.3. La connexion entre les ADAS et le système télématique :**

Les technologies ADAS avancées peuvent être liées à des systèmes télématiques qui permettent aux conducteurs d'être informés des changements urbains dans les villes, de leur capacité à détecter les comportements dangereux dans les véhicules, ainsi que la capacité d'accéder aux données qui les identifient [36].

### **2.5.4. L'utilisation des technologies FCW et AEB**

L'efficacité des technologies FCW et AEB utilisées dans l'ADAS peut réduire considérablement les collisions routières potentielles et rendre la conduite plus facile, plus sûre et plus agréable [36].

## **2.6. Les défis majeurs des ADAS**

### **2.6.1. Les changements météorologiques**

L'un des principaux problèmes d'ADAS est que les performances du système sont fortement affectées par les conditions météorologiques et environnementales changeantes, par exemple pour les applications ADAS qui utilisent une caméra dans des conditions d'éclairage défavorables (trop lumineux / trop sombre, etc.) [36].

### **2.6.2. La consommation d'énergie**

ADAS implique l'exécution de plusieurs algorithmes complexes qui entraînent une consommation d'énergie élevée et une dissipation thermique élevée. En raison de la disponibilité

limitée de l'énergie dans les véhicules, il est essentiel de minimiser la consommation d'énergie du système embarqué utilisé par ADAS [36].

### **2.6.3. La sécurité**

Les véhicules modernes sont de plus en plus connectés à de nombreux systèmes, tels que le Wi-Fi, la communication en champ proche, etc. Cela permet au véhicule de détecter et de recevoir différentes informations, mais le rend également plus vulnérable aux attaques. Il y a eu de nombreuses attaques contre des véhicules. Par exemple, le système télématique d'un Jeep Cherokee a été piraté pour accélérer, freiner et couper le moteur. Ce problème peut avoir des conséquences graves dans ADAS [36].

### **2.6.4. Les contraintes géospatiales**

Tous les pays (ou certaines régions au sein d'un même pays) n'adhèrent pas uniformément aux mêmes conventions de signalisation et de routage, ce qui signifie que les algorithmes utilisés dans ADAS sont souvent formés en un seul endroit et sont difficiles à utiliser de manière cohérente et efficace ailleurs [36].

## **2.7. Conclusion**

Dans ce chapitre nous avons présenté le concept de base des systèmes d'aide à la conduite ADAS et le rôle important de la technologie de vision par ordinateur et des techniques de traitement et de pré-traitement d'images dans le traitement d'images définit le nombre d'accidents et leur intégration dans les véhicules autonomes. Dans le chapitre suivant, nous décrivons un ADAS basé sur la vision qui détecte les objets sur la route (véhicules, piétons, panneaux d'arrêt, feux de circulation, etc.) puis estime la distance entre la caméra et l'objet détecté.

## Chapitre 3 : La conception de système

### 3.1. Introduction

Le processus de détection des objets dans une image passe par deux étapes. La localisation des objets, et leur classification. La localisation consiste à dessiner une boîte de délimitation autour de l'objet, et la classification consiste à associer l'objet localisé à une catégorie.

Le réseau de neurones convolutif (CNN) est l'une des techniques les plus utilisées pour localiser et reconnaître des objets. Les algorithmes de détection des objets basés sur CNN sont largement utilisés dans le développement des systèmes de sécurité et aide à la conduite (ADAS), où ils montrent plus d'efficacité et plus de fiabilité dans les situations critiques (comme l'évitement de collision, la régulation de distance et la vitesse sécurisée, etc.).

Dans ce chapitre, nous allons introduire un système ADAS basé sur la vision, en utilisant un algorithme basé sur les CNN. Notre système utilise la méthode Single Shot Detector SSD pour détecter quelques objets sur la route (des camions, des voitures, des motos, des bicyclettes, des piétons, des feux de circulation, des panneaux d'arrêts, etc.) dans une vidéo. Il permet aussi d'estimer la distance entre la webcam et l'objet détecté dans l'image (ou dans une vidéo).

### 3.2. État de l'art

Cette section comporte deux parties. La première partie décrit les approches et les méthodes proposées pour la détection des objets. La deuxième partie décrit des méthodes et des technologies proposées pour estimer la distance entre l'objectif de la caméra et l'objet détecté par une application d'ADAS.

#### 3.2.1. La détection

La détection d'objets est une technologie informatique qui utilise le traitement d'images et l'intelligence artificielle (l'apprentissage automatique et l'apprentissage profond) pour détecter des instances et des objets dans une image numérique ou une vidéo. Ci-dessous, nous allons citer quelques études réalisées dans le domaine de la détection des objets.

Shaoqing Ren, et al [24] étudiaient la performance de Faster RCNN (voir le chapitre 1) avec VGG16, ils trouvaient que la valeur de la précision moyenne est 24.9% et que l'algorithme peut détecter les objets dans une vidéo de cadence de 5 images par seconde et cela prend 250 millisecondes de temps avec l'utilisation de GPU. Liu, Wei, et al [27] étudiaient la performance de SSD300 (voir le chapitre 1) avec VGG16, ils trouvaient que la valeur de la précision moyenne est 20.9% et que l'algorithme peut détecter les objets dans une vidéo de cadence de 59 images par seconde et cela prend 130 millisecondes de temps avec l'utilisation de GPU. Ils ont utilisé l'ensemble de données de MS COCO (voir le chapitre 1) comme une référence pour faire l'entraînement et l'évaluation de l'algorithme de détection des objets. Dans notre cas, le système a besoin d'une détection rapide et plus de précision, car il est intégré sur un objet en mouvement, donc nous avons choisi le dernier algorithme pour faire la détection des objets de la circulation routière.

Avec l'intérêt croissant de l'industrie pour les applications d'ADAS et les véhicules autonomes, c'est devenue un domaine d'exploration majeur en vision par ordinateur. Dans le précédent chapitre, nous avons cité des approches et des méthodes proposés dans le domaine d'ADAS basé sur la vision (voir le chapitre 2), et qui sont utilisés dans les systèmes de détection des feux de la circulation, la détection et reconnaissance des panneaux de signalisation et aussi la détection des piétons.

Dans cette section, nous allons ajouter quelques approches. B.Lim, et al [49] ont proposé une méthode pour la détection des véhicules, cette méthode est basée sur l'utilisation de l'algorithme STIXEL pour extraire les régions d'intérêts (voir le chapitre 1) et l'architecture CNN pour classifier les véhicules dans l'image. Ils ont utilisé l'ensemble des données KITTI pour faire l'entraînement et l'évaluation.

Henry X. Liu, et al [50] ont proposé un système de détection et de reconnaissance de panneaux STOP. Ce système comporte deux modules principaux. La détection et la reconnaissance. Dans le module de détection, le seuillage des couleurs en teinte, saturation et valeur de l'espace colorimétrique est utilisé pour segmenter l'image. Les caractéristiques des panneaux STOP sont étudiées et utilisées pour le détecter. Pour le module de

reconnaissance, un réseau de neurones est formé pour effectuer la classification et un autre est formé pour effectuer la validation.

### 3.2.2. L'estimation de la distance

Ces dernières années, l'estimation de la distance est devenue une solution importante dans les systèmes d'aide à l'évitement de collision (une application d'ADAS). Cette section est une brève présentation des méthodes actuellement utilisés qui produisent des estimations de distance.

M. Hansard, et al [51] ont proposé la caméra de temps de vol (ToF) qui peut également fournir des informations sur la distance. Une caméra 3D à temps de vol fonctionne en éclairant la scène à l'aide d'une source de lumière modulée, puis en observant la lumière réfléchie, on peut déduire la distance. Les caméras ToF produisent une image en profondeur, où chaque pixel est encodé avec la distance du point correspondant de la scène capturée. La portée de la distance est limitée par la puissance du faisceau lumineux et est également susceptible de provoquer un flou de mouvement.

Abdul Haseeb Muhammad et al [38] ont proposé une méthode d'apprentissage automatique Multi-DisNet qui consiste à estimer la distance d'un objet dans le cadre de référence du laser, qui est à la même distance de l'objet que le cadre de référence de la caméra, grâce à une entrée appelée vecteur de caractéristique V qui contient les caractéristiques de la boîte de délimitation de l'objet détecté dans les images capturées par une caméra et la vérité terrain qui représente la distance à l'objet mesurée par le scanner du laser.

Hoi-Kok Cheung, et al [52] ont proposé une méthode de calcul d'homographie pour compenser les mauvais alignements de l'orientation de la caméra, qui permet d'estimer la distance entre le véhicule qui précède et la caméra numérique avec une grande précision.

B. Lim, et al [49] ont utilisé des images capturées par une caméra stéréo et la formule ci-dessous pour estimer la distance aux objets.

$$Z=f * B / d \quad (3.1)$$

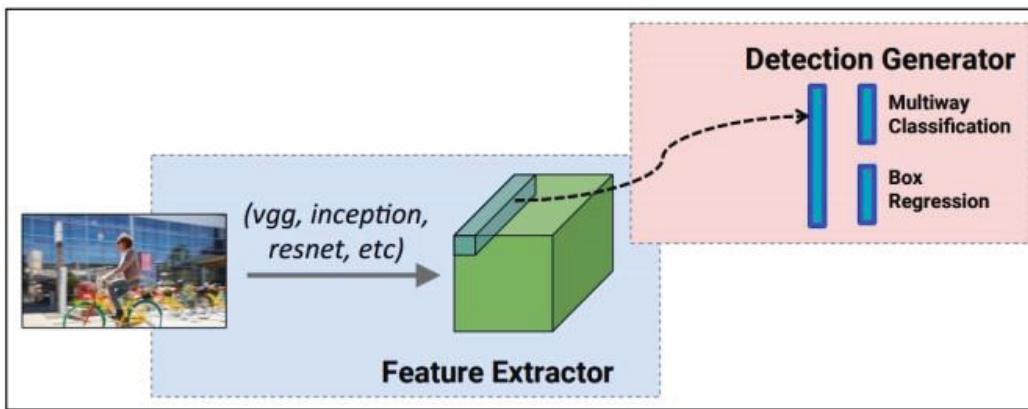
Z est la distance entre l'objet et la caméra, f est la distance focale, B est la distance entre les deux objectifs de la caméra stéréo et d est la disparité représentative de la région d'intérêt.

La disparité représentative est sélectionnée comme une valeur maximale dans l'histogramme des valeurs de disparité.

### 3.3. La description de l'algorithme SSD

Notre système utilise la méthode SSD (nous avons définis le SSD dans le chapitre 1) qui fait partie des algorithmes de la détection des objets. Cet algorithme SSD est composé de deux parties principales [53] (Voir figure 3.1) .

- L'extraction des cartes de convolutions (cartes d'entités) en utilisant un réseau de base (modèle de classification).
- La détection des objets en utilisant des filtres convolutifs.



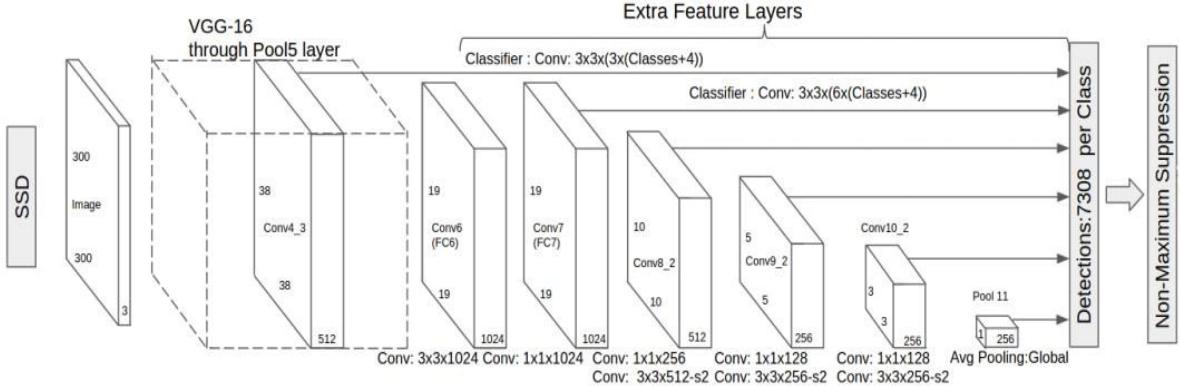
**Figure 3. 1.** Architecture et composition de SSD [53].

Nous allons décrire chacune de ces deux étapes.

#### 3.3.1. L'extraction des cartes de convolution

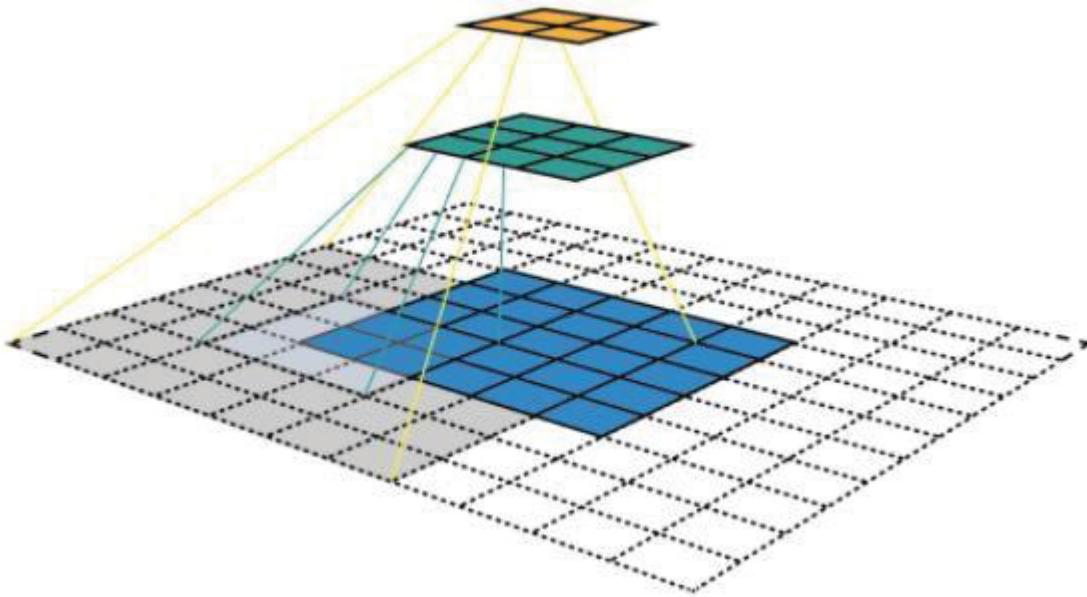
Le but de l'extraction des cartes d'entités est de réduire une image de taille fixe à un ensemble variable de caractéristiques visuelles. Les modèles de classification d'images sont généralement construits en utilisant des méthodes d'extraction de caractéristiques visuelles puissantes comme les réseaux neurones convolutifs ou CNN (Deep Learning). Pour la détection d'objets, les algorithmes utilisent généralement des modèles de classification (réseaux de neurones convolutifs) d'images pour extraire des caractéristiques visuelles.

L'architecture originale de SSD est construite à la base de VGG-16 (voir le chapitre 1), mais en ajoutant des couches entièrement connectées. La raison d'utiliser VGG-16 en tant que réseau de base est sa bonne performance dans les tâches de classification d'image [53].



**Figure 3. 2.** L'architecture de réseau de base (VGG-16) avec une détecteur SSD [27].

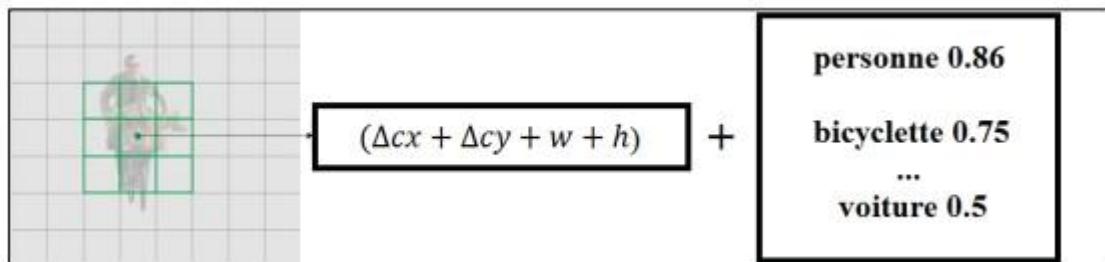
Nous présentons un exemple du fonctionnement d'extraction des cartes. Nous commençons par la couche inférieure (5x5), puis nous appliquons une convolution qui aboutit à la couche intermédiaire (3x3) où une entité (pixel vert) représente une région 3x3 de la couche d'entrée (couche inférieure). Ensuite nous appliquerons la convolution à la couche intermédiaire et obtenons la couche supérieure (2x2) où chaque entité correspond à une région 7x7 sur l'image d'entrée. Des cartes d'entités font référence à un ensemble d'entités créées en appliquant le même extracteur d'entités à différents emplacements de la carte d'entrée dans une fixation de fenêtre coulissante. Les entités d'une même carte d'entités ont le même champ réceptif (le champ réceptif est défini comme la région dans l'espace d'entrée que la caractéristique d'un CNN particulier) et recherchent le même modèle mais à des emplacements différents [54]. La figure ci-dessous représente la visualisation des cartes des caractéristiques CNN et du champ réceptif de notre exemple. Nous pouvons dire que l'architecture de base de SSD utilise le même fonctionnement de l'exemple donné mais il a plusieurs couches de convolution.



**Figure 3. 3.** La visualisation des cartes des caractéristiques CNN et du champ réceptif [54].

### 3.3.2. La localisation et la classification des objets

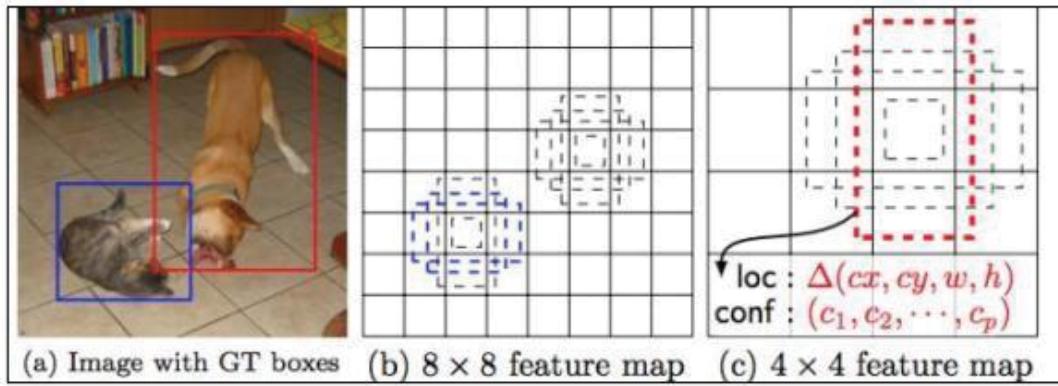
Le SSD calcule les scores de localisation et de classe à l'aide de petits filtres de convolution. Donc après l'extraction des cartes d'entités, SSD applique des filtres de convolution  $3 * 3$  pour chaque cellule pour faire des prédictions [27]. Chaque filtre produit  $(n + 4)$  valeurs (figure 3.4) : 4 scores ou coordonnées de la boîte de délimitation et  $n$  scores pour chaque classe de détection.



**Figure 3. 4.** L'utilisation d'un filtre de  $3*3$  pour la localisation et la classification [27].

Au début, nous avons décrit comment SSD détecte des objets d'une seule couche. En fait, il utilise plusieurs couches (cartes d'entités multi-échelles) pour détecter des objets indépendamment. Comme le réseau de base (extracteur) réduit progressivement la dimension spatiale, la résolution des cartes d'entités diminue également. Le SSD utilise des

couche à plus haute résolution pour détecter les objets plus petits et les couches à résolution plus faible pour détecter les objets à plus grande échelle. L'utilisation de cartes d'entités multi-échelles améliorent considérablement la précision, parce qu'elles nous aident à faire la détection de différents objets à différentes échelles dans les cartes d'entités [53]. Dans l'image ci-dessous (figure 3.5), on peut noter que le chat est détecté dans la carte d'entité  $8*8$  avec 2 cases, et le chien est détecté dans la carte d'entité  $4*4$  [27].



**Figure 3.5.** Utilisation des cartes d'entités de différentes couches pour la détection [27].

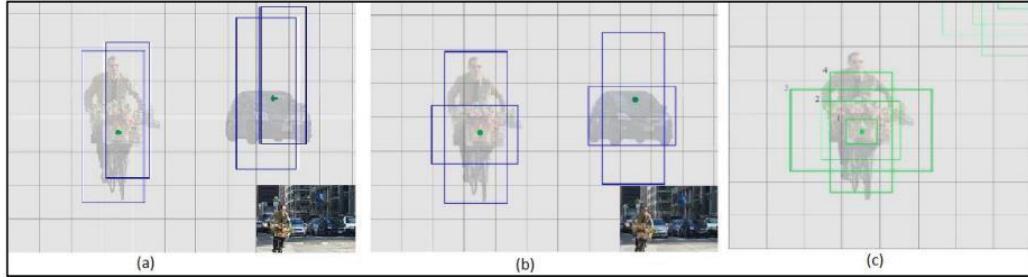
### 3.3.3. Les Boites de délimitation par défaut

Nous savons déjà comment faire la classification des objets, mais Comment pouvons-nous prévoir les boîtes de délimitation ? Nous associons un ensemble de boites de délimitation par défaut à chaque cellule de carte d'entités, ces boites par défaut sont appliquées à plusieurs cartes d'entités de différentes résolutions.

Les zones par défaut sont attachées aux cartes d'entités de manière convolutive, de sorte que la position de chaque instance de boîte par défaut est fixée par rapport à sa cellule correspondante. À chaque cellule, les prédictions des décalages sont faites par rapport aux formes de la boîte par défaut dans la cellule, ainsi que les scores de probabilité qui indiquent la présence d'une instance de classe dans chacune de ces zones [53]. Les prédictions de la boîte de délimitation, ci-dessous (figure 3.6.a) fonctionnent bien pour une catégorie mais pas pour d'autres. Si nos prédictions couvrent plus de formes en différentes position, comme celle-ci dessous (figure 3.6.b), notre modèle peut détecter plus de types d'objets.

Afin de minimiser la complexité, les boîtes par défaut sont présélectionnées manuellement et avec soin pour couvrir un large éventail d'objets de la vie réelle. SSD conserve également les boîtes par défaut à un minimum (4 ou 6) avec une prédiction par boîte par défaut. Au lieu d'utiliser une coordination globale pour l'emplacement de la boîte, les prédictions de la boîte de délimitation sont relatives aux boîtes de limites par défaut de chaque cellule ( $\Delta cx$ ,  $\Delta cy$ ,  $w$ ,  $h$ ), c'est-à-dire les décalages à la case par défaut dans chaque cellule [53].

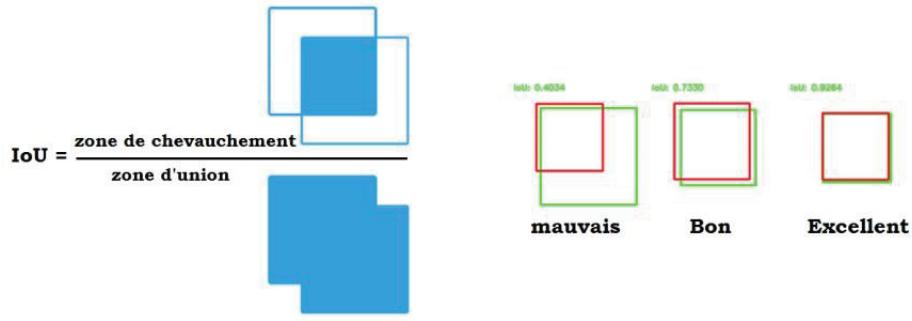
Pour chaque couche de carte d'entités, elle partage le même ensemble de zones par défaut centré sur la cellule correspondante. Mais différentes couches utilisent différents ensembles de boîtes par défaut pour personnaliser les détections d'objets à différentes résolutions. SSD définit une paire de valeur d'échelle pour chaque couche de carte d'entités [27]. Les 4 cases vertes ci-dessous (Figure 3.6.c) illustrent 4 zones de délimitation par défaut.



**Figure 3. 6.** Utilisation de plusieurs boîtes par défaut dans une cellule [27].

### 3.3.4. Les boîtes vérité et l'intersection sur l'union

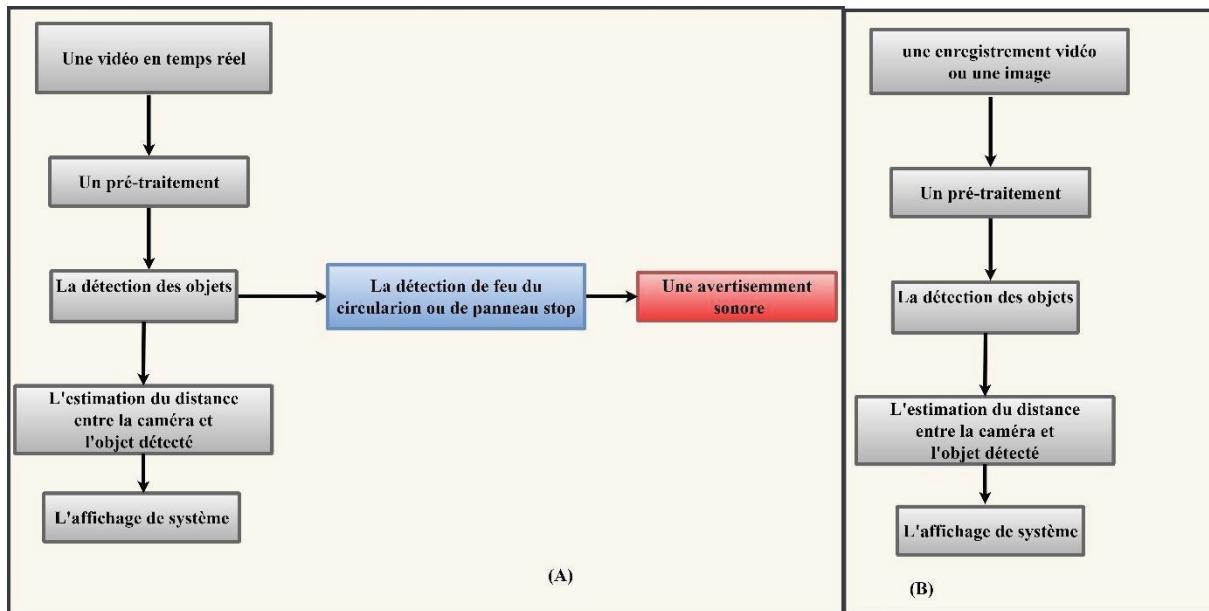
L'intersection sur l'union (en anglais Intersection Over Union (IoU)) est le rapport entre la zone intersectée sur la zone jointe pour deux boîtes de délimitation. Il est parfois appelé l'indice de Jaccard (voir figure 3.7) [55]. Par exemple pendant le temps de l'entraînement, les boîtes par défaut correspondent au rapport hauteur / largeur, à l'emplacement et à l'échelle aux boîtes de vérité. Nous sélectionnons les boîtes avec le chevauchement le plus élevé avec les boîtes englobantes de vérité. IoU (intersection over union) entre les boîtes délimitations prédites et la boîte vérité doit être supérieur à 0,5. Nous prenons enfin la case prédite avec un chevauchement maximal avec la case vérité.



**Figure 3. 7.** Diagramme d'explication d'IoU (l'indice de Jaccard) [55].

### 3.4. La conception de notre système utilisé pour la détection

Notre système détecte les objets de la circulation routière (voiture, camion, piéton, bicyclette, feux de circulation, moto, bus, panneau stop, etc.) dans les images ou les enregistrements vidéo ou les vidéos en temps réel (les vidéos capturées par une webcam). La figure 3.8.a représente le diagramme général de notre système, qui permet de détecter les objets de la circulation routière dans une image ou un enregistrement vidéo. La figure 3.8.b représente le diagramme général de notre système, qui permet cette fois ci de détecter les objets de la circulation routière dans une vidéo en temps réel.



**Figure 3.8.** La conception en générale du notre système.

### 3.4.1. La détection et la reconnaissance des objets de circulation routière

#### 3.4.1.1. L'architecture de réseau de base

Pour notre modèle SSD300 (SSD300 signifier que l'image d'entrée est de taille 300\*300) nous avons utilisé l'architecture ci-dessous (figure 3.9).

- 11 couches de convolution.
- 6 couches de convolution doublées (1,2,6,7,8,9).
- 3 couches de convolution triplées (3,4,5).
- Il existe une couche utilisant la technique de convolution dilatée pour augmenter le champ de vision et réduire le coût de calcul.

Chacune des 5 premiers couches est normalisée et séparée de la suivante avec une couche de pooling 3 x 3 pour réduire la quantité des paramètres.

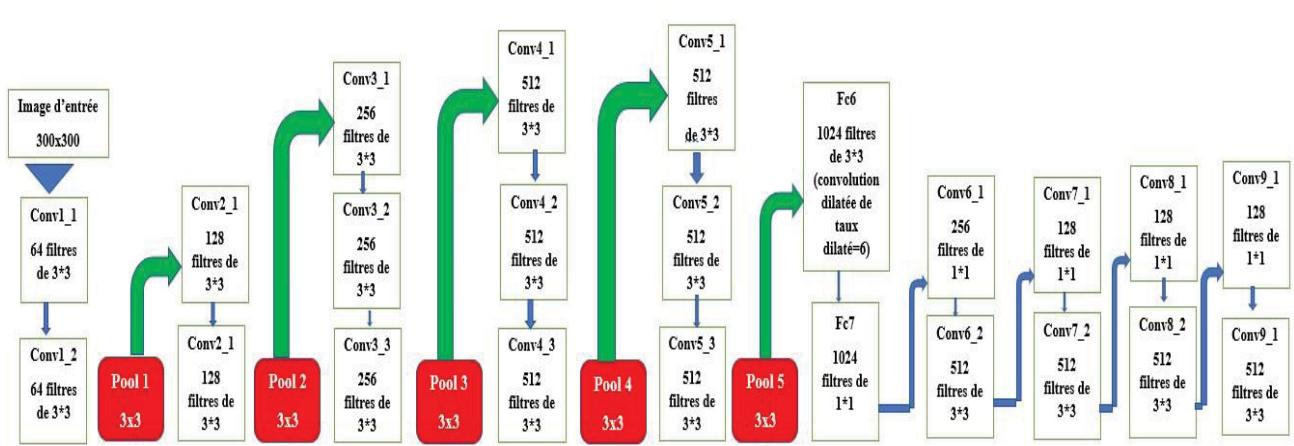
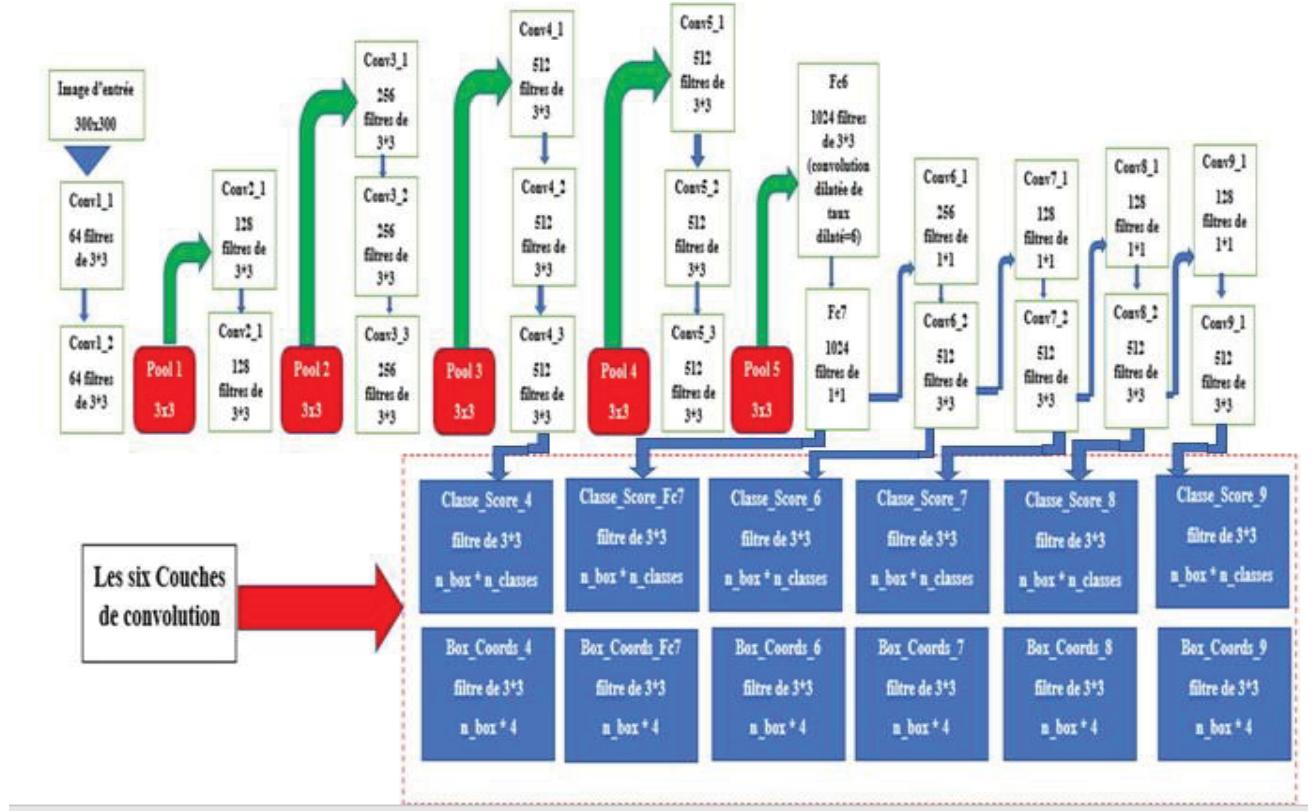


Figure 3. 9. Diagramme de réseau de base de modèle SSD300.

#### 3.4.1.2. Les couches du localisation et classification de modèle SSD

Pour notre modèle, nous avons utilisé 6 couches de convolutions (figure 3.10) pour la détection des objets (localisation, classification) qui sont appliqués à différentes cartes d'entités dans différentes échelles dans notre réseau de base (CONV4\_3, CONV\_Fc7, CONV6\_2, CONV7\_2, CONV8\_2, CONV9\_2).

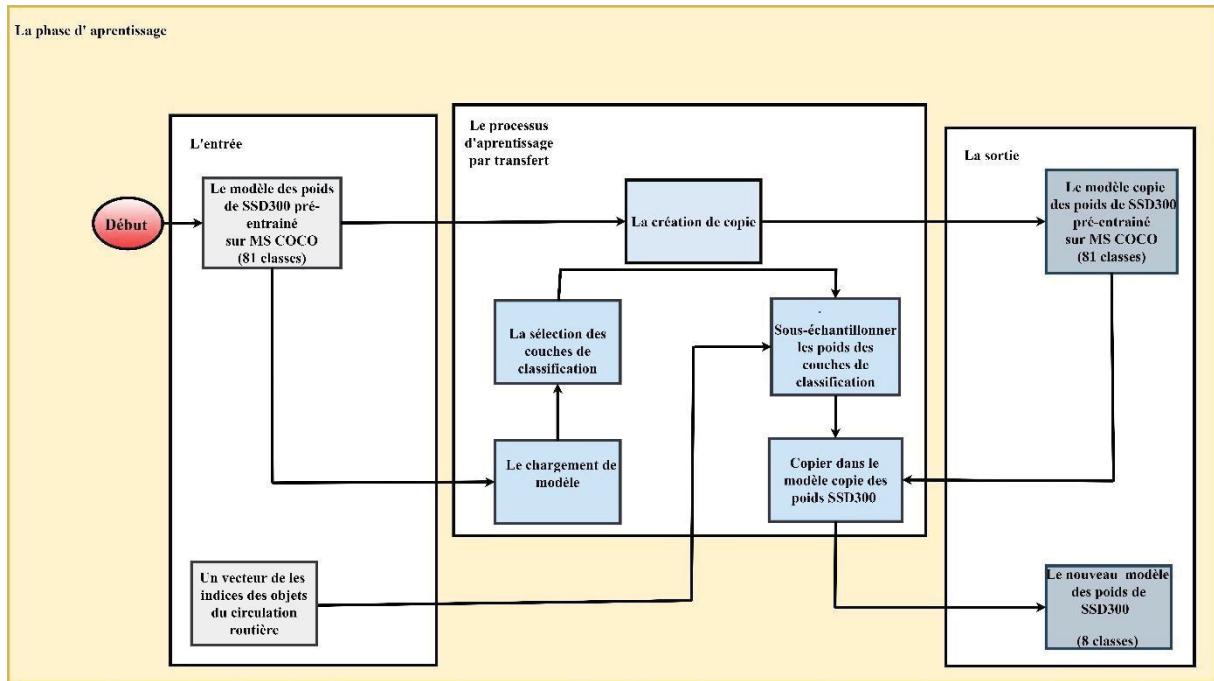


**Figure 3. 10.** Addition des couches de localisation et classification au réseau de base.

### 3.4.1.3. La phase d'apprentissage par transfert de notre modèle

L'apprentissage par transfert est une technique d'apprentissage automatique, où les connaissances acquises pendant la formation sur un type de problème sont utilisées pour former d'autres tâches, c'est-à-dire lorsqu'on effectue un apprentissage par transfert, on prend un modèle qui a été formé sur quelque chose et on utilise une partie ou la totalité de ce modèle pour entraîner un nouveau modèle, en mettant à jour les poids de ce nouveau modèle [55].

Dans notre modèle SSD300, nous avons utilisé un modèle pré-entraîné sur la base de données MS COCO (voir le chapitre 1), qui contient 81 catégories, parmi celles-ci 8 catégories font partie des objets de la circulation routière. Nous avons utilisé l'apprentissage par transfert pour créer un nouveau modèle qui détecte les objets de la circulation routière. Le diagramme ci-dessous représente les étapes de création de notre nouveau modèle SSD300.



**Figure 3.11.** Le diagramme de la phase d'apprentissage de système.

### Rechercher des indices des objets de la circulation routière et créer des vecteurs

Nous devons rechercher dans un premier temps manuellement les indices d'objets de la circulation routière dans l'ensemble des données MS COCO (par exemple l'indice 1 représente la classe piéton, l'indice 2 représente la classe bicyclette, ...etc.). Ensuite, nous devons créer un vecteur contenant les noms des objets de la circulation routière (voiture, camion, piéton, bicyclette, feux de circulation, moto, bus, panneau stop) et un vecteur de ces indices (3, 8, 1, 2, 10, 4, 6, 12). En utilisant les indices des objets, nous allons sélectionner les neurones qui prédisent la localisation des 4 boîtes qui englobent les objets de la circulation routière et qui prédisent aussi les classes.

### Le chargement de modèle SSD pré-entraîné et la création d'une copie

Dans cette étape, à partir du modèle SSD300 pré-entraîné sur MSCOCO nous extrayons les poids dont nous avons besoin dans notre nouveau modèle.

### La sélection des couches de classification

A ce niveau, nous devons déterminer exactement les poids des couches qui classifient les objets de la circulation routière que nous devons sous-échantillonner (les

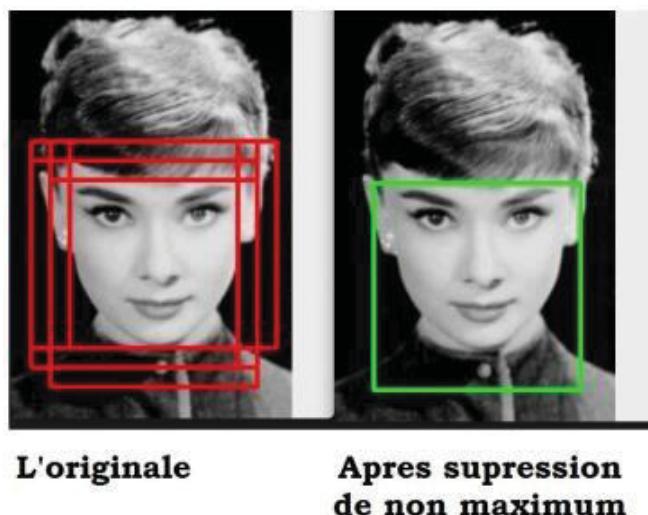
couches de classification), mais les poids des autres couches restent inchangés. Les couches de classification dans SSD300 sont : classes\_score\_4, classes\_score\_Fc7, classes\_score\_6, classes\_score\_7, classes\_score\_8 et classes\_score\_9 (voir la figure 3.8).

## Sous-échantillonner les poids des couches de classification

Dans la dernière étape, nous allons obtenir le noyau et les tenseurs de biais (les biais de neurones, les neurones qui classifient les objets de la circulation routière à partir du modèle de pondérations source (SSD300 pré-entraîné sur MS COCO). Ensuite, nous calculons les indices de sous-échantillonnage pour les derniers neurones. La taille et le nombre du noyau restent inchangés. Enfin, nous remplaçons le noyau correspondant et les tenseurs de biais dans notre modèle de pondérations de destination (le nouveau modèle SSD300 qui détecte les objets de la circulation routière) par notre noyau sous-échantillonné et nos tenseurs de biais nouvellement créés.

#### 3.4.1.4. Suppression de non maximum

Étant donné le grand nombre de boîtes générées lors d'une passe réseau de SSD au moment de l'inférence (phase d'utilisation), SSD utilise une suppression non maximale pour supprimer les prédictions pointant vers le même objet (filtre). La figure 3.12 illustre cette opération. SSD trie les prédictions par les scores de confiance, partant de la prédition de confiance supérieure, SSD évalue si les boîtes de limites prédictes précédemment ont une IoU supérieure à 0,45 avec la prédition actuelle pour la même classe. Si trouvé, la prédition actuelle sera ignorée.



**Figure 3. 12** La suppression de non maximum [54].

### 3.4.2. Estimation de la distance entre la caméra et l'objet détecté

Notre système utilise une méthode simple et efficace pour estimer la distance entre la boîte englobante de l'objet détecté et l'objectif de la caméra. Dans ce qui suit, nous décrivons cette méthode.

Tout d'abord, nous allons utiliser les coordonnées des points de la boîte de délimitation prédictive par notre modèle SSD300 pour identifier la hauteur et la largeur de la boîte de délimitation. Les points utilisés sont le point de plus bas à gauche et le point le plus haut à droite qui aide à créer une boîte englobante autour de l'objet cible.

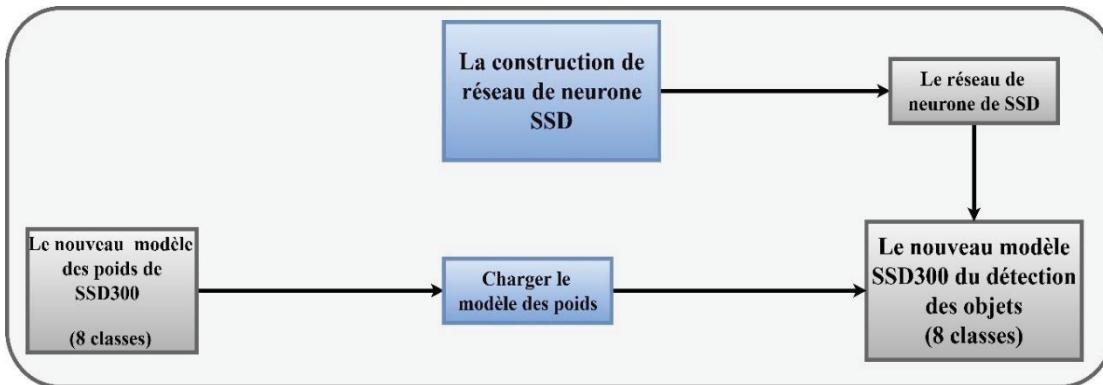
Enfin, nous appliquons cette formule pour estimer la distance [56].

$$\text{Distance} = (2 * 3.14 * 180) / (\text{la hauteur de la boîte} + \text{la largeur de la boîte} * 360) * 1000 + 3.$$

### 3.4.3. La phase d'utilisation de modèle SSD300 et l'estimation de la distance

#### 3.4.3.1. La construction de modèle

Tout d'abord, nous avons créé le réseau de neurones, puis, nous avons chargé le modèle des poids que nous avons obtenus dans la phase précédente. Le diagramme ci-dessous représente les étapes de la construction.



**Figure 3. 13.** Le diagramme représente les étapes de la construction.

#### 3.4.3.2. Acquisition d'images et le pré-traitement

Ensuite, nous extrayons les frames à partir d'un enregistrement vidéo ou une vidéo en temps réel (une webcam) et nous appliquons des algorithmes de pré-traitement pour changer

la taille d'image en 300\*300, car la taille d'image d'entrée dans notre modèle SSD300 doit être 300\*300. La figure 3.14 représente le diagramme qui représente cette opération.

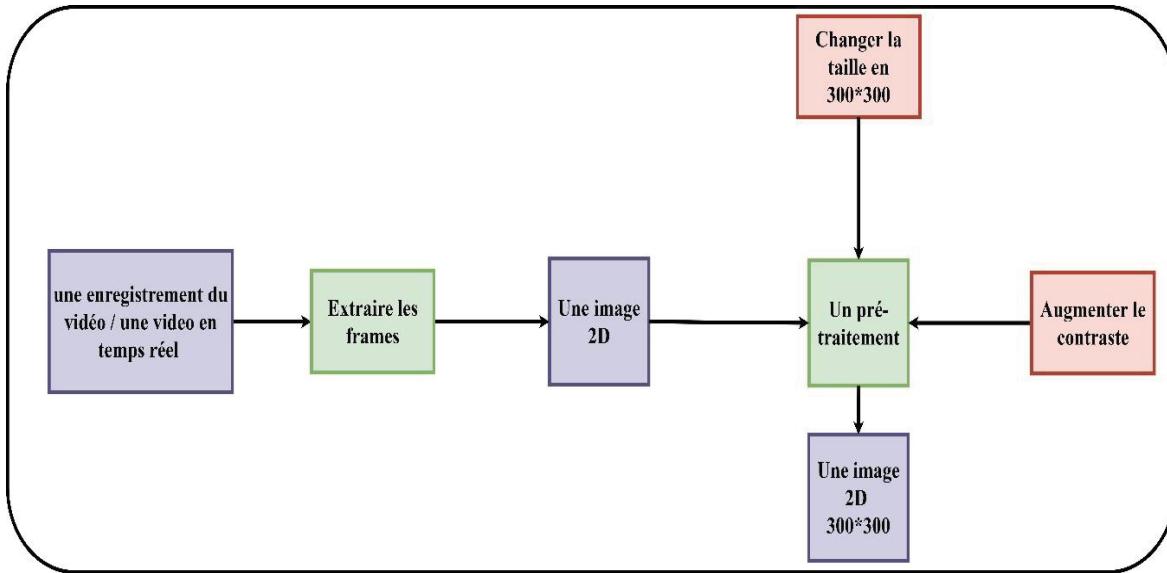
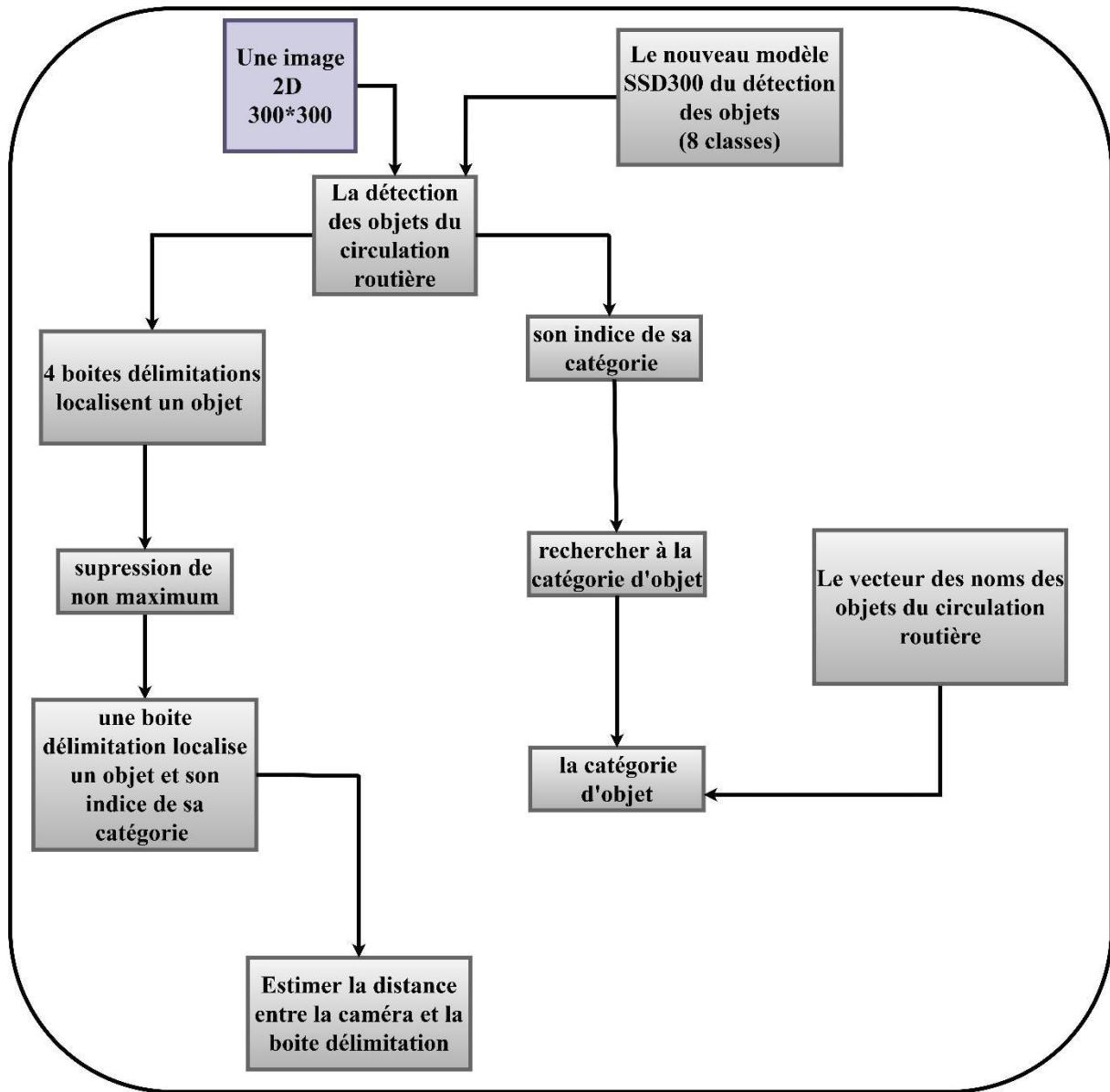


Figure 3.14. Le diagramme représente lecture du vidéo et le pré-traitement.

#### 3.4.3.3. Détection des objets et l'estimation du distance

La figure 3.13 représente le diagramme représentant le processus utilisé par notre système pour localiser et reconnaître les objets aux alentours du véhicule, et estimer sa distance par rapport à la caméra. Dans la méthode de suppression non maximum : les boîtes par défaut avec un seuil de perte de confiance inférieur à 0,01 et IoU inférieures 0,45 sont ignorées et seules les N prédictions supérieures sont conservées. Cela garantit que seules les prédictions les plus probables sont conservées par le réseau, tandis que les moins probables sont supprimées.



**Figure 3.15.** Le diagramme représente le processus de détection des objets et l'estimation de la distance.

#### 3.4.3.4. L'affichage du système

Notre système est sensé détecter les objets suivants : les voitures, les camions, les bus, les motos et les bicyclettes. Il affiche un rectangle bleu autour de ces objets avec leur label et sa distance estimée. Si le système détecte les objets suivants : les feux de circulation, les panneaux de STOP, il affiche un rectangle rouge autour de ces objets avec leur label et sa distance estimée. Notre système peut aussi avertir le conducteur pour les feux de circulation ou en cas de panneaux stop par un signal sonore.

### 3.5. Conclusion

Lors de ce chapitre, nous avons décrit la conception détaillée du système de détection et de reconnaissance des objets de la circulation routière, ainsi que l'estimation de la distance par rapport à ces objets. Nous avons décrit dans un premier temps la méthode de détection SSD, ainsi que le transfert d'apprentissage à partir d'un modèle pré-entraîné pour qu'il soit adapté à nos besoins. Nous avons par la suite défini la technique utilisée pour estimer la distance entre la caméra (une webcam) et les objets détectés. Dans le chapitre suivant, nous allons décrire l'implémentation de notre système.

## Chapitre 4 : L'implémentation et les résultats obtenus

### 4.1. Introduction

Dans ce chapitre, nous allons décrire la mise en œuvre des différentes étapes de notre système proposé pour la détection des objets de la circulation routière, ainsi que l'estimation de la distance. Dans un premier temps, nous allons commencer par présenter le langage de programmation et les outils qui ont été utilisés dans le développement de notre application, ensuite nous présentons les algorithmes utilisés, ainsi que les résultats obtenus dans diverses situations.

### 4.2. Les outils

Dans cette section, nous décrivons les différents outils pour la mise en œuvre de notre système.

#### 4.2.1. Le matériel

Notre configuration matérielle inclut les dispositifs suivants :

- Modèle : HP ProBook 4540s.
- Processeur : Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz.
- Mémoire : 6.00 Go.
- Écran : 17.3 pouces.
- Système d'exploitation : Windows 10, 64 bits.

#### 4.2.2. Environnements et outils de développement

Le langage de programmation et les bibliothèques utilisés pour effectuer ce travail sont décrits dans ce qui suit.

##### Python

Python est un langage de programmation interprété, orienté objet, de haut niveau avec une sémantique dynamique lancé par GUIDO VAN ROSSUM, facile à apprendre et disponible gratuitement pour toutes les plateformes. Il est très utilisé par une grande communauté de développeurs et de programmeurs. Les packages (bibliothèques) de python encouragent la modularité et la réutilisation du code.

## **OpenCV**

Open Source Computer Vision (OpenCV) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de traitement d'images et de vision par ordinateur, accessibles par des API pour les langages C, C++ et Python. Elle est distribuée sous une licence BSD (libre) pour toutes les plateformes. La bibliothèque OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs.

## **NumPy**

NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

## **TensorFlow**

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des Framework les plus utilisés pour l'apprentissage profond. Son nom est inspiré du fait que les opérations courantes sur des réseaux de neurones sont faites sur des tableaux multi-dimensionnelles, appelées Tenseurs. Un Tenseur à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google utilisent TensorFlow : Gmail, Google Photos et reconnaissance de la parole.

## **Keras**

Keras est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur les Frameworks TensorFlow, Theano ou CNTK. Elle a été développée en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches. Elle a été développée dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur travaillant chez Google.

<https://www.python.org/>

<https://opencv.org/>

<http://www.numpy.org/>

<https://www.tensorflow.org/>

<https://keras.io/>

<https://riverbankcomputing.com/software/pyqt/>

<https://www.sublimetext.com>

## PyQt5

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt. Il permet ainsi de créer des interfaces graphiques en Python. Il a été créé par la société britannique Riverbank computing.

## Sublime Text

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Il a été créé par Jon Skinner. C'est un éditeur de texte gratuit prenant en charge plusieurs langages de programmation différents, dont Python, JavaScript, C/C++ etc., tout comme Visual studio code et Atome.

### 4.3. L'implémentation des composantes de notre système

Dans cette section, nous allons décrire les deux parties de notre système, à savoir la partie de la détection des objets de la circulation routière et la partie d'estimation des distances.

#### 4.3.1 Détection des objets dans notre système

##### 4.3.1.1. La création du modèle des poids SSD300 (8 classes)

Comme nous avons expliqué dans le chapitre précédent, nous devons obtenir un modèle qui permet de détecter les objets de la circulation routière. En utilisant le langage Python avec L'environnement tensorflow et des bibliothèques bien connues dans le domaine du Deep learning (CNN), nous avons appliqué la technique d'apprentissage par transfert pour déterminer ce modèle.

#### Faire une copie de modèle SSD300 pré-entraîné et la charger

Tout d'abord, nous allons faire une copie du modèle VGG\_SSD300 pré-entraîné sur la base de données MS COCO, qui a été créé par Wei Lee (le modèle .h5 est disponible dans [58]), ce modèle peut détecter 81 catégories d'objets. Ensuite, nous allons faire une copie (modèle destination) et charger les deux modèle (sources et destination), la fonction ci-dessous permet de créer cette copie.

```

weights_source_path='/content/VGG_coco_SSD_300x300_iter_400000.h5'
weights_destination_path='/content/drive/My Drive/ssd_keras-master_exact/VGG_coco_SSD_300x300_iter_400000_subsampled_8_classes.h5'
shutil.copy(weights_source_path, weights_destination_path)
weights_source_file = h5py.File(weights_source_path, 'r')
weights_destination_file = h5py.File(weights_destination_path)

```

**Figure 4.1** La création de la copie.

La boucle ci-dessous est divisée en deux parties. La première partie sélectionne les couches de classification dans le modèle source ensuite elle utilise le vecteur des indices qui indique les éléments qui classifient les objets de la circulation routière, elle extrait les filtres et biais qui sont utilisés pour classifier les objets de la circulation routière.

Il faut savoir que MS COCO contient 80 classes, mais le modèle a également une classe arrière-plan, ce qui fait 81 classes. Par exemple la couche 'conv4\_3\_norm\_mbox\_loc' prédit 4 boites de délimitation pour chaque position spatiale, donc la couche 'conv4\_3\_norm\_mbox\_conf' doit prédire l'une des 81 classes pour chacune de ces 4 boites. Alors le nombre des éléments est  $4 * 81 = 324$  éléments.

Nous avons choisi 8 catégories d'objets parmi les 81, ce sont les objets issus de la circulation routière, mais notre modèle aura également une classe d'arrière-plan, ce qui fait 9 classes au total. Nous devons prédire l'une de ces 9 classes pour chacune des quatre boites de délimitation à chaque position spatiale. Cela fait  $4 * 9 = 36$  éléments. Donc, dans chaque couche de classification, nous allons extraire 36 éléments.

## La sélection des couches et sous-échantillonnage des filtres et des biais

```
classifier_names = ['conv4_3_norm_mbox_conf',
                     'fc7_mbox_conf',
                     'conv6_2_mbox_conf',
                     'conv7_2_mbox_conf',
                     'conv8_2_mbox_conf',
                     'conv9_2_mbox_conf']

n_classes_source = 81
classes_of_interest = [0, 3, 8, 1, 2, 10, 4, 6, 12]
for name in classifier_names:

    bias = weights_source_file[name][name]['bias:0'].value
    height, width, in_channels, out_channels = kernel.shape
    if isinstance(classes_of_interest, (list, tuple)):
        subsampling_indices = []
        for i in range(int(out_channels/n_classes_source)):
            indices = np.array(classes_of_interest) + i * n_classes_source
            subsampling_indices.append(indices)
        subsampling_indices = list(np.concatenate(subsampling_indices))
    elif isinstance(classes_of_interest, int):
        subsampling_indices = int(classes_of_interest * (out_channels/n_classes_source))
    else:
        raise ValueError("`classes_of_interest` must be either an integer or a list/tuple.)
```

Figure 4. 2. La sélection des couches et extraction des filtres et des biais.

## Copier les filtres et les biais et créer un nouveau modèle des poids

Dans la deuxième partie de la boucle, nous créons les nouveaux tenseurs des biais et les filtres par la fonction de `sample_tensors()`, ensuite nous devons remplacer les anciens tenseurs des biais et les filtres par les nouveaux biais et les filtres dans chaque couche de classification. Finalement, nous créons un nouveau modèle des poids par la fonction `Flush()` qui permet de détecter les 8 catégories qui sont des objets.

```
new_kernel, new_bias = sample_tensors(weights_list=[kernel, bias],
                                       sampling_instructions=[height, width, in_channels, subsampling_indices],
                                       axes=[3], # The one bias dimension corresponds to the last kernel dimension.
                                       init=['gaussian', 'zeros'],
                                       mean=0.0,
                                       stddev=0.005)

del weights_destination_file[name][name]['kernel:0']
del weights_destination_file[name][name]['bias:0']

weights_destination_file[name][name].create_dataset(name='kernel:0', data=new_kernel)
weights_destination_file[name][name].create_dataset(name='bias:0', data=new_bias)

weights_destination_file.flush()
```

Figure 4. 3. Remplacer les filtres et les tenseurs dans le modèle de la destination circulation routière.

Les deux figures suivantes représentent la différence entre les couches de classification de modèle des poids pré-entraînés sur MS-COCO (voir la figure 4.4) et notre modèle des poids (voir la figure 4.5).

conv4_3_norm_mbox_conf (Conv2D) (None, 38, 38, 324)	1493316	conv4_3_norm[θ][θ]
fc7_mbox_conf (Conv2D) (None, 19, 19, 486)	4479462	fc7[θ][θ]
conv6_2_mbox_conf (Conv2D) (None, 10, 10, 486)	2239974	conv6_2[θ][θ]
conv7_2_mbox_conf (Conv2D) (None, 5, 5, 486)	1120230	conv7_2[θ][θ]
conv8_2_mbox_conf (Conv2D) (None, 3, 3, 324)	746820	conv8_2[θ][θ]
conv9_2_mbox_conf (Conv2D) (None, 1, 1, 324)	746820	conv9_2[θ][θ]

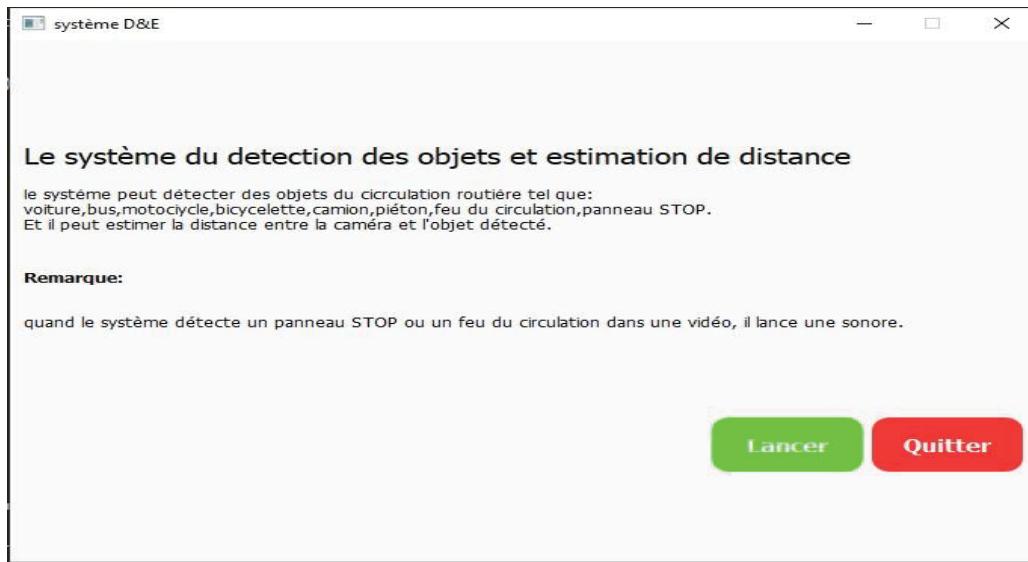
**Figure 4. 4.** Les couches de classification dans le modèle des poids SSD300 pré-entraîné sur MS-COCO.

conv4_3_norm_mbox_conf (Conv2D) (None, 38, 38, 36)	165924	conv4_3_norm[θ][θ]
fc7_mbox_conf (Conv2D) (None, 19, 19, 54)	497718	fc7[θ][θ]
conv6_2_mbox_conf (Conv2D) (None, 10, 10, 54)	248886	conv6_2[θ][θ]
conv7_2_mbox_conf (Conv2D) (None, 5, 5, 54)	124470	conv7_2[θ][θ]
conv8_2_mbox_conf (Conv2D) (None, 3, 3, 36)	82980	conv8_2[θ][θ]
conv9_2_mbox_conf (Conv2D) (None, 1, 1, 36)	82980	conv9_2[θ][θ]

**Figure 4. 5.** Les couches de classification de notre modèle SSD300.

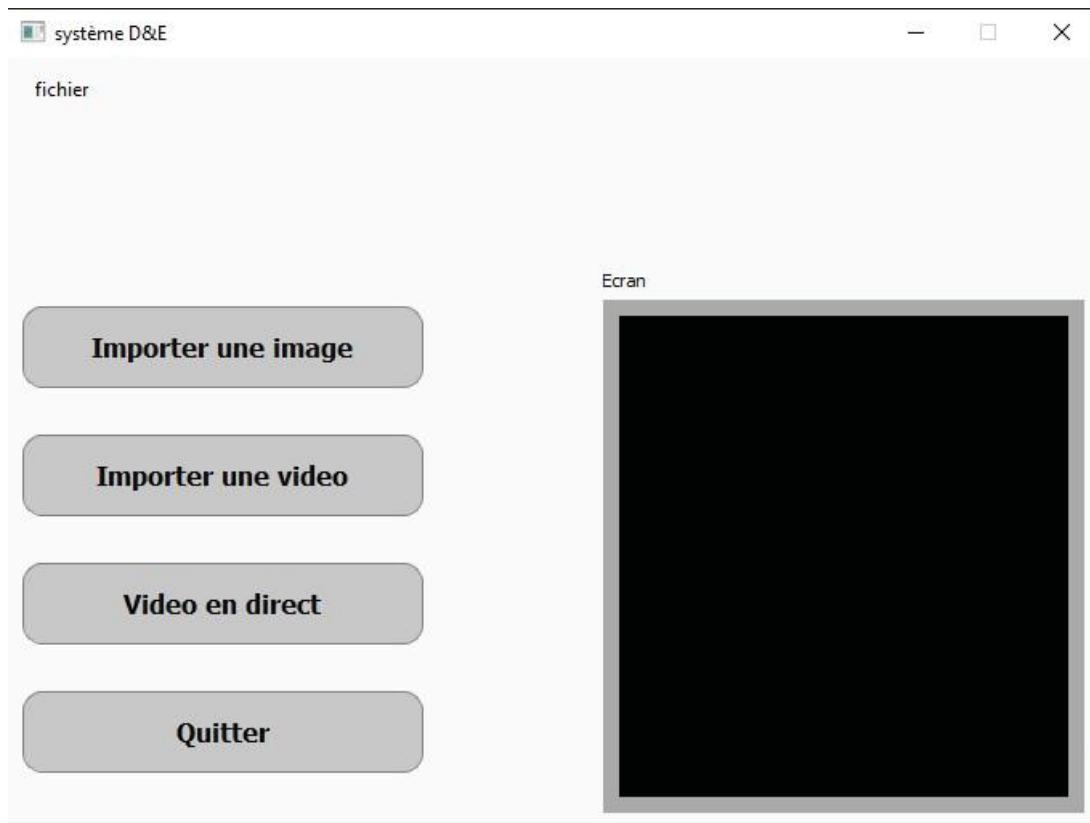
#### 4.3.1.2. La détection des objets de la circulation routière

Au début, le système va créer le réseau de SSD300 (le réseau de base et le modèle de détection), puis il charge le modèle des poids pour mettre à jour les valeurs des filtres et des biais. Donc il obtient un modèle SSD300 qui permet de détecter les objets de la circulation routière.



**Figure 4. 6.** L'interface principale de système.

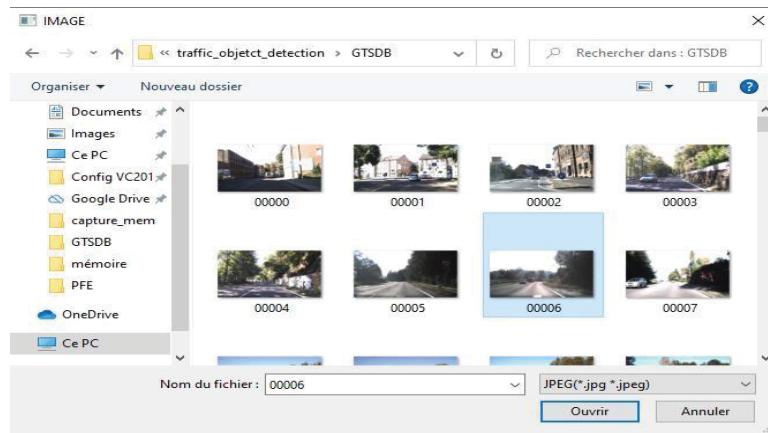
Notre système utilise ce modèle pour détecter les objets de la circulation routière dans une image ou un enregistrement vidéo ou une vidéo en temps réel.



**Figure 4. 7.** Les taches du système

- **La détection des objets dans une image**

Pour que le système puisse détecter des objets dans une image, on appuie sur le bouton « importe une image » pour choisir une image (les extensions des images sont .jpg, .jpeg, .png) comme il est montré ci-dessous.

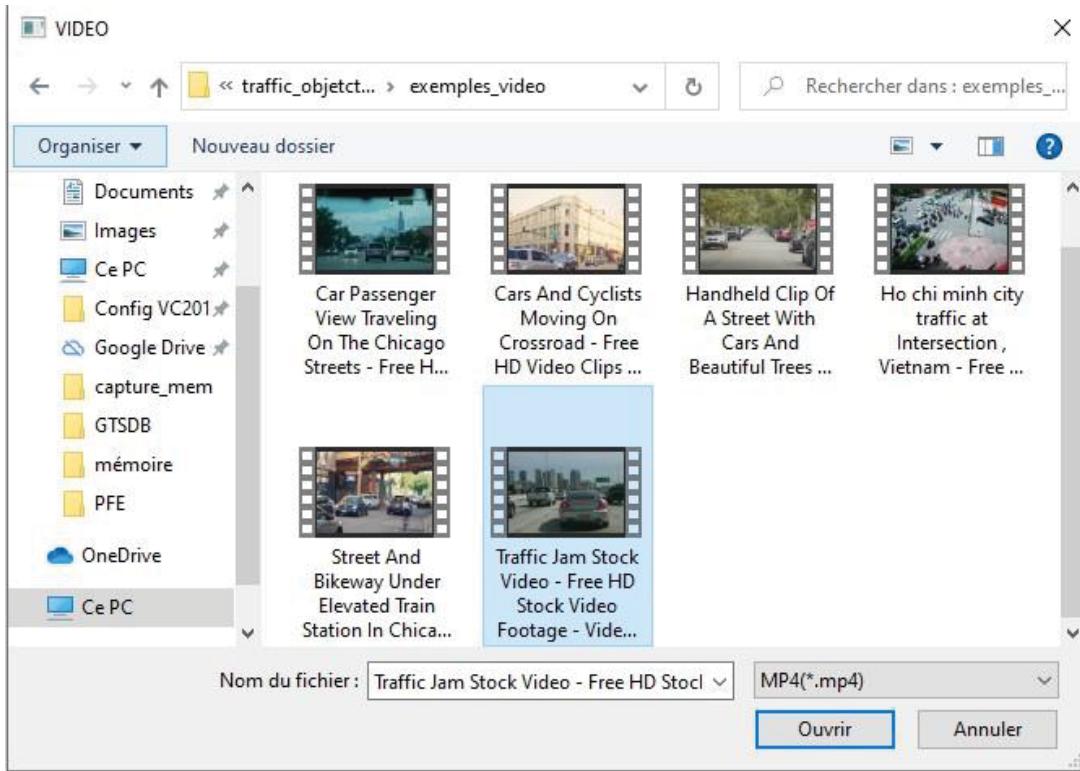


**Figure 4. 8.** Le choix d'une image.

Ensuite, le système traite l'image (il applique un changement sur la taille de l'image pour la transformer en 300\*300, pour qu'on puisse utiliser le modèle SSD300 obtenu et la méthode de suppression non maximale pour localiser les objets de la circulation routière dans l'image (il dessine une boîte autour de l'objet) et reconnaître sa catégorie.

- **La détection des objets dans un enregistrement vidéo**

Pour que le système puisse détecter des objets dans une vidéo, il faut appuyer sur le bouton « importe une vidéo » pour choisir une vidéo comme il est montré ci-dessous. Puis, le système va extraire les images et changer leurs tailles en 300\*300.



**Figure 4. 9.** Le choix d'un enregistrement vidéo.

- **La détection des objets en temps réel**

Pour détecter des objets en temps réel, on appuie sur le bouton « vidéo en direct ». Le système utilise les vidéos capturées par une caméra (webcam), et il applique les mêmes tâches qu'on a appliquée sur un enregistrement vidéo.

Dans le cas où le système détecte un feu de la circulation ou bien un panneau stop dans une vidéo (un enregistrement vidéo ou une vidéo en temps réel), il dessine une boîte rouge autour de ces objets, et lance un signal sonore.

#### 4.3.2. Le développement de la partie d'estimation entre la caméra et l'objet détecté

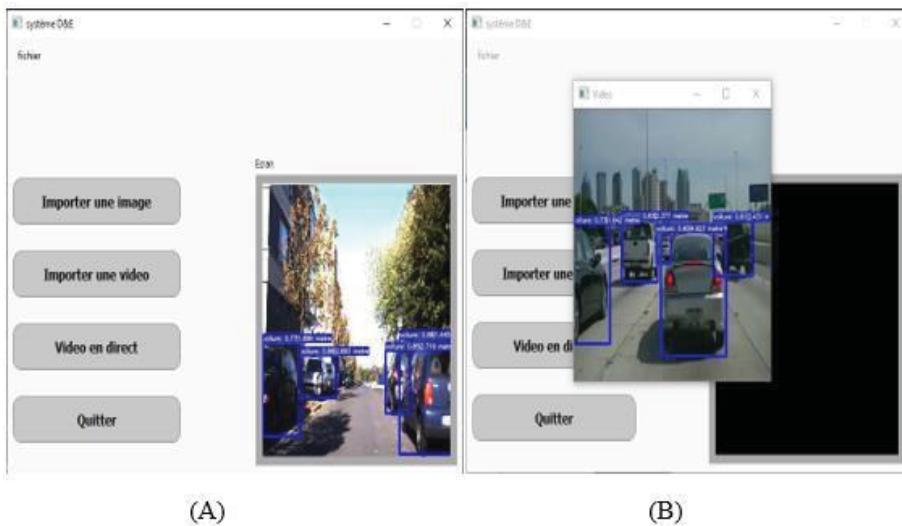
Dans cette partie, le système utilise la fonction ci-dessous qui permet d'estimer la distance entre la caméra et l'objets détecté (plus précisément la boîte de délimitation qui entoure l'objet). Cette fonction récupère les coordonnées prédites par le modèle SSD300 et la méthode de suppression non maximale pour calculer les valeurs de hauteur et de largeur de la boîte, ensuite il utilise la formule que nous avons expliquée dans le chapitre précédent pour estimer la distance.

```
def dist_calculator(startX,startY,endX,endY):  
    box_width=int (endX-startX)  
    box_height=int (endY-startY)  
  
    distance = (2*3.14*180)/((box_width+box_height)*360)*1000+3
```

**Figure 4. 10.** La fonction d'estimation de la distance.

#### 4.4. Les résultats obtenus et la discussion

Notre système affiche les images traitées dans la partie « écran » comme le montre la figure 4.11.a, où il affiche les vidéos traitées (un enregistrement vidéo ou une vidéo capturée par un webcam) dans une fenêtre comme le montre la figure 4.11.b.



**Figure 4. 11.** L'affichage de notre système.

On distingue deux situations, elles sont décrites dans ce qui suit.

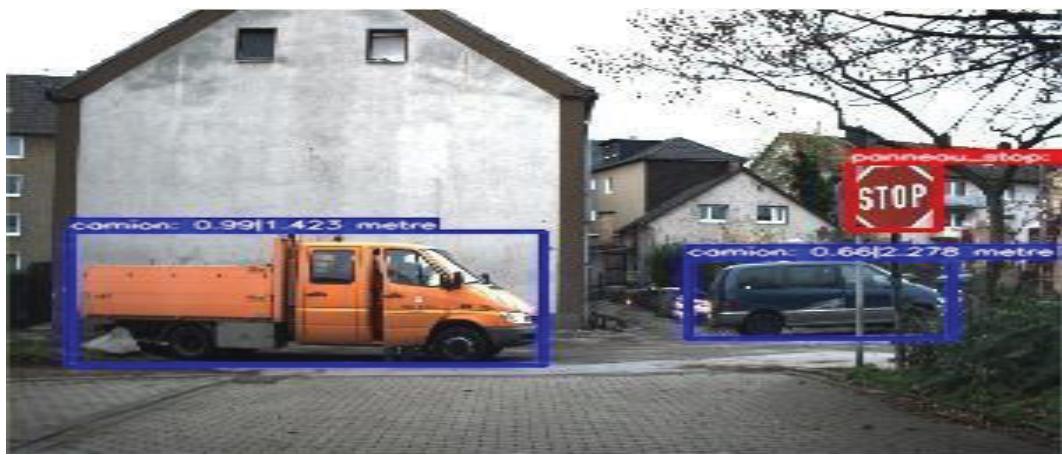
## La première situation



**Figure 4.12.** Détection des objets et estimation de distance dans la première situation.

Notre système a localisé les feux de la circulation et les panneaux STOP et les repère par une boite rouge, les autres objets par une boite bleue (voir les figures 4.12 et 4.13), et lance aussi un signal sonore pour avertir le conducteur. Nous pouvons voir dans la figure 4.12 que notre système a fait la reconnaissance de 5 objets parmi 8, mais il n'a pas pu détecter les petits objets (car il utilise le modèle SSD300 et ce modèle n'est pas optimisé pour les petits objets). Pour l'estimation de distance, nous avons trouvé que le système a un taux d'erreur entre 0.3-0.6 mètre, car la fonction utilisée par le système est basée sur la largeur et la hauteur de la boite pour estimer la distance.

### La deuxième situation



**Figure 4.13.** La détection des objets dans la deuxième situation.

D'autres résultats sont présentés dans la figure 4.13 ci-dessous. Nous remarquons que le modèle a fait une confusion entre le camion et la voiture, car ils ont une forme similaire. Donc nous constatons que notre système est moins robuste quand les objets qu'ont les mêmes formes.

Suite à un ensemble d'expériences sur notre système, nous observons que :

- Le SSD fonctionne moins bien pour les objets à petite échelle.
- L'apprentissage par transfert prend un peu de temps pour créer un modèle SSD par contre l'apprentissage par entraînement prend beaucoup de temps car l'optimisation des hyperparamètres est difficile, nous pouvons connaître l'effet d'un paramètre après avoir vu l'évolution des métriques d'entraînement pour choisir les valeurs optimales.
- Les cartes d'entités multi-échelles améliorent la détection d'objets à différents niveaux.
- L'utilisation de la caméra monoculaire et la bibliothèque OpenCV pour estimer la distance n'est pas toujours la solution idéale, elle ne donne pas des valeurs exactes.
- À cause de limitations de notre matériel, le système affiche les vidéos en faible fps (en anglais frame per seconde).

## 4.1 Conclusion

Dans ce chapitre, nous avons montré l'implémentation détaillée d'un système de détection des objets de la circulation routière et d'estimation de la distance entre la caméra et l'objet détecté, nous avons aussi montré les limites des algorithmes utilisés, ainsi que les résultats obtenus par notre système.

## Conclusion générale

L'objectif principal de ce mémoire est de proposer un système de perception qui peut être intégré dans les systèmes d'aide à la conduite. Ce système permet de détecter les objets qui entourent un véhicule à partir d'images fournies par une caméra embarquée, et aussi d'estimer la distance de l'objet en question par rapport à la caméra.

Nous avons décrit dans ce mémoire un algorithme de détection d'objets performant et rapide en utilisant des technologies d'apprentissage profond (Deep Learning, CNN), et nous avons proposé un algorithme pour l'estimation de la distance entre la caméra et l'objet détecté, dans le but d'exploiter ces deux techniques dans des systèmes d'aide à la conduite.

Le système de perception est principalement basé sur deux algorithmes.

- Le premier permet de détecter les objets de la circulation routière tels que les voitures, les camions, les feux de la circulation, les piétons, les panneaux STOP, les bicyclettes, les motocycles et les bus.
- Le deuxième permet d'estimer la distance entre la caméra intégrée et l'objet détecté.

La coopération de ces deux algorithmes permet de détecter les objets présents autour du véhicule et d'estimer sa distance par rapport à la caméra embarquée. Les tests effectués sur plusieurs vidéos et images ont montré que les résultats de ce programme sont satisfaisants pour la détection de plusieurs objets et pour l'estimation de la distance (voitures, camions, etc.).

Néanmoins, ces approches ne permettent pas encore d'apporter une solution unifiée pour répondre aux nombreuses difficultés de la détection : temps de calcul, détection des petits objets, calcul exact de la distance, etc.

En ce qui concerne nos prochaines perspectives de travail, l'objectif est d'améliorer l'ensemble du système de détection et d'estimation de distance en prenant en compte les points suivants.

- Trouver une solution qui permet à notre système de détecter les objets de petite taille.

- Identifier l'état du feu du circulation (lumière rouge ou verte), et avertir le conducteur.
- Utiliser une caméra stéréoscopie pour estimer la distance entre les caméras et l'objet détecté.
- Améliorer l'estimation de la distance pour permettre au système de proposer au conducteur une vitesse sécurisée dont le but de réduire la gravité de la collision avec les objets en faces en cas d'accident.
- Identifier des autres panneaux de signalisation.

## Bibliographie

1. «Introduction To Deep Learning» .Disponible sur : <https://medium.com/tebs-lab/introduction-to-deep-learning>. Publié le le 13 juin 2018, consultée le 1 juin 2021.
2. C. Crouspeyre, Comment les Réseaux de neurones à convolution fonctionnent. Disponible sur : <https://medium.com/@CharlesCrouspeyre/comment-les-réseaux-de-neurones-à-convolutionfonctionnent-b288519dbcf8>, publié le 17 juillet 2017, consultée le 23 mars 2021.
3. Deepanshu Tyagi, « Introduction To Feature Detection And Matching. ». Disponible sur:<https://medium.com/data-breach/introduction-to-feature-detection-and-matching>. Publiée le 3 Janvier 2019, consultée le 15 mars 2021.
4. Deepanshu Tyagi, « Introduction to SIFT( Scale Invariant Feature Transform). » , page web, <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform>, publiée le 16 Mars 2019, consultée le 15 mars 2021.
5. « What are Haar Features used in Face Detection. » ,Disponible sur: <https://medium.com/analytics-vidhya/what-is-haar-features-used-in-face-detection>, publiée le 12 novembre 2019, consultée le 17 mars 2021.
6. Jose Luis Masache Narvaez. « Adaptation of a Deep Learning Algorithm for Traffic Sign Detection ». Mémoire de master. Canada : l'université du Western Ontario, 2019.
7. Kang S., Byun H., Lee SW. « Real-Time Pedestrian Detection Using Support Vector Machines. ».2002. vol 2388.
8. Richa Agrawal , « Person Detection in Various Posture using HOG Feature And SVM Classifier. » ,Disponible sur : <https://medium.com/@richa.agrawal228/person-detection-in-variousposture-using-hog-feature-and-svm-classifier>, publié le 31 Octobre 2018, consultée le 20 mars 2021.
9. Lindgren Walter, et al. « Requirements for the Design of Advanced Driver Assistance Systems - The Differences between Swedish and Chinese Drivers ».2008.International Journal of Design. 2. pp. 41-54.
10. Kukkala, Vipin Kumar, et al. «Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles. ».25 septembre 2018. IEEE Consumer Electronics Magazine, vol. 7, no. 5, pp. 18-26.
11. R,Grossman .« Jeu de données thermiques FLIR GRATUIT pour l'entraînement des algorithmes». Disponible sur: <https://www.flir.com/fr/oem/adas/adas-dataset-form/>, publiée 19 septembre 2019. Consultée le 1 avril 2021.
12. Jon L. Grossman. « Thermal Infrared vs. Active Infrared: A New Technology Begins to be Commercialized ». Disponible sur : <https://irinfo.org/03-01-2007-grossman/>.publiée 4 février 2019. consultée le 05 avril 2021.

13. Kim, Giseok, and Jae-Soo Cho. « Vision-Based Vehicle Detection and Inter-Vehicle Distance Estimation for Driver Alarm System. ». Novembre 2012. Optical Review, vol. 19, no. 6, pp. 388–393.
14. J. Hui, « SSD object detection: Single Shot MultiBox Detector for real-time processing », Disponible sur: [https://medium.com/@jonathan\\_hui/ssd-object-detection-single-shotmultiboxdetector-for-real-time-processing-9bd8deac0e06](https://medium.com/@jonathan_hui/ssd-object-detection-single-shotmultiboxdetector-for-real-time-processing-9bd8deac0e06). Publiée le 14 mars 2018, consultées le 21 septembre 2021
15. Zeiler, Matthew D. et Rob Fergus. "Visualiser et comprendre les réseaux convolutifs." Dans Conférence européenne sur la vision par ordinateur, ,2014, pp. 818-833.
16. Eddie Forson, « Understanding SSD MultiBox—Real-Time Object Detection In Deep Learning », Disponible sur: <https://towardsdatascience.com/understanding-ssd-multibox-real-timeobjectdetection-in-deep-learning-495ef744fab>. Publiée le 18 octobre 2017, consultée le 1 aout 2021.
17. Paul pias, « Object detection and distance measurement», Disponible sur : <https://github.com/paul-pias/Object-Detection-and-Distance-Measurement>. Publiée en 28 juillet 2020, consultées en 14 septembre 2021.
18. Pierluigiferrari, Disponible sur : <https://drive.google.com/file/>. Publiée le 7 mai 2020, consultée le 5 septembre 2021.