REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Université Mohamed Khider-BIKSRA
Faculté des Science Exactes et des Sciences de la Nature et la Vie

-Département de l'informatique-

Mémoire

Présenté en vue de l'obtention du diplôme de

Master

INFORMATIQUE

THEME :

Un système de détection des objets de la circulation routière et d'estimation de leur distance.

Spécialité : Image et Vie Artificielle

Par : **Djenaihi Lamine**

Encadreur : **Dr. Mokhtari Bilal**

Devant les jurys :

Présidant :

Examinateur :

Année Académique 2019-2020

Remerciement

Tout d’abord, je tiens à remercier Allah de m'avoir donné la possibilité de réaliser ce projet.

Je tiens aussi à remercier mon superviseur monsieur Mokhtari Bilal, j’ai eu l'occasion de profiter à la fois de sa compétence scientifique et de sa grande disponibilité, et de m'avoir aidé pour résoudre les difficultés rencontrées lors de ce projet, et d'avoir répondu à toutes mes questions.

Je m'exprime ma gratitude à tous les professeurs du département de l’informatique.

Je tiens à remercier sincèrement mes proches pour leur soutien constant tout au long de ma carrière universitaire.

Dédicace

Je dédié ce travail à Ma mère bien-aimée, ma première enseignante qui a toujours eu confiance en moi, à mon père respecté dont le soutien m'a aidé à atteindre mes objectifs, à ma sœur Djihane et Mon frère Houssem pour leurs encouragements.

Je le dédier aussi à mes meilleurs amis et collègues Larbi, Hamza, Hichem, Ziad, Aymen, et à toutes les personnes qui m’ont encourager pour terminer ce travail.

Liste des abréviations

CNN	Convolutional Neural Network
DPI	Dots Per Inch
PPP	Point Par Pouce
cm	centimètre
mm	millimètre
PCX	PiCture eXchange
GIF	Graphic Interchange Format
JPEG	Joint Photographique Experts Group
DXF	Data eXchange Format
CGM	Computer Graphics Metafile
bit	Binary digIT
RNA	Réseau de Neurones Artificielle
CONV	La couche de Convolution
ReLU	Rectified Linear Unit
POOL	Pooling
FC	Fully Connected
ZFNet	Zeiler & Fergus Neural Network
LeNet	Lecun Neural Neural Network
VGGNet	Visual Geometry Group Neural Network
AlexNet	Alex Neural Network
ResNet	Residual Neural Network
HOG	Histogram of Oriented Gradients
SIFT	Scale-Invariant Feature Transform

SURF	Speeded Up Robust Features
SVM	Support Vector Machine
Faster R-CNN	Faster Region Convolutional Neural Network
Mask R-CNN	Mask Region Convolutional Neural Network
YOLO	You Only Look Once
SSD	Single Shot Detection
PASCAL VOC	PASCAL Visual Object Commun
COCO	Common Object in Context
IRM	Imagerie par Résonance Magnétique
DGSN	Direction Générale de la Sûreté Nationale
ADAS	Advenced Driver Assisstant System
PMD	Photonic Mixer Device
IMU	Inertial Measurement Units
GPS	Global Positioning Systems
LDW	Lane Departure Warning
ACC	Adaptative Cruise Control
LIDAR	Laser Detection And Ranging
RADAR	Radio Detection And Ranging
RVB	Rouge, Vert, Bleu
HSV	Hue, Satuaration, Value
FCW	Forward Collision Warning
AEB	Auto Brake Emergenc

Listes des figures

Tableau de Matière

Introduction générale

Les développements technologiques de ces vingt dernières années ont favorisé un accès aux systèmes numériques de notre vie quotidienne. Parmi les composantes majeures des systèmes numériques, une grande importance est accordée à l'image. La représentation et le traitement des images numériques sont actuellement l'objet de recherches très actives. Le traitement des images est un domaine très vaste qui a connu un développement important depuis quelques décennies.

L'un des types de l'intelligence artificielle les plus puissants et les plus convaincants est la vision par ordinateur. La vision par ordinateur est le domaine de l'informatique qui se concentre sur la reproduction de certaines parties de la complexité du système de vision humain et qui permet aux ordinateurs d'identifier et de traiter les objets dans les images et les vidéos de la même manière que les humains le font. Jusqu'à récemment, la vision par ordinateur ne fonctionnait que de manière limitée. Grâce aux progrès de l'intelligence artificielle et aux innovations en matière d'apprentissage profond, le domaine a pu faire de grands bonds ces dernières années et a pu surpasser les humains dans certaines tâches liées à la détection et à la reconnaissance des objets.

L'industrie automobile connaît l'un des changements les plus fondamentaux de son histoire. Les systèmes avancés d'aide à la conduite et les véhicules autonomes sont l'une des technologies innovantes qui sont à l'origine de ce changement. Dans un véhicule, les systèmes avancés d'aide à la conduite sont un ensemble de systèmes comprenant des capteurs tels que l'image, le Lidar, le radar et des processeurs informatiques pour offrir aux utilisateurs une expérience de conduite intelligente, sécurisée et confortable. La plupart des accidents sont dus à une erreur humaine, c'est pourquoi les systèmes avancés d'aide à la conduite aident à adapter, automatiser et améliorer les véhicules pour la sécurité et une meilleure conduite. Ils sont conçus pour alerter le conducteur d'un danger potentiel et l'aider à garder le contrôle de son véhicule afin d'éviter les accidents et d'en réduire la gravité si possible.

Dans notre travail, nous avons développé un programme informatique basé sur des techniques de vision par ordinateur et de traitement d'image. Ce programme permet la détection des différents objets du circulation routière (par exemple voiture, piéton, camion, etc.). Il permet

par ailleurs de localiser l'objet et reconnaitre sa catégorie et d'estimer la distance entre la caméra et cet objet. Pour ce faire nous avons fait appel à un capteur (une caméra) monté sur le tableau de bord d'un véhicule. Les images ainsi produites par la caméra sont traitées par notre programme qui génère des données (coordonnées, distance, messages d'alarmes sur la nature de l'obstacle, etc.) qui peuvent être exploités par un système embarqué sur le véhicule.

Le présent mémoire s'articule autour de quatre chapitres. Le chapitre 1 traite essentiellement deux parties. Dans la première, nous présentons la notion et les techniques principales de traitement d'image et principalement l'acquisition et la segmentation. La deuxième partie est consacrée à la vision par ordinateur et l'apprentissage a profonds. Dans le chapitre 2, nous présentons la notion de système d'aide à la conduite et les capteurs utilisée dans ce système, et nous détaillerons les systèmes d'aide à la conduite basée sur la vision (les capteurs du vision), et nous identifions les avantages et les défis majeurs influençant ces systèmes. Dans le chapitre 3, nous présentons une méthode qui permet d'identifier automatiquement les objets (véhicules, piétons, etc.), elle est basée sur des techniques de vision par ordinateur (haut niveau) et les architectures d'apprentissage a profonds, ainsi qu'une méthode qui permet d'estimer la distance entre une caméra et l'objet détecté. Dans le chapitreIV, nous présentons le développement de notre système et résultats obtenus lors de son application sur des images, des enregistrements vidéo, et des vidéos prises par une webcam. Nous présentons également les points forts et les points faibles de notre système. Nous terminerons par une conclusion générale.

Chapitre 1

Le traitement d’image et

la vision par ordinateur

CHAPITRE 1 : Le traitement d'image et la vision par ordinateur

1.1. Introduction

Le traitement d'image est un domaine très large qui a connu et qui a assisté un développement important depuis quelques dizaines d'années. Il représente l'ensemble des techniques qui sont appliqués à l'image numérique pour l'améliorer ou extraire des informations. La disponibilité de l'image et les vidéos (on trouve par exemple sur Facebook, Instagram, Google, etc.) aide au développement de la vision par ordinateur, cette technologie permet aux ordinateurs de comprendre et d'analyser le monde réel. Les pays développés investissent des milliards de dollars pour développer cette technologie, car le taux d'erreurs est moins et le gain est plus haut que l'utilisation des travailleurs humains. Elle peut être utilisée dans plusieurs domaines (La santé, l'agriculture, les véhicules autonomes, etc.).

Dans la première partie de ce chapitre, nous allons définir l'image numérique, ces types, ces caractéristiques, puis les principales techniques de traitement d'images.

Dans la deuxième partie, nous allons définir la vision par ordinateur, puis l'apprentissage automatique, et l'apprentissage approfondis. Dans l'apprentissage approfondis, nous décrivons essentiellement les réseaux de neurone convolutif CNN. Ensuite, nous allons définir la classification des images, et la détection des objets.

Dans la troisième partie, nous allons décrire les bases de données le plus utilisées comme une référence dans la recherche et le développement des techniques de la vision par ordinateur.

La dernière partie montre quelques domaines qui utilisent les techniques de traitement d'images et la technologie de la vision par ordinateur.

1.2. Le traitement d'images

1.2.1. Définition de l'image numérique

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique 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 : coordonnées cartésiennes d'un point de l'image. f(x, y) : niveau d'intensité.

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

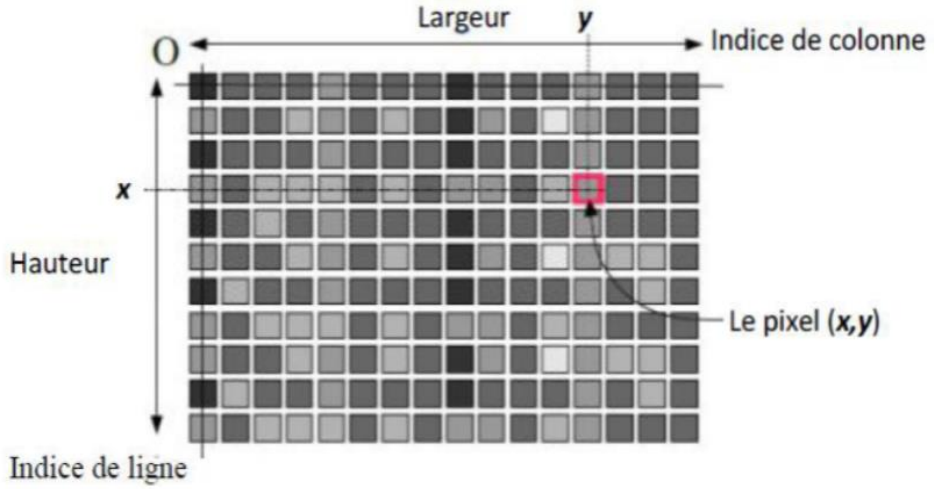

Figure 1. 1. Représentation d'image numérique [1].

1.2.2. Les caractéristiques des images numériques

1.2.2.1. Le pixel

Une image numérique constitue d'un ensemble de points appelés pixels pour former une image. Le pixel représente ainsi le plus petit élément constitutif d'image numérique, et chaque pixel contient une couleur. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image [1].

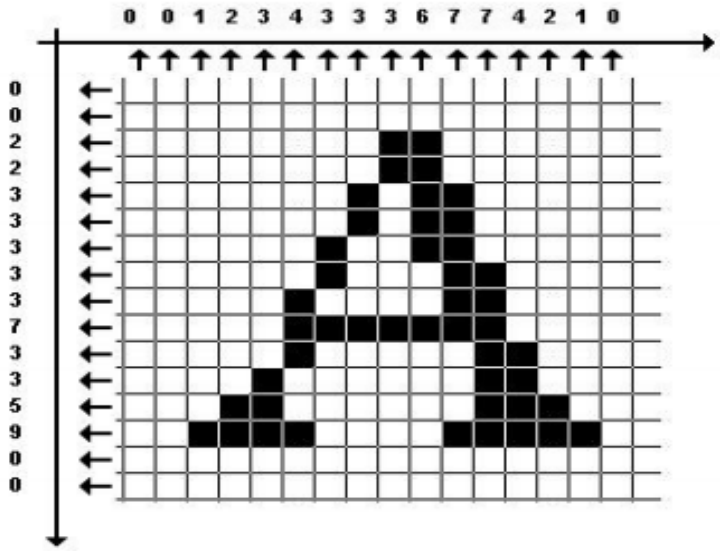

Figure 1. 2. Le groupe de pixels forme la lettre A [2].

1.2.2.2. La résolution

La résolution est le nombre de pixel par unité de surface, elle s'exprime plus souvent en Points Par Pouce (PPP, en anglais DPI pour Dots Per Inch), un pouce représentant 2.54 cm [3]. La résolution définit la précision et la qualité d'une image. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une surface de 1 pouce), plus votre image est précise dans les détails.

Remarque [3] :

1 pouce = 2,54 cm.

1 pouce = 25,40 mm = 100 pixels.

1 inch = 2,54 cm = 1 pouce.

1.2.2.3. La dimension

La dimension est la longueur et la hauteur d'une image numérique, ces mesures est en pixels. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuse(pixels). Le nombre de lignes de cette matrice multipliée par le nombre de colonnes nous donne le nombre total de pixels dans une image [2].

1.2.2.4. La profondeur

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

- 32 bits/pixel = 1,07 milliards 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

Le poids d'une image se détermine en fonction de ces deux paramètres : dimensions, profondeur. Le poids de l'image est alors égal à sa dimension multipliée par sa profondeur. Par exemple, pour une image 640x480 en vraies couleurs (True colors) :

- Nombre de pixels (dimension) : 640 x 480 = 307200
- Poids de chaque pixel (profondeur) : 24 bits = 3 octets

- Le poids de l'image est ainsi égal à : 307200 x 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 (parasite) dans une image est considéré comme 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 est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

Des images lumineuses (brillantes).

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].

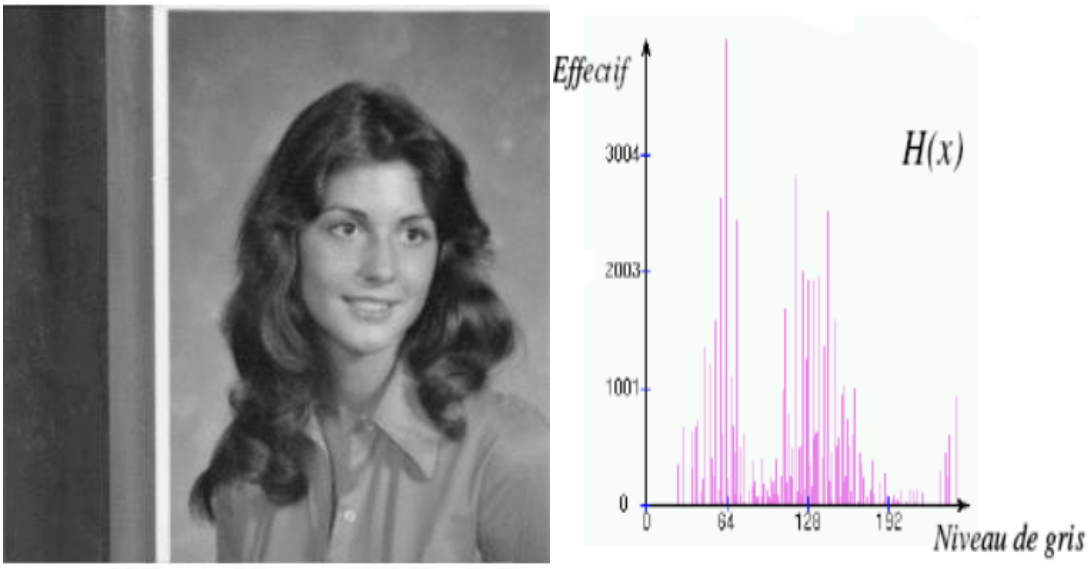

Figure 1. 3. Représentation d'un histogramme d'une image sous Matlab Avec H(x) est le nombre de pixels dont le niveau de gris est égal à x.

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 porte 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 : BMP (Windows Bitmap), PCX (PiCture eXchange), GIF (Graphic Interchange Format), JPG ou JPEG (Joint Photographique Experts 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 des formules mathématiques. Cela permet alors d'agrandir l'image indéfiniment sans perte de

qualité et d'obtenir un faible encombrement. Des format standards des images vectorielle : DXF (Data eXchange Format), CGM (Computer Graphics Metafile),…etc [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 x 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 ne niveaux de gris autorise 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 (ou plus) images en niveaux de gris sur trois (ou plus) composantes. 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 (ou plus) composantes[2].

1.2.4.4. L'image à valeurs réelles

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 résultat comme des images à valeurs réelles. En général, une image à valeurs 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

Pour pouvoir 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 photos numériques. En médecine, on utilise des imageurs 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].

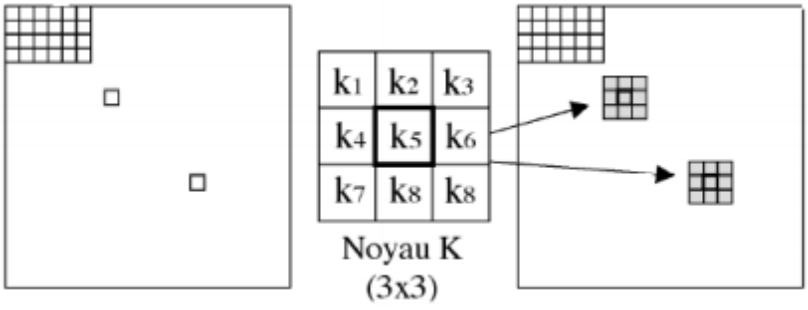

Figure 1. 4. Filtrage d'une image par un noyau de convolution [11].

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 :

$$S[i,j] = Somme_{u,v} (E[i,j] * h[i-u,j-v])$$

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].

h[0,0]	h[0,1]	h[0,2]
h[1,0]	h[1,1]	h[1,2]
h[2,0]	h[2,1]	h[2,2]

Les figures ci-dessous représente les résultat d'un filtrage linéaire tel que : la figure I.5.a représente l'image originale et la figure I.5.b représente l'application de filtre moyenneur 7*7 et la figure 1.5.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 applique le filtrage non linéaire comme : le filtre médian et le filtres de nagao. Dans la figure 1.6.b, ils ont appliqués un filtre median de taille 7*7, et dans la figure 1.6.c, ils ont appliqués un filtre nagao de taille 9*9 sur une image de niveau de gris.

(a) (b) (c)

Figure 1. 6. L'application des filtrages non linéaire [5].

I.2.5.3. La segmentation

On utilise la segmentation pour obtenir une partition de l'image en ses différentes régions d'intérêt. La segmentation est un traitement qui consiste à créer une partition de l'image considérée, en sous-ensemble 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 pixels, segmentation basée régions et segmentation basée contours [2].

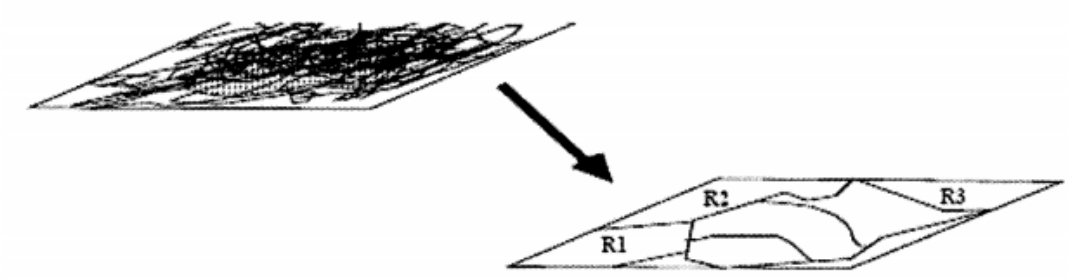

Figure 1. 7. La segmentation d'une image [6].

I.2.5.3.1. Segmentation basée sur les pixels

Le principe consiste à 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

utilise cette technique comme les méthodes de seuillage et les méthodes de classification (clustering). Figure 2.6 un exemple montre les résultats de segmentation basée sur les pixels.

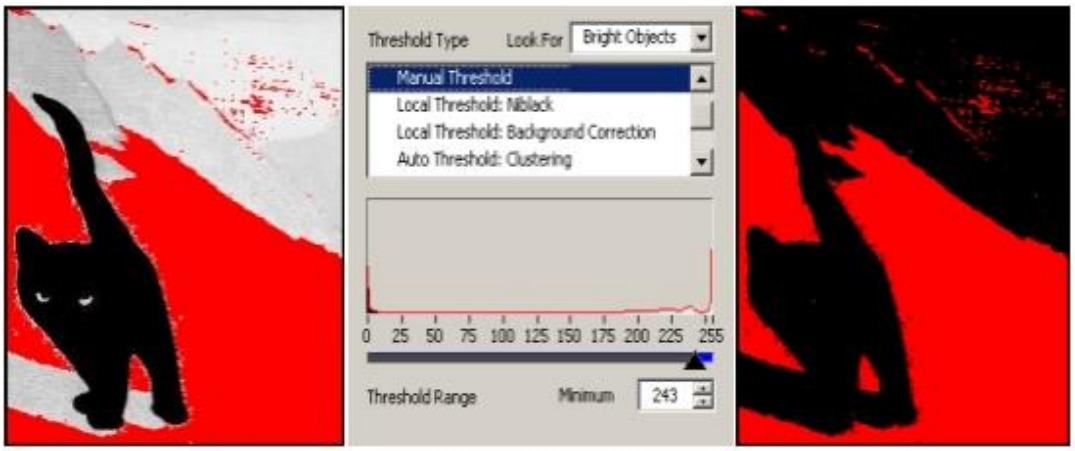

Figure 1. 8. La segmentation basée sur les pixels [2].

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 région :

Les méthodes de croissance de régions qui agrègent les pixels voisins (méthodes ascendantes) selon le critère d'homogénéité (intensité, 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 les contours

Leur principe s'intéresse aux contours 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 découpé 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. Il 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'objectifs de la vision par ordinateur est de mettre l'ordinateur de comprendre ce qui est vu, et extraire des informations complexes sous une forme qui peut être utilisée dans d'autres processus. Les spécialités les plus connus et les plus utilisé dans ce domaine c'est **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'étude 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 peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée (prédiction) [8]. Dans le domaine de

classification des images, il utilise les résultats de descripteur caractéristique 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 photographie).

1.3.3. L'apprentissage profond

L'apprentissage profond (en anglais « Deep Learning ») est un ensemble des 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].

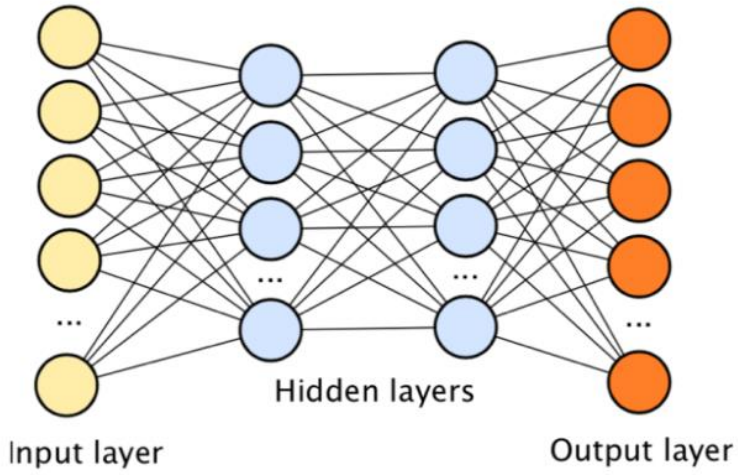

Figure 1. 10. Illustration montre l'architecture de *Deep Learning [10]*.

Il existe différents algorithmes de Deep Learning, Nous pouvons ainsi citer :

- 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 le plus utilisée dans les domaines de classification de les 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 partie d'un CNN est la partie convolutive. 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.

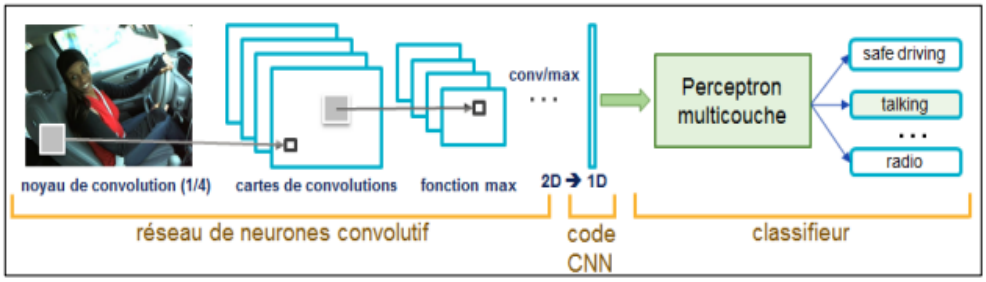

Figure 1. 11. Architecture et composition d'un réseau des neurones convolutifs [7].

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 Convolutifs

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 le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
- La marge (à 0) ou « zero 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).

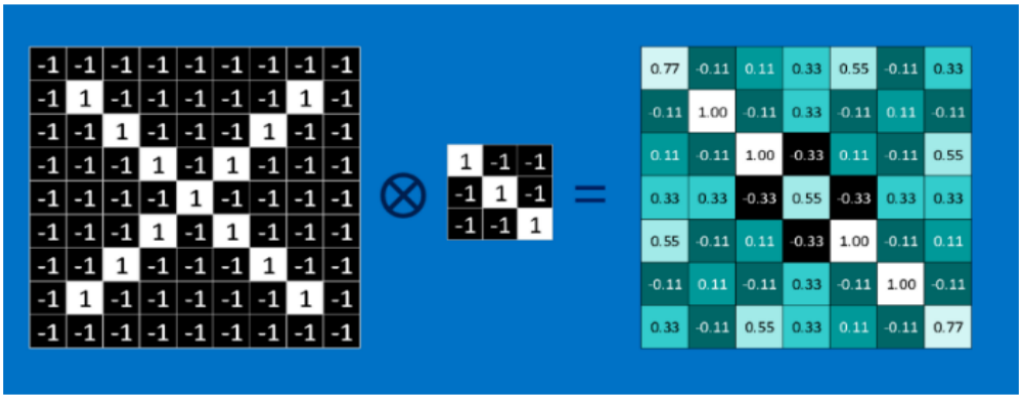

Figure 1. 12. Illustration de l'opération de convolution entre une image et un filtre [13].

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

$$F(x) = max(0, x) \qquad (1)$$

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 sur-apprentissage.

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 la valeur maximale en entrée (Voir figure 1.13). On parle dans ce cas de « Max-Pool 2x2 ».

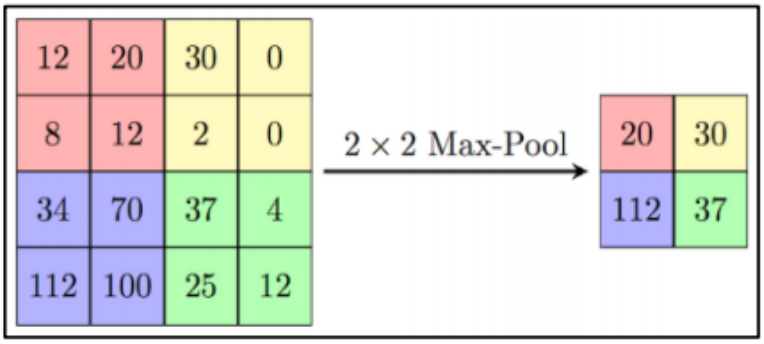

Figure 1. 13. Illustration de l'opération de pooling « POOL 2×2 » [15].

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'entrainement 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 [9][15] :

1.3.3.1.2.1. LeNet

LeNet était le réseau neuronal convolutif le plus 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 célèbre est AlexNet, qui popularise le réseau neuronal convolutif en vision par ordinateur,il était développé 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 les gagnants 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 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 était développée par Szegedy et al, de Google 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 compétent 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 la gagnante 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 faire 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 prédite. 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].

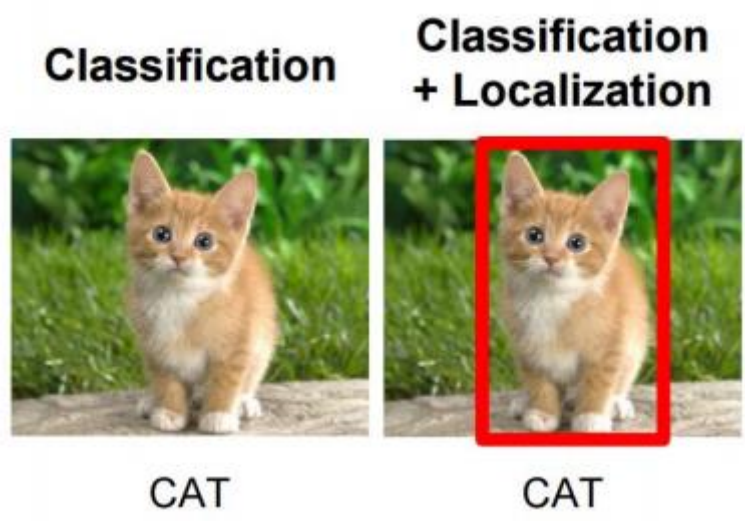

Figure 1. 14 Illustration de la différence entre la classification et la détection [7].

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 telles 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 sont :

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, il 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. Il 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. Les caractéristiques de Transformation d'une caractéristique invariante à l'échelle

Transformation d'une caractéristique invariante à l'échelle (en anglais « Scale-Invariant Feature Transform (SIFT feature)») et a été présenté 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, il est utilisé pour suivre des images, détecter et identifier des objets (qui peuvent aussi être partiellement cachés)[18].

1.3.5.2.3. Les caractéristiques de robuste accélérée

Une caractéristique robuste accélérée (en anglais « Speeded Up Robust Features »(SURF features)) est un algorithme de déscripteur de caractéristique et un descripteur, présenté par des chercheurs de l'école polytechnique fédérale de Zurich et de Université catholique de Louvain pour la première fois en 2006 puis dans une version révisée en 2008. Il est utilisé dans le domaine de vision par ordinateur, pour des tâches de détection d'objet ou de reconstruction 3D. Il est partiellement inspiré par le descripteur SIFT, qu'il surpasse en rapidité et, selon ses auteurs, plus robuste pour différentes transformations d'images[19].

1.3.5.2.4. Les caractéristiques de pseudo-Haar

Les caractéristiques pseudo-Haar (en anglais « Haar-like features ») sont décrites pour la première fois dans un article de Paul Viola et Michael Jones paru en 2001 dans la revue scientifique (International Journal of Computer Vision (IJCV)), dans laquelle ils décrivent une nouvelle méthode de détection de visage. 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. des 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 scannent l'image entière en appliquant des fenêtres coulissantes de différentes tailles et échelles et en générant des recadrages d'image plus petits qui sont ensuite analysés individuellement pour déterminer s'il y a un objet à l'intérieur de la fenêtre coulissante. En raison du nombre important de candidats analysés, ce processus est coûteux en termes de calcul [21].

1.3.5.3.1.2. L'extraction des caractéristiques

Pour analyser chaque recadrage d'image généré au cours du processus des fenêtres coulissantes, nous avons besoin de caractéristiques visuelles qui nous donnent des informations significatives sur l'image. Comme un exemple les caractéristiques d'HOG utilisées dans la détection humaine et les caractéristiques Haar-like utilisées dans la reconnaissance des visages. Cependant, la plupart des descripteurs de caractéristiques sont conçus pour détecter un type spécifique des objets, et leurs performances peuvent être affectées par les conditions d'éclairage[21].

1.3.5.3.1.3. La classification

Une fois que nous avons le vecteur descripteur de caractéristique de chaque fenêtre coulissante, l'étape suivante consiste à classer les éléments de l'image dans une classe d'objets cibles 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 est proposée par les chercheurs Paul Viola et Michael Jones en 2001. Elle a été proposée au départ pour la détection de visages dans une image numérique ou séquence vidéo puis utilisée pour détecter d'autres objets comme les voitures ou les avions…, En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones

nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter (elle utilise les caractéristiques de pseudo-Haar), pour entraîner la classificateur AGHaar (un algorithme de l'apprentissage automatique). Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles [8].

1.3.5.3.2.2. La méthode de les séparateurs à vaste marge avec HOG

Cette méthode est plus utilisée dans le domaine de détection des véhicule et détection des piétons [22][23]. Il nécessite de préparer la base des images en une collection d'échantillons positifs et négatifs, Les échantillons positifs sont les objets d'intérêt. Et les échantillons négatifs sont les images qui n'ont pas d'objet d'intérêt. Le HOG est utilisé comme un vecteur de caractéristique pour entraîner le modèle de les séparateurs à vaste marge (SVM). Le descripteur HOG utilise une fenêtre de détection coulissante qui se déplace autour de l'image. À chaque position de la fenêtre du détecteur, Ce descripteur est calculé pour la fenêtre de détection. Il est ensuite montré au SVM formé, par exemple classe l'objet comme « personne » ou « ne pas un 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

Le structures de détecteur à deux étages consiste à suivre un processus de deux étapes. Tout d'abord, l'algorithme se concentre sur la génération d'une région d'intérêt (elle est une zone proposée à partir de l'image originale) ou de propositions, puis classe chaque région

D'intérêt dans des 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 de réseau convolutif régional plus rapide (en anglais « Faster Region Neural Network») [24] est de remplacer l'algorithme de propositions des régions d'intérêt par un réseau neurone (proposition des régions d'intérêt par un réseau neurone convolutif). Plus précisément, il a introduit le réseau de proposition de région (en anglais « Region Proposals Network »(RPN)). Une fois que nous avons nos propositions de région, nous les alimentons directement dans ce qui est essentiellement un R-CNN rapide. Nous ajoutons une couche de pooling, des couches entièrement connectées, et enfin une couche de classification softmax et

un régresseur de boîte de délimitation (voir figure 1.16). Dans un sens, Faster R-CNN = RPN + Fast R-CNN.

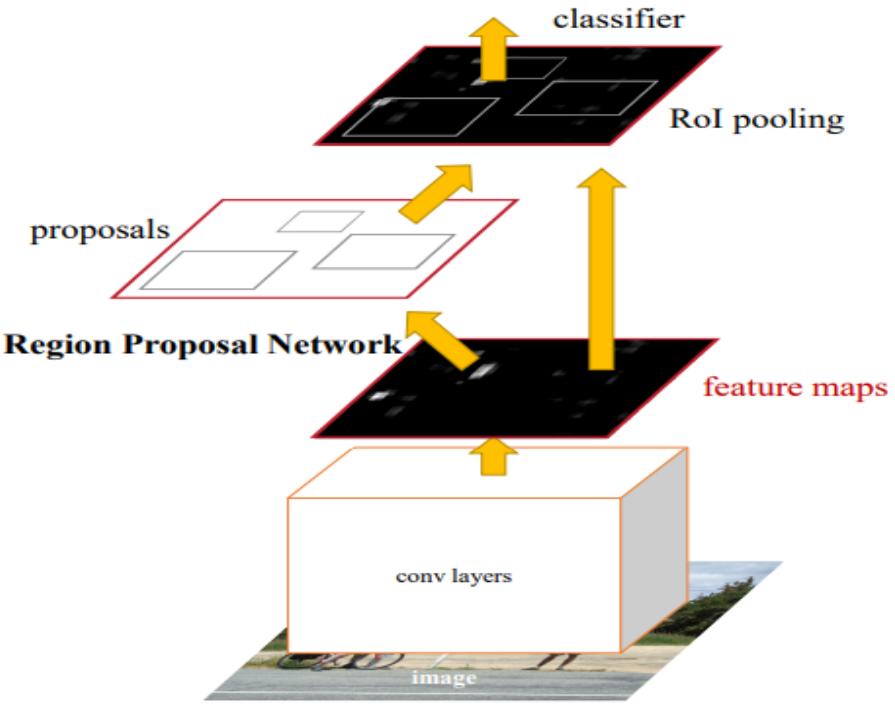

Figure 1. 15. Architecture de modèle Faster R-CNN[24].

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

Le masque R-CNN (en anglais « Mask Region Neural Network»)[25] est un détecteur à deux étages : la première étage scanne l'image et génère des propositions (zones susceptibles de contenir un objet). La deuxième étape classifie les propositions et génère des boîtes et des masques de délimitation. Les deux étapes sont reliées à la structure de base. La structure de base est un réseau neuronal convolutif standard (généralement, ResNet50 ou ResNet101) qui sert d'extracteur de caractéristiques. Les premiers couches détectent les éléments de bas niveau (bords et coins), et les couches suivantes détectent successivement les éléments de plus haut niveau (voiture, personne, ciel)(voir figure 1.16).

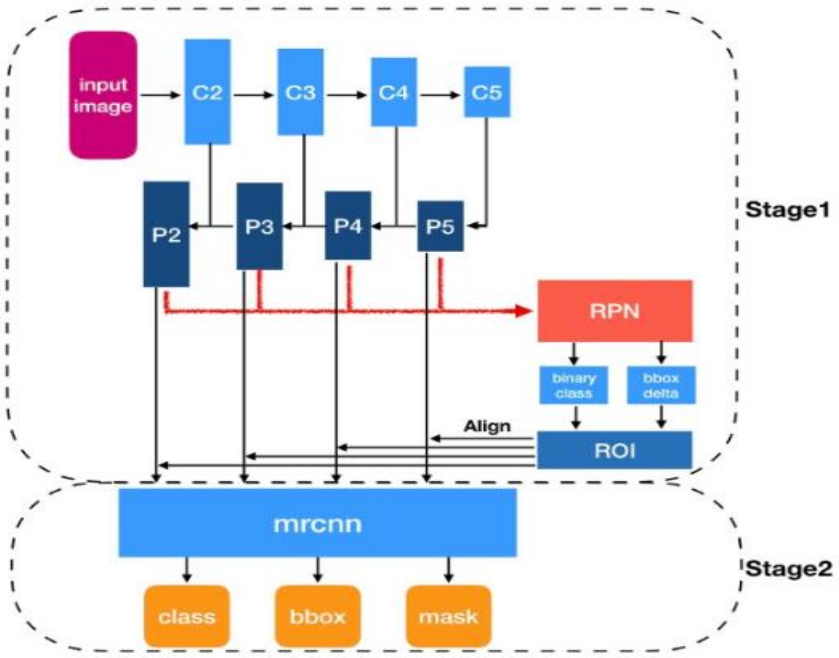

Figure 1. 16. L'illustration de l'architecture de Mask R-CNN [25].

1.3.5.4.2. Le détecteur à un étage

Le détecteur à un étage prédit directement la classe et l'emplacement de l'objet en utilisant un seul réseau neuronal convolutif (le réseau est capable de trouver tous les objets d'une image en un seul passage via le convnet) [21]. L'objectif principal des détecteurs à un étage est d'améliorer la vitesse de détection, cependant, 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] fait directement les prédictions des boites de délimitations 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.

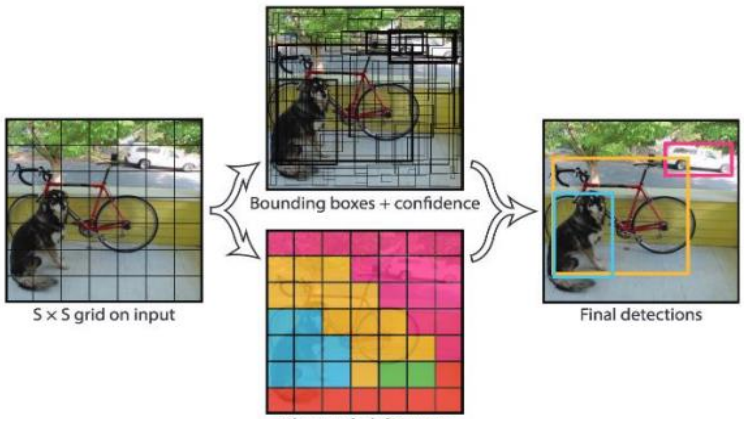

Figure 1. 17. Modèle YOLO[26].

Initialement, le modèle prend une image en entrée. Il le divise en une grille SxS. Chaque cellule de cette grille fait les prédictions d'un nombre B des zones de limitations avec un score de confiance. Ce score de confiance est simplement la multiplication de la probabilité de détecter une classe d'objet par l'indice d'IoU (l'abréviation de « Intersection overlapp Union ») entre les zones de limitations prédites (les boites de délimitation B utilisé dans chaque cellule) et les boîtes de vérité (étiquetés).

1.3.5.4.2.2. Le Détecteur mono-coup (SSD)

SSD (l'abréviation de « Single Shot Multibox Detection ») [27] est un algorithme de détection d'objets créé en Décembre 2016 par Wei Liu et son équipe de recherche.

Le nom de SSD signifie que les tâches de localisation et de classification des objets sont effectuées avec un seul passage dans le réseau, ceci est réalisé en utilisant une technique de régression Multi Box. (Multi Box est une méthode pour des propositions de coordonnées de boîte délimitation indépendantes des classes rapides) avec plusieurs boîtes de délimitation.

1.3.6. Les bases de données d'images utilisées

Les algorithmes de l'apprentissage automatique et l'apprentissage profond que nous avons définis, ils nécessitent une base d'images pour trouver le meilleur modèle qui permet de détecter des objets ou classifier des images avec un haut précision, les chercheurs dans les sociétés connus comme google, Facebook, Microsoft,…etc, ils sont créés des bases d'images et ils sont considérés comme des références pour faciliter à les chercheurs de l'IA de faire la recherche sans perdus le temps dans la collection d'images, ci-dessous nous avons cité quelques bases qui sont connus et gratuit dans la recherche du vision par ordinateur:

1.3.6.1. The PASCAL Visual Object Commun

PASCAL VOC (l'abréviation de « The PASCAL Visual Object Commun ») est une compétition, il a lancé en 2005, elle introduisait une base d'images qui a 20 classes. Dans la dernière compétition en 2012, la base d'images de l'entraînement et de validation était constituée de 27 450 objets de détection dans 11 530 images avec 20 classes différentes. Pour la segmentation, la base d'images de validation comprend 6929 objets segmentés dans 11 530 images [28].

1.3.6.2. ImageNet

La base d'images d'ImageNet est représentée par le concours ILSVRC(l'abréviation de « The ImageNet Large Scale Visual Recognition Challenge ») qui a débuté en 2010 . De nombreux modèles de classification et de détection d'objets ont été retenus à l'issue de ce concours, dont AlexNet. Pour la détection d'objets, ImageNet comprend 465 567 images pour la formation et 20 121 images pour la validation pour 200 classes ddifférentes. Elle est organisée de manière hiérarchique, selon WordNet, cela signifie que chaque nœud de la hiérarchie est représenté par des centaines et des milliers d'images[28].

1.3.6.3. Common Object in Context

COCO (l'abréviation de « Common Object in Context ») est une base d'image, elle introduisait par Microsoft en 2015, elle contient une segmentation d'instance de 80 catégories d'objets dans leur contexte naturel. Les étiquettes du COCO comprennent également des étiquettes et des points clé ont été ajoutés en 2016. L'ensemble de données COCO comprend 2,5 millions d'instances étiquetées dans 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 millions 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 connaissent une utilisation très répandus dans le monde entier, et un développement extraordinaire dans plusieurs domaines comme la militaire, la santé, l'industrie,… etc . Dans cette partie, nous avons parlé de quelques domaines pour prendre comme des exemples :

1.4.1. La militaire

Pour les armées modernes, la vision par ordinateur et le traitement d'images sont une technologie habilitants vitaux qu'aide les systèmes de sécurité à détecter les troupes ennemies ou les saboteurs et améliore les capacités de ciblage des systèmes de missiles guidés. Les

concepts militaires tels que la connaissance de la situation reposent largement sur les capteurs des images pour fournir des renseignements sur le champ de bataille utilisés pour la prise de décisions tactiques. [29].

1.4.2. Des soins de santé

L'objectif de la vision par ordinateur dans le domaine des soins de santé est d'établir un diagnostic plus rapide et plus précis que celui que pourrait poser un médecin. Actuellement, les cas d'utilisation les plus répandus de la vision par ordinateur et des le traitement d'images sont liés au domaine de la radiologie et de l'image médicale. Les solutions basées sur la vision par ordinateur trouvent un soutien croissant auprès des médecins en raison de leur diagnostic de maladies et d'affections à partir de divers scanners tels que les rayons X, et l'IRM [30].

1.4.3. Les Drones

La vision par ordinateur joue un rôle important dans la détection des différents types d'objets lors des vols en plein air. Un traitement d'image embarqué très performant et un réseau de neurones de drones sont utilisés pour la détection, la classification et le suivi des objets lors des vols en l'air. Le réseau de neurones des drones aide à détecter les différents types d'objets comme les véhicules, les contreforts, les bâtiments, les arbres, les objets à la surface de l'eau ou à proximité, ainsi que les terrains divers. La vision par ordinateur utilisée dans les drones permet également de détecter des êtres vivants comme les humains, les baleines, les animaux terrestres et autres mammifères marins avec un haut niveau de précision [31].

1.4.4. Les véhicules autonomes

Un domaine qui a capté l'imagination du public est celui des voitures sans conducteur, qui dépendent fortement de la vision par ordinateur et de l'apprentissage approfondi. Bien qu'elle ne soit pas encore en mesure de remplacer complètement le conducteur humain, la technologie des véhicules autonomes a considérablement progressé au cours des dernières années. L'intelligence artificielle analyse les données obtenues auprès de millions d'automobilistes, apprenant du comportement des conducteurs pour automatiser le repérage des voies, estimer la courbure de la route, détecter les dangers et interpréter les panneaux et les signaux de circulation [32].

1.5. Conclusion

L'objectif de ce chapitre est de donner brièvement des notions sur l'image numérique et les techniques de traitement d'images. Nous avons présenté brièvement la technologie de la vision par ordinateur et le rôle important de l'intelligence artificielle dans cette technologie. La vision par ordinateur avec la disponibilité d'images et des vidéos sur internet permettait aux industries des véhicules de créer les systèmes d'aide à la conduite.

CHAPITRE 2

Le système avancé d'aide à la conduite

CHAPITRE 2 : Le système avancé d'aide à la conduite

2.1. Introduction

Le nombre des accidents de la route représente un problème d'intérêt majeur dans le monde. Selon l'organisation mondiale de la santé, 1.2 million de personnes meurent et plus de 50 millions de personnes sont blessés dans des accidents routières chaque année [33]. En 2018, l'Algérie a enregistrée 3.310 décès et 23.570 blessés dans 23.024 accidents de la route, et les récentes études de La Direction Générale de la Sûreté Nationale (DGSN) montrent que le facteur humain demeure la principale cause de ces accidents (conducteurs et piétons), suivi par l'état des véhicules et des routes et de l'environnement, a souligné la responsable, précisant que l'excès de vitesse, le dépassement dangereux, l'imprudence des piétons, les manœuvres dangereuses, et le non-respect de la signalisation sont les principales facteurs causant ces accidents.

Au cours des dernières décennies, la sécurité est devenue une préoccupation majeure pour l'industrie automobile. Des organisations telles que « European New Car Assessment » fournissent aux clients les informations sur le système de sécurité qu'offrent les différentes marques et constructeurs automobile. Ces classements sont basés essentiellement sur les systèmes de sécurité passifs protègent les occupants des véhicules contre les blessures après un accident, par exemple, les ceintures de sécurité, les sacs gonflables, et des tableaux de bord rembourrés. Cependant, au cours des dernières années les constructeurs automobiles ont mis beaucoup d'efforts dans sécurité active, ils s'intéressent de plus en plus à la création des systèmes de sécurité actifs, aussi appelés les Systèmes Avancées d'Aide à la Conduite [34].

Dans ce chapitre, nous introduisons les systèmes avancés d'aide à la conduite, en définissons les technologies utilisées, notamment celles basées sur la vision par ordinateur. Nous allons aussi essayer de donner les différents avantages d'utilisation de ce système, ainsi que les différents problèmes qu'on pourra rencontrer lors de leur mise en pratique.

2.2. Définitions

Les Systèmes Avancées d'Aide à la Conduite (en anglais « Advanced Driver Assistant System (ADAS)») sont des systèmes embarqués intelligents conçus pour être intégrés dans les véhicules, et ils ont connu un grand avancement au cours de ces derniers années, ce qui rend la conduite dans un environnement urbain et périurbain plus facile et plus efficace par l'exploitation des informations fournies par des capteurs installés sur le véhicule afin de percevoir l'environnement autour du véhicule. Les ADASs sont classifiés selon les types des

capteurs utilisés, la plus d'entre eux sont principalement basés sur la vision (caméra), la détection de la lumière et la télémétrie (lidar), et la détection et télémétrie radio (radar). D'autres technologies comme PMD (l'abréviation de mot anglais Photonic Mixer Device) et GPS (l'abréviation de mot anglais Global Positioning Systems) et IMU (l'abréviation de mot anglais Inertial Measurement Units) qui sont aussi utilisés pour améliorer les fonctionnalités des autres technologies[35]. La figure 2.1 présente les types de technologies utilisés dans les applications d'ADAS.

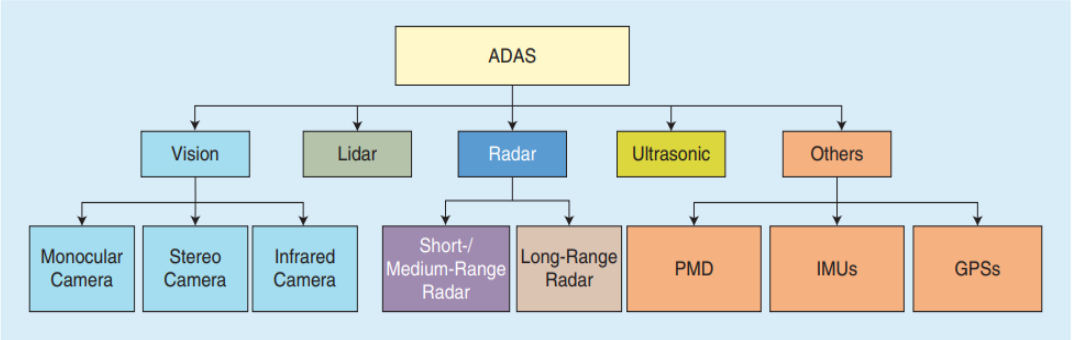

Figure 2. 1. Les catégories d'un ADAS [35].

La plupart des applications d'ADAS font les mêmes processus afin d'aider les conducteurs, ces processus sont : premièrement, Les ADASs utilisent les capteurs que nous avons cités précédemment pour acquis les informations sur l'environnement du véhicule puis, il reconnaît le type de l'information acquis (Comme des images, des signaux, des lumières,… etc.)). Deuxièmement, L'ADAS fait une amélioration sur les informations captés (Comme des images, des signaux,etc.) en utilisant les algorithmes de prétraitement et de traitement, il utilise les algorithmes et les modèles de l'intelligence artificielle (l'apprentissage automatique et l'apprentissage approfondis) pour comprendre et analyser l'environnement du véhicule [35]. Dernièrement, quand le système comprends les informations captées et il trouve que le véhicule ou le conducteur est dans une situation critique (Par exemple une collision avec un autre véhicule, le conducteur quittait sa voie, …etc), ce système prend une décision, soit en prenant le contrôle (freinage d'urgence automatique par exemple), ou prévenir le conducteur pour éviter un accident, ils sont des sécurités actives dans la plupart des systèmes. Un système de sécurité actif peut être défini comme étant un ensemble d'éléments liés au véhicule, à l'homme et à l'environnement, qui par leur présence ou leur fonctionnement peuvent minimiser la gravité de l'accident ou de l'éviter. Ils entrent en action avant l'accident (contrairement aux systèmes de sécurité passive qui interviennent en action pendant l'accident) [36].

Les ADAS pourront être regroupées en deux catégories : ceux destinés à la sécurité du véhicule et de ses occupants, et ceux destinés à la sécurité des usagers de la route. Dans la première catégorie, on peut mentionner par exemple Adaptative cruise control (ACC) qui aide à réguler la vitesse du véhicule afin de maintenir une distance de sécurité raisonnable par rapport au véhicule précédent, ou encore Lane Departure Warning (LDW) qui reconnaît le marquage des voies et qui est activé lorsqu'un conducteur s'apprête à quitter une voie sans utiliser le clignotant. Dans la deuxième catégorie, on peut trouver par exemple les systèmes de détection des piétons qui permettent de reconnaitre les piétons et avertir le conducteur pour qu'il puisse prendre une décision pour éviter la collision avant que le système intervienne et utiliser un freinage d'urgence. Une étude réalisée par ABI Research publiée en 2015 montre que les constructrices automobiles Mercedes-Benz, Volvo et BMW dominent le marché des systèmes d'évitement de collision pour la protection des piétons [36]. On peut aussi trouver les systèmes de détection de somnolence qui alerte le conducteur dès que des signes caractéristiques de fatigue ou d'inattention sont détectés [8]. Quelques soit donc les systèmes proposés, les industriels automobiles cherchent à proposer des ADASs permettant de renforcer la sécurité routière par l'assistance des conducteurs, ce qui permet de réduire les erreurs qu'il peut les commettre.

2.3. Les types de capteurs utilisés

Les ADASs sont classifiés selon ces types des capteurs utilisés. Les capteurs les plus utilisés sont les capteurs de vision, LIDAR, RADAR, ultrasoniques. Ci-dessous nous avons décrit ces capteurs, mais les autres capteurs sont utilisés pour renforcer les fonctions du système comme GPS est utilisé pour connaître l'emplacement du véhicule dans l'environnement, IMU est utilisé pour fournir une position fiable et un discernement de mouvement pour les applications d'ADAS de stabilisation et de navigation, et PMD est utilisé pour faire une détection optique rapide et une démodulation simultanée des signaux lumineux incohérents. [35].

2.3.1. Le capteur de vision

La caméra est le capteur de vision le plus utilisée dans les ADASs basé sur la vision, elle acquit des images qui fournissent des informations visuelles au système pour comprendre et analyser la situation du véhicule dans un environnement (comme un exemple connaître la présence des objets autour du véhicule), aussi elle peut être utilisée pour surveiller l'intérieur de véhicule. Les avantages de l'utilisation de la caméra sont la facilité de l'intégration dans le

véhicule, elle est peu coûteuse, et le plus important, c'est qu'elle peut fournir beaucoup des informations précises à l'environnent au système par rapport les autres captures [35].

2.3.2. Le capteur de LIDAR

La télédétection par laser (En anglais LIght Detection And Ranging (LIDAR)) a une technique très simple pour connaître l'environ de la véhicule, c'est que lidar tire un faisceau de laser puis en mesurant le temps nécessaire pour que le faisceau de laser rebondisse sur ce capteur, Ces systèmes peuvent produire des images 3D à haute résolution et générer une image en 3D de 360° d'environnement avec des informations de profondeur précises. Les avantages de Lidar sont qu'il est utile dans la détection des objets, et plus que ça, les véhicules qui a LIDAR, il peut voir une gamme jusqu'à 60 mètres. Mais ces types des capteurs sont lourds et volumineux et très coûteux, et il a un autre inconvénient, c'est que le changement du météo (par exemple la pluie, un brouillard) peut influencer sur l'exactitude de système [35].

2.3.3. Le capteur de RADAR

Le capteur de détection et de télémétrie radio (en anglais Radio Detection And Ranging (RADAR)) a une grande performance dans les systèmes d'estimation de la distance sécurisé car il donne des donnés précis. Le rôle de radar est l'émission des micro-ondes puis le système mesure le décalage de la fréquence d'ondes (par exemple acoustique ou électromagnétique) observée entre RADAR (l'émission) et l'objet (la réception), ce décalage représente la distance entre l'émetteur et le récepteur varie au cours du temps. Par rapport au LIDAR, le RADAR peut détecter des objets à une longue distance et il n'effluence pas par les conditions de météo (la pluie, un brouillard) et il est peu coûteux. Selon son fonctionnement de calculer la distance, les radars peuvent être classés comme étant à une courte portée (0.2 - 30 mètres), une moyenne portée (30 – 80 mètres), une longue portée (80 - 200 mètres) [35].

2.3.4. Les capteurs ultrasoniques

Ce type de capteur utilise des ondes sonores pour mesurer la distance à un objet. Ces capteurs sont principalement utilisés pour détecter des objets très proches de véhicule [35].

2.4. Les ADAS basés sur la vision

Les industrielles automobiles font des recherches pour créer des ADASs plus sécurisés basés sur la vision artificielle(caméra). En 2015 Tesla a créé le Système Autopilot, qui est une application d'ADAS basé sur la vision qui permet de détecter les voies sur la route, et qui offre un régulateur de vitesse adaptatif, le stationnement automatique et d'autres fonctionnalités.

Pour toutes ces fonctionnalités, le conducteur est responsable et la voiture nécessite une surveillance constante [37].

Dans cette section, nous avons décrit les types des caméras utilisés dans les ADASs basé sur la vision puis nous avons défini les fonctionnements de l'ADAS basé sur la vision et l'utilisation de la vision par ordinateur dans ce fonctionnement, enfin nous décrivions quelques exemples de ce système.

2.4.1. Les types des caméras utilisées dans l'ADAS basée sur la vision

Il y a 3 types de caméra sont plus utilisés dans les ADAS basés sur la vision ils sont les caméras monoculaires, les caméras stéréos, et les caméras thermiques. Ci-dessous, nous allons décrire ces types.

2.4.1.1. Les caméras monoculaires

Le système de caméra monoculaire a une seule lentille, ce que veut dire qu'il donne une seule image à chaque instant, et l'exigence de traitement de l'image est faible par rapport aux autres types de caméras. Ces types de caméras peuvent être utilisés dans diverses applications telles que la détection des obstacles, des piétons (voir figure 2.2), les panneaux de signalisation, et aussi utilisé pour surveiller le conducteur à l'intérieur de véhicule comme par exemple le système de détection de fatigue. Les images de la caméra monoculaire manquent d'informations de profondeur et ne sont donc pas des capteurs fiables pour l'estimation de distance.

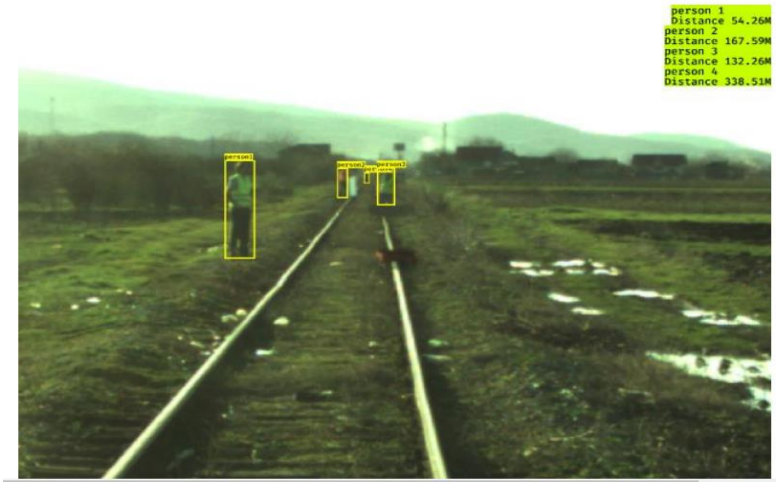

Figure 2. 2. Image RVB d'une caméra monoculaire) [38].

2.4.1.2. Les caméras stéréos

Ce type de caméras se compose de deux ou plusieurs lentilles, l'affichage est présenté comme une matrice d'images. Les caméras stéréos sont plus utiles pour extraire les informations en trois dimensions provenant de deux ou plusieurs images bidimensionnelles en appariant des paires stéréos et en utilisant une carte de disparité pour estimer la profondeur relative d'une scène [35]. Les caméras stéréos sont utilisées dans diverses applications comme la reconnaissance des panneaux de signalisation, les voies, les piétons, et l'estimation de la distance, avec beaucoup plus de précision par rapport aux caméras monoculaires. La plupart des véhicules utilisant les ADASs sont équipés de caméras stéréos, les caméras sont situées à l'intérieur de véhicule, derrière le rétroviseur, légèrement incliné vers le bas, et en face à la route.

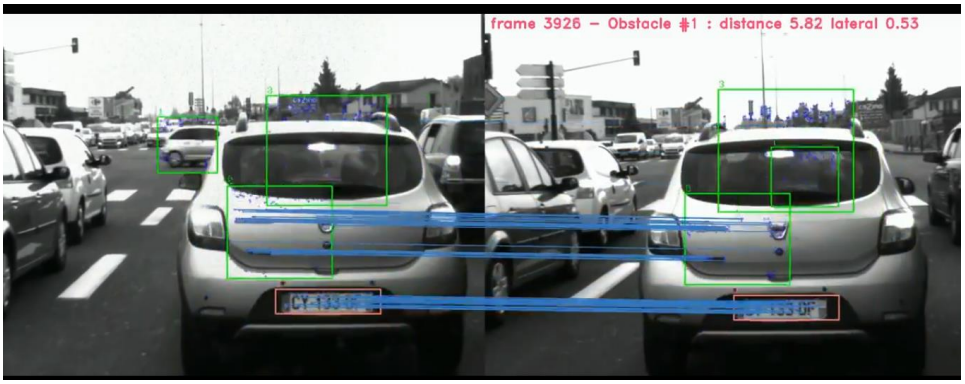

Figure 2. 3. Une image de caméra stéréoscopique [35].

2.4.1.3. Les caméras thermiques (infrarouges)

Il existe deux types de caméra thermiques, une caméra thermique active utilisant une source lumineuse proche infrarouge (avec une longueur d'onde 750 nm à 1400 nm) qui peut être intégré directement dans les véhicules pour illuminer la scène (impossible de voire avec l'œil humain), et une caméra numérique standard pour capturer la lumière réfléchie. La figure 2.5 montre la différence entre les images dans la caméra normale et la caméra thermique active dans la nuit [39]. Une caméra thermique passive utilise des capteurs infrarouges, où chaque pixel sur le capteur infrarouge peut être considéré comme un capteur de température qui peut capturer le rayonnement thermique émis par toute matière, il ne nécessite aucune illumination spéciale de la scène et il est le plus utilisé dans les applications d'ADAS par rapport aux caméras infrarouges active (voir la figure 2.4) [40]. Les caméras thermiques restent une solution

d'utiliser pour la vision nocturne pour aider le conducteur à voir plus clair dans des conditions de faible luminosité.

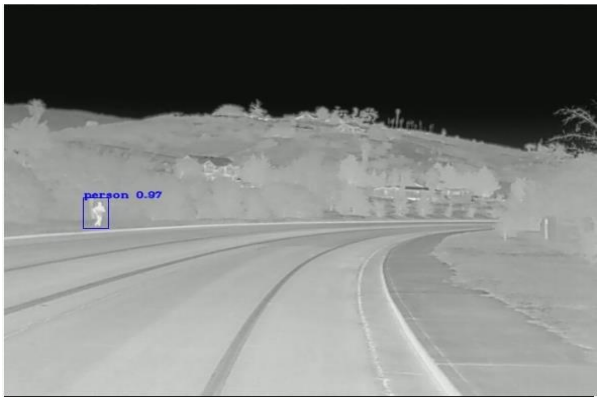

Figure 2. 4. Une capture d'une image à partir d'une vidéo de caméra thermique passive.

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

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

Afin que le système puisse prendre une décision sur l'environs d'un véhicule, il a besoin des informations visuelles aperçues en utilisant des caméras. Dans un premier temps, il capture donc une image (ou une vidéo), puis il améliore la qualité de l'image, ensuite, il utilise des algorithmes qui permet de reconnaître l'environnement autour de véhicule (Par exemple la détection des voies, la détection des véhicules,…etc) et selon le cas, il prend les décisions ou les actions nécessaires (Alerte, freinage d'urgence, signal sonore, … etc.).

2.4.2.1. L'acquisition des images

Il s'agit du processus de capture d'images 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, un Vert, un Bleu (RVB)) c'est-à-dire que chaque image contient trois canaux d'informations (voir le chapitre 1). La cadence d'image dans l'ADAS basé sur la vision est variée entre 5 images par seconde et 60 images par seconde (tout dépend à l'application), comme un exemple les systèmes de l'estimation de la distance (voir la figure 2.2) et la vitesse nécessitent une cadence d'image plus élevée en raison de changement rapide de la distance parcourue par les véhicules sur la route [35].

2.4.2.2. L'extraction et la compréhension de les informations

Au début, l'image obtenue après l'acquisition peut contenir du bruit, et ce bruit peut influencer à l'efficacité des algorithmes. Le traitement préliminaire vise donc à améliorer la qualité des images pour extraire des informations au l'environs du véhicule. Il existe plusieurs techniques de prétraitement d'images comme les méthodes de rehaussement de contraste. Le rehaussement des contrastes se fait en changeant les valeurs initiales de façon à utiliser toutes les valeurs possibles, ce qui permet d'augmenter le contraste entre les cibles et leur environnement. Par exemple, il y a aussi les méthodes de conversion de l'espace de couleur, cela consiste à convertir l'espace de couleur (L'espace RGB, HSV, CMJ, ... etc) d'une image à une autre. Son principal avantage est qu'il permet de séparer les effets négatifs (Un ombre, une illumination irrégulière, ... etc) [35].

Après, il existe des systèmes utilisent les techniques de traitement d'image. Et d'autre systèmes utilisent la technologie de la vision par ordinateur pour extraire les informations et le comprendre, il existe deux catégories de ces systèmes :

- Les systèmes utilisent les descripteurs de caractéristique (Comme les descripteurs HOG, Haar like, SIFT, … etc) pour extraire les informations (les caractéristiques) de les objets qui existent dans l'image capturée à partir d'une vidéo (c'est-à-dire localiser l'objet dans l'image). Puis ils utilisent les algorithmes d'apprentissage automatique pour classifier les objets localisés (c'est-à-dire reconnaitre la classe de l'objet comme les piétons, les camions, les panneaux de signalisation, … etc).
- Les systèmes utilisent les algorithmes (comme YOLO,Mask-RCNN, Faster R-CNN,…etc) qui sont basés sur les architectures de CNN(comme AlexNet, ResNet, VGG,…etc). Ce type des algorithmes utilise les architectures de CNN pour extraire les caractéristiques de l'objet pour localiser l'objet dans l'image, et aussi pour classifier les objets localisés.

Les deux catégories des systèmes font la détection des objets qui existe autour du véhicule, ils peuvent aussi de surveiller les objets ou les conducteurs des véhicules pour reconnaitre ces actions (comme le dépassement des véhicules, la respecte de la distance et la vitesse sécurisée,… etc). La différence entre les deux catégories est quand il existe beaucoup des objets autour de véhicule, la première catégorie des systèmes prends beaucoup de temps pour détecter tous les objets et de les surveiller par contre les autres systèmes font la détection et la surveillance de tous les objets en temps réel.

2.4.2.3. La décision de système :

Dans la phase précédente, le système comprend et analyse les images capturées par la caméra qui représente l'environnement du véhicule. Donc, dans cette phase, le système s'agit de l'interprétation les résultats des phases précédentes pour prendre des décisions. Dans cette phase, la plupart d'ADASs basés sur la vision utilisent la technologie d'avertissement de collision avant et Freinage automatique d'urgence (En anglais Autonomous emergency braking(AEB)) comme une décision ou un sortie. La technologie système d'avertissement de collision avant (En anglais Forward Collision Warning (FCW)) est un dispositif de sécurité active qui permet d'avertir le conducteur en diffusant un signal audible et/ou tactile, ou visuel pour alerter le conducteur à une éventuelle situation de collision. La technologie AEB représente un système intelligent de sécurité actif utilisé dans les ADASs basés sur la vision pour aider les conducteurs à éviter ou de diminuer l'impact d'une collision des véhicules avec les utilisateurs de la route (d'autre véhicules, des piétons, … etc) par l'application d'un freinage d'urgence.

Le défi majeur dans cette phase est que le système peut donner des sorties erronées avec un certain degré de confiance, c'est-à-dire les résultats des phases précédentes sont corrects, mais la sortie du système est fausse.

2.4.3. Des exemples d'ADAS basé sur la vision :

Il existe des ADASs qui utilise une seule caméra ou bien une caméra couplée avec des capteurs RADAR, LIDAR, etc. (voir la figure 2.6). La caméra peut être installée à l'intérieur comme à l'extérieur du véhicule. Dans ce qui suit, nous allons introduire quelque des ADASs

qui utilisent un certain type du caméra (nous avons montré dans la partie précédentes les types des caméras utilisés).

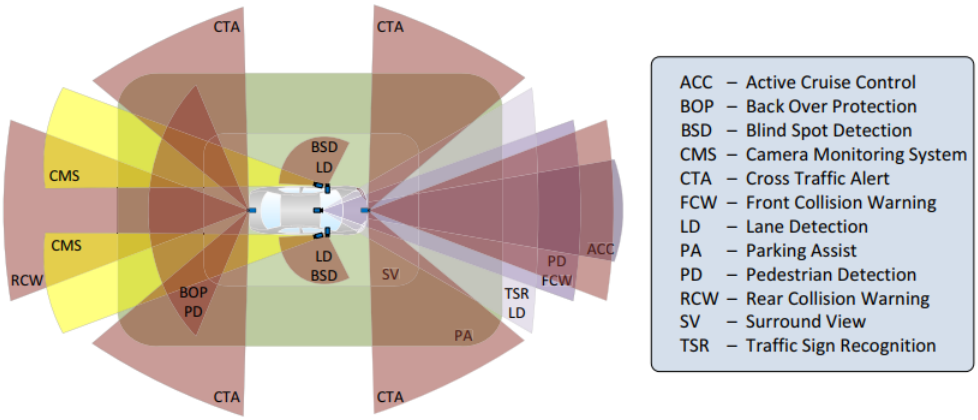

Figure 2. 6. Une représentation des applications d'ADAS [62].

2.4.3.1. Le système de protection des piétons

Les accidents des piétons représentent une grande partie des accidents mortels dans le monde, pour cela les industries et les chercheurs cherchent à proposer des ADASs basé sur la vision pour diminuer le nombre de ces accidents, comme par exemple le système assistant d'alerte de Mercedes-Benz et le système de l'aide d'évitement les piétons de Toyota [36]. Le principe de système de protection des piétons consiste à l'aide des séquences d'images acquises par la caméra installer à un véhicule détecte les piétons puis alerte le conducteur pour éviter les collisions. La figure 2.7 montre le système de détection des piétons en utilisant une caméra. Il utilise les techniques du vision par ordinateur pour détecter les piétons comme les descripteurs (le descripteur du HOG, Haar-like, sift par exemple) et la segmentation (par exemple méthodes de segmentation des régions) ou des algorithmes de détections d'objets qui sont utilisés avec les algorithmes de classification et de régression (apprentissage automatique) comme l'algorithme Adboost, l'algorithme de AGHaar ou l'algorithme de SVM. Pour classifier les piétons et les non-piétons et de reconnaitre leurs mouvements. Pour la surveillance des piétons, les algorithmes le plus utilisés sont Kalman, mean-shift ou alpha-beta tracker [41].

Pour les travaux récents, les chercheurs sont implémentés l'architecture CNN (comme AlexNet, ResNet) pour créer ce système, car cette architecture permet de détecter les piétons et classifier en temps réel [41]. Mais il reste la difficulté de la différenciation entre les êtres

humains et les objets voisins dans des conditions environnementales parmi les grands problèmes qui influence les résultats de ce système.

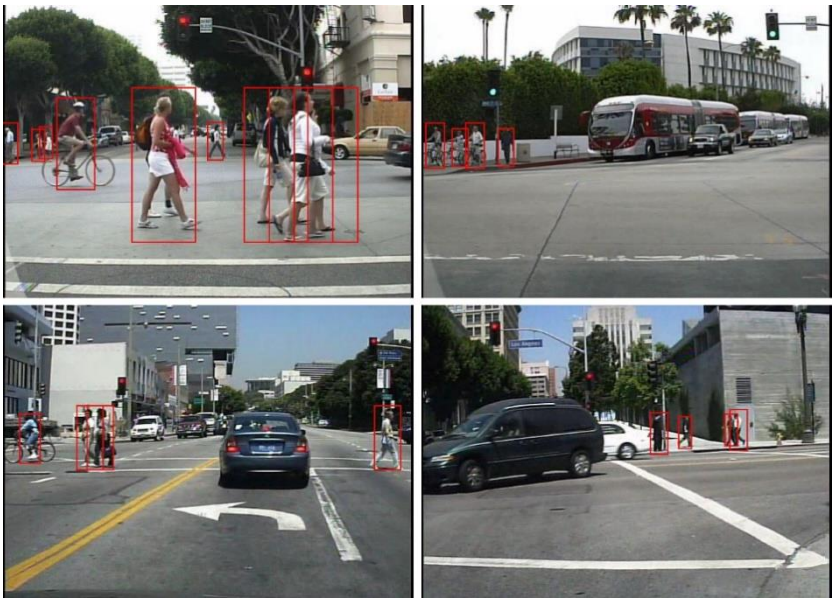

Figure 2. 7. Le système de détection des piétons par une caméra monoculaire[48]

2.4.3.2. Le système assistant de stationnement

Le système assistant de stationnement est ADAS basé sur la vision aide le conducteur à décider de garer son véhicule. Le système consiste à analyser les dimensions des places de station disponibles, cela peut être réalisé en traitant les données d'image capturées à l'avant, à l'arrière et sur les côtés. Il peut aussi de capturer des images autour le véhicule en utilisant une caméra à un objectif de grand angle ou les quatre caméras « fich-eye ».

C.wang et al [43] introduit une méthode d'automatique parking basée sur un système de vision à 360 degrés, couvert par quatre caméras fish-eye embarqué et pré installé autour d'un véhicule. Le système analyse les places de stationnement à l'aide d'un algorithme basé sur la transformation du radon (une méthode de traitement d'image est utilisée pour détecter les objets). De plus, le système planifie le parcours du parking en utilisant une double planification de trajectoire circulaire (une méthode de vision par ordinateur pour guider le conducteur de garer son véhicule dans la place de stationnement).

K.Eyal et al 44] ont proposé un système d'assistant de stationnement, ce système est composé de trois caméras monoculaires standard connectées à un ordinateur, qui sont fixées au véhicule. Au début, il recherche à une place de station libre puis il informe le conducteur, il utilise l'algorithme du transformation d'Hough pour identifier les limites de stationnement disponible puis elle utilise la méthode de filtrage de descripteurs locaux pour filtrer la région

de l'ombre qui existe dans cette place, quand la place est sélectionnée par le conducteur, la position relative entre le véhicule et le lieu de stationnement est surveillée. Il utilise les l'algorithme flux optique par la méthode Lucas-Kanade pour surveillée les objets qui est autour le véhicule. Des instructions vocales et visuelles de guidage du stationnement sont présentées au conducteur. Le système alerte le conducteur à les objets qu'est surveillé par l'algorithme de flux optique. La figure 2.8 montre l'affichage de guidage du stationnement sur l'ordinateur (L'image en haut à gauche - la vidéo de la caméra avant. L'Image en haut à droite - la caméra vidéo du côté droit. L'image en bas à gauche - le mouvement de la voiture détecté par le système. L'image en bas à droite - les instructions que le système donne au conducteur).

Figure 2. 8. L'affichage de guidage d stationnement sur l'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 le plus importantes dans les ADASs basés sur la vision, car les panneaux régularisent le trafic, indique l'état de la route, et ils sont omniprésents. Les panneaux sont caractérisés par leurs formes (un triangle, un cercle, ou en octogonale), leurs couleurs (rouge, un blanc), et aussi par leur contenu. La figure 1.8 montre un exemple d'un système de détection et de reconnaissance des panneaux de routière. Les chercheurs ont proposé des approches pour développer ce système, parmi ces approches :

Ben Romdhane et al [21] utilisent l'espace couleur Teinte-Saturation-Valeur (TSV) pour générer des régions candidates. Les caractéristiques d'HOG (voir le chapitre 1) sont utilisées comme descripteurs de caractéristiques, et la méthode de les machines à des vecteurs du support est utilisée pour identifier la catégorie de panneaux de signalisation.

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

Figure 2. 9. Système de détection et de la reconnaissance des panneaux de signalisation[21].

2.4.3.4. La détection de fatigue

La conduite en état de somnolence est une cause majeure des accidents de la route, et expose les conducteurs à des risques de collision très graves. Les méthodes de détection de fatigue chez le conducteur sont basées sur la description de certaines expressions faciale, il existe des systèmes qui utilise des mesures appliquées à ces caractéristiques extraits à partir des yeux et de la bouche. De plus, l'utilisation des algorithmes d'apprentissage (Comme méthode de viola & jonas, ou l'algorithme Multi Task-CNN, … etc) permettant de classifier les couplets de mesures en deux états possibles : fatigué, ou normal. Ces valeurs permettent de distinguer les signes de fatigue (bâillements, yeux fatigués) des autres expressions de visage (sourire, parler, etc.) [8]. Si le conducteur est fatigué, le système alerte le conducteur pour prendre une décision (par exemple arrêter le véhicule). La figure ci-dessous représente le résultat de système de détection de fatigue.

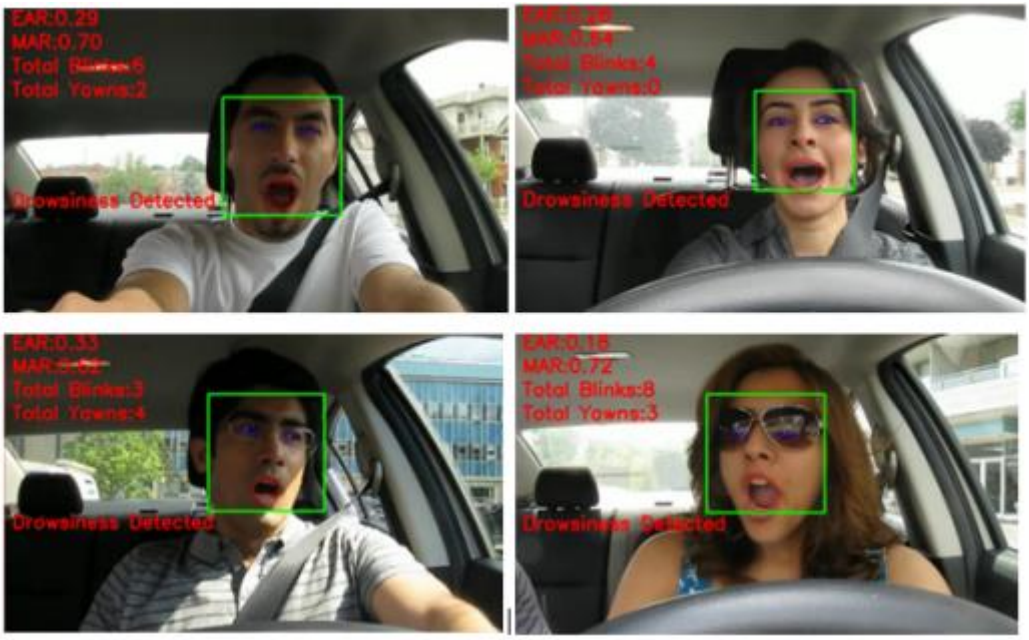

Figure 2. 10. Alerte de somnolence lorsque le seuil passe la 4ème fois Détections [8].

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 sécurisé est parmi les composants le plus important dans l'ADAS basés sur la vision, car la collision par l'arrière représente une grande partie de l'ensemble des accidents (29,5 % aux États-Unis et 29 % en Allemagne). Le manque d'attention (dormir, utiliser des appareils de communication, etc.) représente 91 % de l'accident lié au conducteur. Si les conducteurs sont conscients de la collision plus tôt, 60 % des collisions par l'arrière peuvent être évitées, et 90 % des collisions peuvent être évitées en les remarquant plus tôt, au bout d'une seconde [38]. Ce système consiste à détecter et surveiller les véhicules, à estimer la distance avec chaque véhicule et à estimer la vélocité du véhicule, après il propose au conducteur la vitesse sécurisée pour éviter la collision. Au début, la plupart des systèmes font une amélioration à la séquence des vidéos (Par exemple augmente le contraste et illumination, l'utilisation le filtre de Median,…etc). Pour la détection et la surveillance des objets, il existe des systèmes utilise des algorithmes des descripteurs de les caractéristiques (comme SURF, Haar-like), ou les techniques de traitement d'images(Comme le filtre de Gaussian derivative, le filtre de Kalmen, …etc) et les algorithmes de l'apprentissage automatique(par exemple Adboost, la méthode SVM, le classificateur d'Haar)[56], aussi d'autre systèmes utilise les algorithmes qui utilisent l'architecture de CNN(comme YOLO, R-CNN, fast R-CNN)[45]. Pour l'estimation de la distance, Muhammad Abdul Haseeb et al [38] ont proposé une nouvelle méthode, cette méthode basée sur le réseau neuronal multicouche

caché, appelé DisNet, qui est utilisé pour apprendre et prédire la distance entre l'objet et la caméra, Hasan et al. [45] ont proposés une méthode d'estimation de la distance entre les véhicules en utilisant la vision stéréo, mais il ne peut mesurer la distance entre les véhicules que dans un rayon de 20 m.

Pour l'estimation de la vitesse, Yu et al [45] utilisent la méthode de calibrage de la caméra pour obtenir la relation entre la caméra et le monde réel, puis utilise la méthode de correspondance d'image pour calculer le changement de localisation du véhicule afin d'estimer la vitesse du véhicule, Wang [46] a utilisé la connaissance a priori de la taille du véhicule pour réaliser le suivi des objets, puis il a utilisé la différence dans des cadres du vidéo pour calculer la vitesse du véhicule.

2.4.3.5. Le système de la détection des feux de circulation

Le système de la reconnaissance des feux de circulation est devenu très importante pour l'ADAS basé sur la vision. Il fournit des informations essentielles au conducteur sur les croisements et les passages pour piétons et il peut réduire le nombre d'accidents à cause de la distraction au feu de circulation. Il détecte le feu de circulation dans un environnement urbain puis reconnaître son statut pour avertir le conducteur. Les chercheurs ont proposé des méthodes et des algorithmes pour créer ce système :

Tae-Hyun H et al [47] ont proposé en 2006 une approche de détection des feux de circulation qui consiste en un seuillage de couleur sur la partie supérieure de l'image complété par une convolution gaussienne, afin de détecter l'émission de lumière des feux de circulation.

Gwang-Gook. LEE et al [48] ont proposé une méthode qui détecte le feu de circulation et reconnaît la lumière qui allume (rouge ou vert ou orange). Au début, il applique l'algorithme l'amélioration discriminante de les couleurs rouge et vert (un algorithme de prétraitement) pour changer l'intensité du couleur rouge et vert, puis il applique la méthode de segmentation des régions et le filtre de région pour localiser le feu de circulation. Enfin, il utilise les architectures de LeNet (voir chapitre 1) pour classifier le feu de circulation et AlexNet (voir chapitre 1) pour classifier et les de la lumière.

2.5. Les avantages des ADASs :

2.5.1. Le potentiel de réduction des accidents

Les études montrent que les ADASs peuvent réduire les accidents mortels de piétons et de cyclistes jusqu'à 30 % ou plus dans certaines circonstances. Une étude réalisée en 2013

par les assureurs allemands d'accident estime que 18,1 % des décès de piétons tué par des camions en Allemagne pourraient être évités par l'adoption complète de caméras de recul avec freinage automatique [36].

2.5.2. Coût réduit :

Les ADASs sont relativement peu coûteuses (des centaines à des milliers de dollars), en plus de son facilité d'installation [36].

2.5.3. La connexion entre les ADASs et le système télématique :

Les technologies avancées des ADASs peuvent être reliées avec les systèmes télématiques, ce qui permet d'informer les conducteurs au changement urbain des villes, leur capacité d'identifier des comportements dangereux dans les véhicules comme par exemple la possibilité d'accéder aux données identifiant les lieux des projets de réaménagement des routes [36].

2.5.4. L'utilisation des technologies FCW et AEB :

La capacité des technologiex FCW et AEB utilisées dans l'ADAS peuvent réduire considérablement les collisions possibles dans les routes et rendre la conduire plus facile, plus sécurisé, et plus agréable [36].

2.6. Les défis majeurs des ADASs :

2.6.1. Les changements météorologiques :

L'un des problèmes majeurs des ADASs est que les performances du système sont fortement influencées par l'évolution des conditions environnementales et météorologiques, par exemple pour les applications ADASs qui utilise une caméra avait dans des conditions d'éclairage défavorables (trop lumineux/trop sombre, etc.) [36].

2.6.2. La consommation d'énergie :

Les ADASs impliquent l'exécution de plusieurs algorithmes complexes qui entraînent une consommation d'énergie et une dissipation thermique élevées. En raison de la disponibilité limitée de l'énergie dans les véhicules, il est essentiel de minimiser la consommation électrique du système embarqué utilisé par les ADASs [36].

2.6.3. La sécurité :

Les véhicules modernes sont de plus en plus connectés à des nombreux systèmes différents, tels que le Wi-Fi, la communication en champ proche, etc. Cela permet au véhicule

de détecter et de recevoir diverses informations, mais le rend également plus vulnérable aux attaques. Des nombreux piratages de véhicules ont été commis. Par exemple, le système télématique d'une Jeep Cherokee a été piraté pour accélérer, freiner et couper le moteur. Ce problème peut avoir des conséquences graves dans les ADASs [36].

2.6.4. Les contraintes géospatiales :

Tous les pays (ou certains États d'un pays) n'adhèrent pas de manière uniforme aux mêmes conventions de signalisation et de route, ce qui rend les algorithmes utilisés dans l'ADAS souvent formés dans un lieu et difficiles à utiliser efficacement dans d'autres lieux [36].

2.7. Conclusion

Dans ce chapitre, nous avons présentés le concept de base des systèmes avancés d'aide à la conduite ADAS, et nous avons définis le rôle important de la technologie de la vision par ordinateur et les techniques de prétraitement et de traitement d'images dans la création des ADAS basés sur la vision, dans le but de réduire le nombre des accidents, ainsi que leur intégration dans les véhicules autonomes. Dans le chapitre suivant, nous allons décrire un ADAS basé sur la vision qui détecte les objets dans la route (les véhicules, les piétons, les panneaux d'arrêts, les feux de circulation, etc.), puis estimer la distance entre la caméra et l'objet détecté.

Chapitre 3

La conception de système

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 boite 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 donc 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 circulations, 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. Etat de l'art :

Cette section est décomposée en 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 du caméra et l'objet détecté par une application d'ADAS.

3.2.1. La detection :

La détection d'objets est une technologie informatique qui utilise le traitement d'images et l'intelligence artificielle (l'apprentissage automatique et l'apprentissage approfondis) 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 que la 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, elle est devenue un domaine d'exploration majeur au sein de la vision par ordinateur. Dans la précédemment 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 rajouter quelques d'autres 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 reconnaissance le panneau de 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 de panneau de STOP sont étudiées et utilisées pour le détecter. Pour le module de reconnaissance, le 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 souvent 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 capturé 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.

$$Z=f * B / d \tag{3.1}$$

Telle que « 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 la chapitre 1) qui fait parti 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.

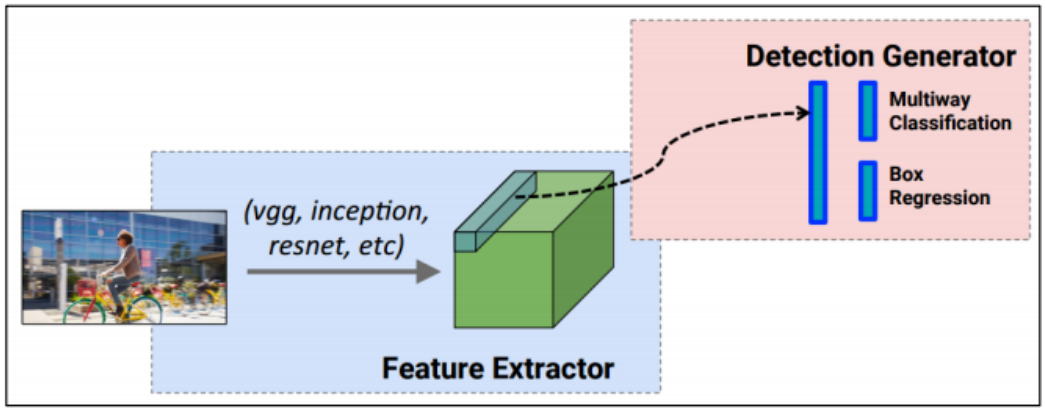

Figure 3. 1. Architecture et composition de SSD [53].

Dans ce qui suit, nous allons décrire chacune de ses 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'image sont généralement construits en utilisant des méthodes d'extraction de caractéristiques visuelles puissant comme les réseaux neurones convolutifs ou CNN (Deep Learning). Dans 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 rejetant les couches entièrement connectées. La raison d'utiliser VGG-16 en tant que réseau de base est de sa forte performance dans les tâches de classification d'image [53].

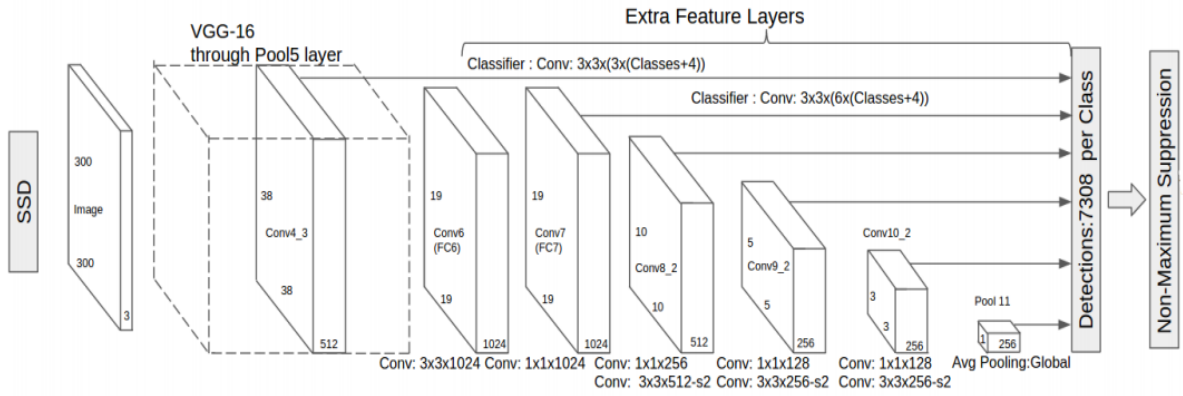

Figure 3. 2. L'architecture de réseau de base (VGG-16) avec une détecteur SSD [27].

Nous devons donner un petit exemple représente le 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). Et puis appliquez la convolution à la couche intermédiaire et obtenez la couche supérieure (2x2) où chaque entité correspond à une région 7x7 sur l'image d'entrée. Des cartes d'entités qui 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 regarde) 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 du notre exemple. Nous pouvons dis que l'architecture de base de SSD utilise le même fonctionnement de l'exemple donné mais il a plusieurs couches de convolution.

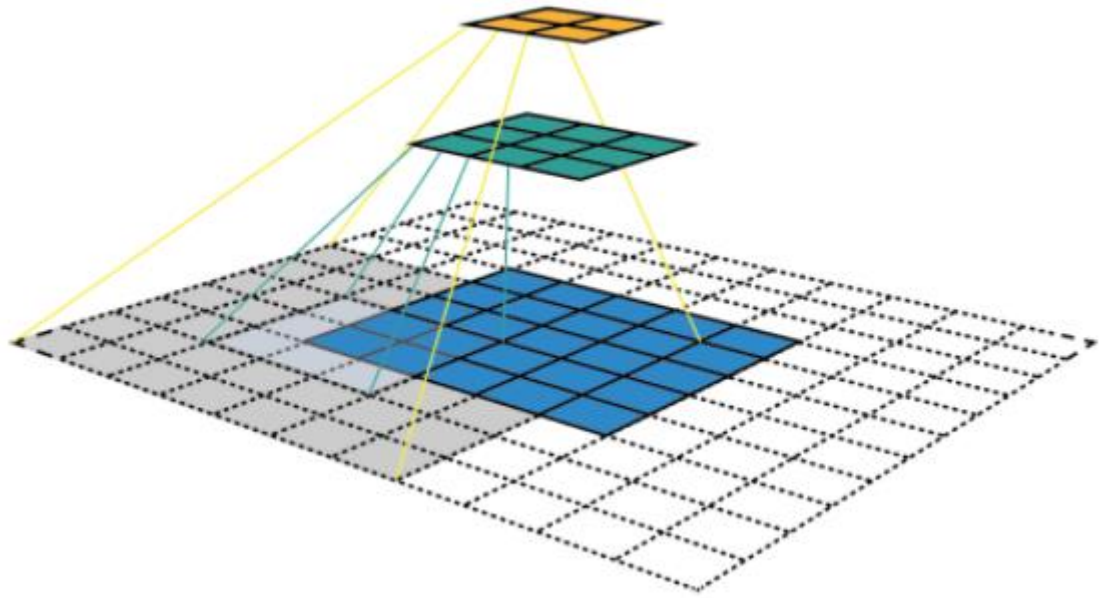

Figure 3. 3. La visualisation des cartes des caractéristiques CNN et du champ réceptif [54].

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.

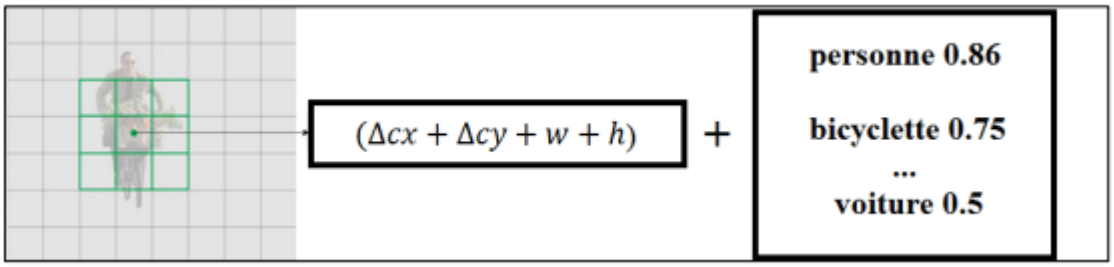

Figure 3. 4. L'utilisation d'un filtre de 3*3 pour la localisation et la classification [27].

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 couches 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 en différents échelles dans les cartes d'entités [53]. Considérez l'image ci-dessous (figure 3.5), observez 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].

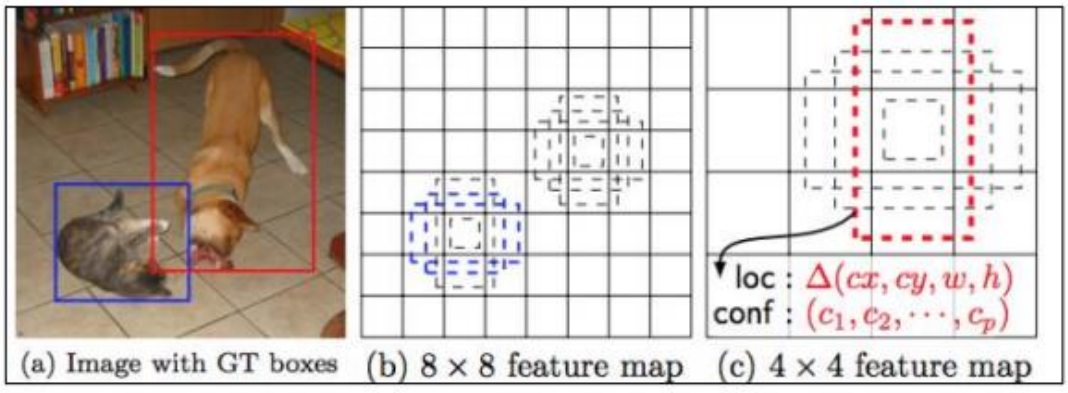

Figure 3. 5. Utilisation des cartes d'entités de différentes couches pour la détection [27].

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 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.

Pour 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. Maintenant, 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 (Δcx, Δcy, 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.

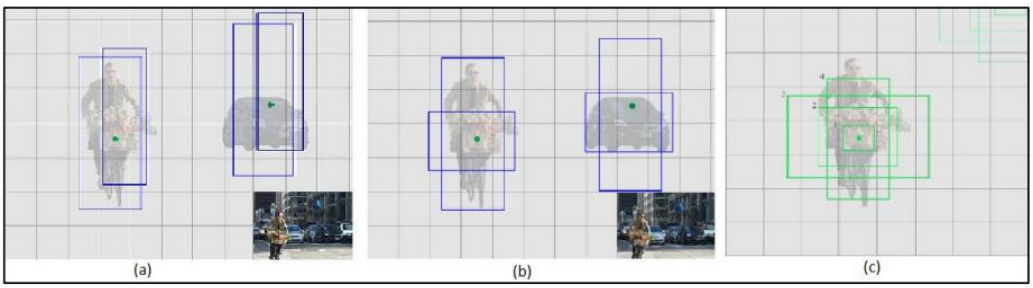

Figure 3. 6. Utilisation de plusieurs boites par défaut dans une seule cellule [27].

3.3.4. Les boites 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 boites de délimitation, Il est parfois appelé l'indice de Jaccard (voir figure 3.7) [55]. Par exemple pendent le temps de l'entrainement, les boites par défaut correspondent au rapport hauteur / largeur, à l'emplacement et à l'échelle aux boites de vérité. Nous sélectionnons les boîtes avec le chevauchement le plus élevé avec les boîtes englobantes de vérité. IoU (intersection over union) entre les boites délimitations prédites et la boite vérité doit être supérieur à 0,5. Nous prenons enfin la case prédite avec un chevauchement maximal avec la case vérité.

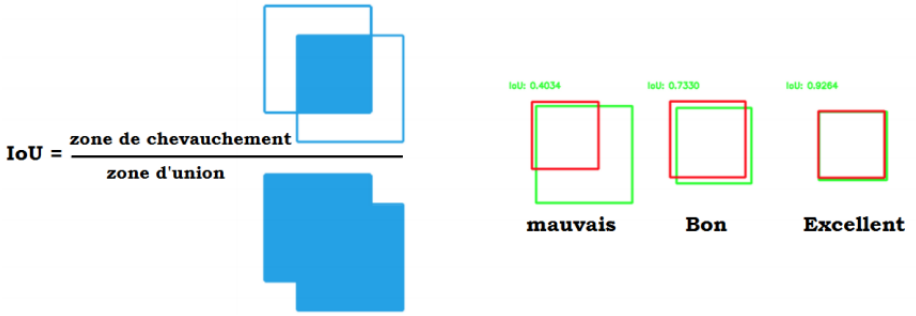

Figure 3. 7. Diagramme d'explication d'IoU (l'indice de Jaccard) [55].

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

Notre système détecte les objets du circulation routière (voiture, camion, piéton, bicyclette, feue 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 du 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 du circulation routière dans une vidéo en temps réel.

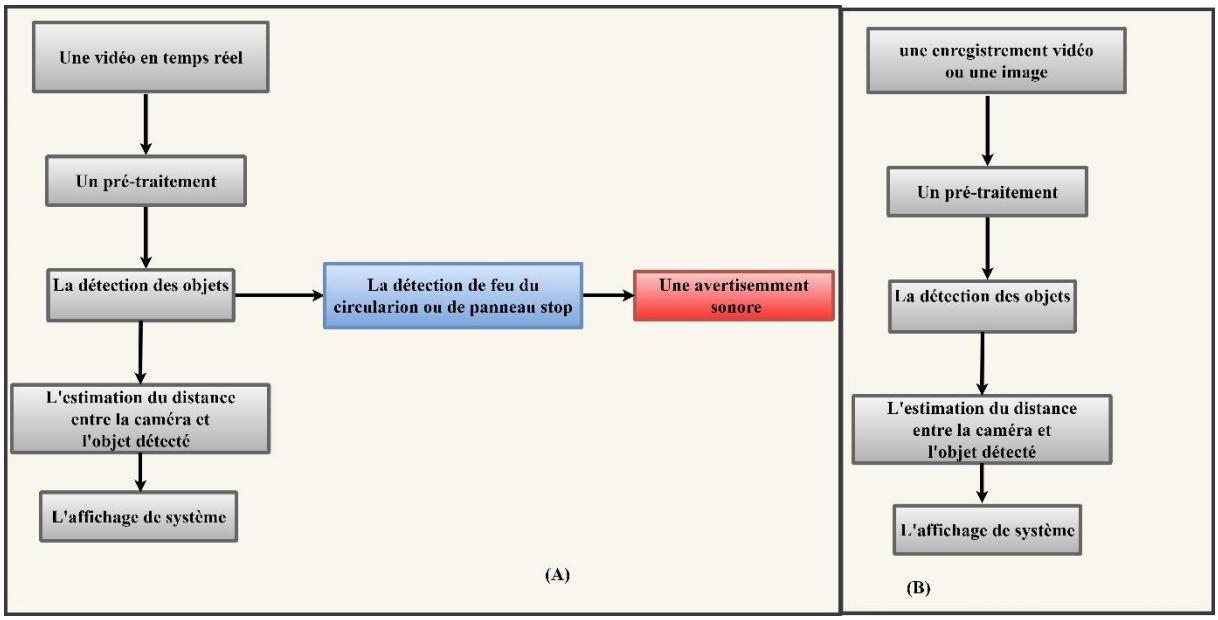

Figure 3. 8. La conception en générale du notre système.

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 a de taille 300*300) nous avons utilisé l'architecture ci-dessous (figure 3.9), qui est composée de :

- 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.

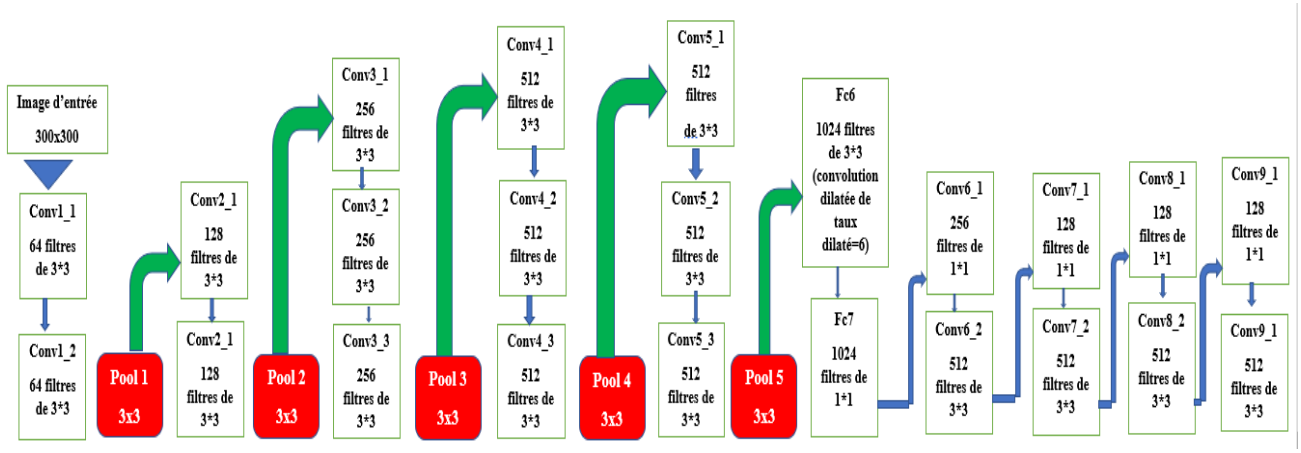

Figure 3. 9. Diagramme de réseau de base de modèle SSD300.

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 du base (CONV4_3, CONV_Fc7, CONV6_2, CONV7_2, CONV8_2, CONV9_2).

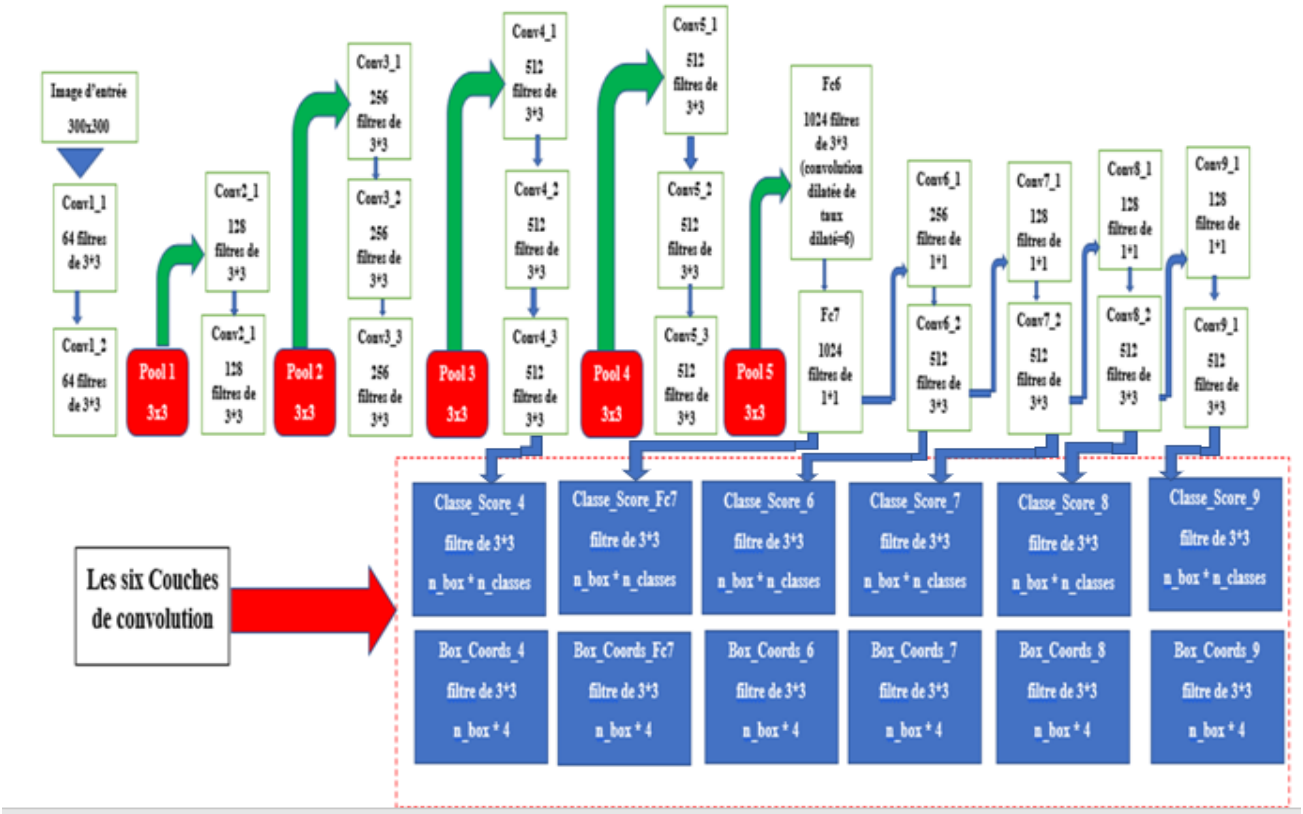

Figure 3. 10. Addition des couches de localisation et classification au réseau de base.

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 aux autres tâches, c'est-à-dire lorsque vous effectuez un apprentissage par transfert : vous prenez un modèle qui a été formé sur quelque chose et vous utilisez une partie ou la totalité de ce modèle pour entraîner un nouveau modèle, en mettant à jours 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 du circulation routière. Nous avons utilisé l'apprentissage par transfert pour créer un nouveau modèle qui détecte les objets du circulation routière le diagramme ci-dessous représente les étapes de création de notre nouveau modèle SSD300.

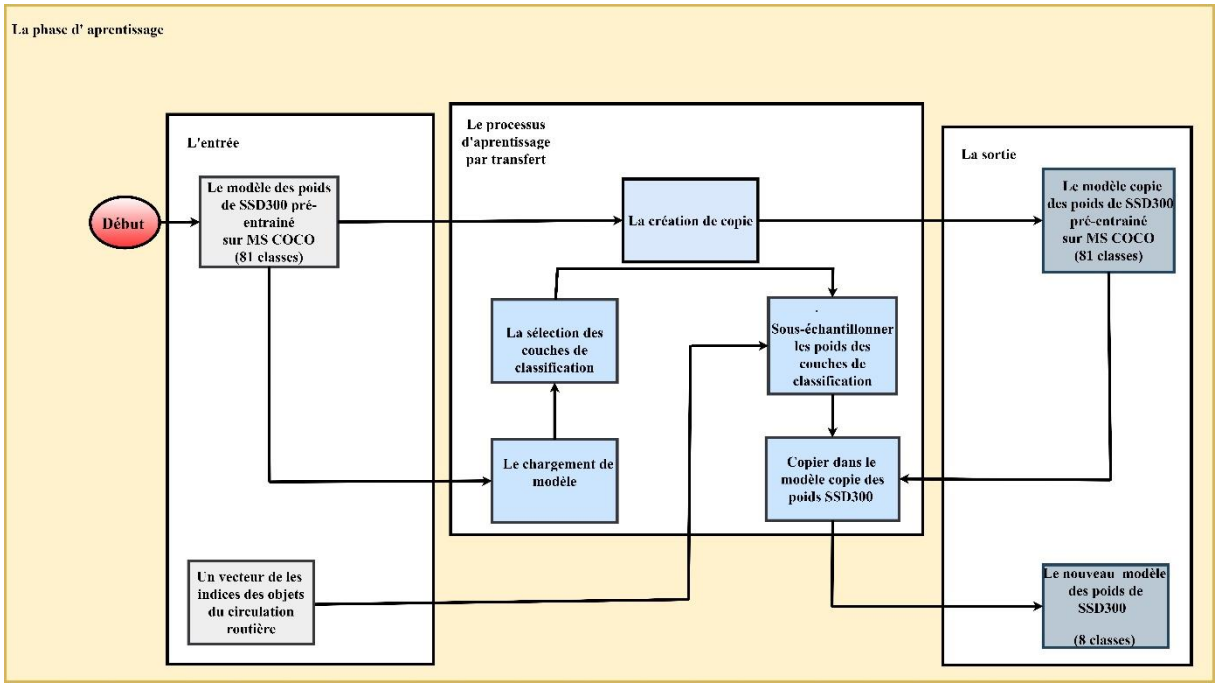

Figure 3. 11. Le diagramme de la phase d'apprentissage de système.

- **Rechercher des indices des objets du circulation routière et créer des vecteurs**

Nous devons rechercher dans un premier temps manuellement les indices d'objets du circulation routière dans l'ensemble des donnée 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 circulation routière (voiture, camion, piéton, bicyclette, feue 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 prédit la localisation des 4 boites qui englobants les objets du circulation routière et qui prédit aussi les classes.

- **Le chargement de modèle SSD pré-entrainé et la création d'une copie**

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

- **La sélection des couches de classification**

A ce niveau, nous devons déterminer exactement les poids des couches qui classifier les objets du circulation routière que nous devons sous-échantillonner (les couches de classification), mais les poids des autres couches reste inchangée. Les couches de classification dans SSD300 sont : classes_score_4, classes_score_Fc7, classes_score_6, classes_score_7, classes_score_8, 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 du circulation routière à partir du modèle de pondérations source (SSD300 pré-entraîné sur MS COCO). Après, nous allons calculer les indices de sous-échantillonnage pour les derniers neurones. La taille et le nombre du noyau restent inchangés. Enfin, nous allons remplacer le noyau correspondant et les tenseurs de biais dans notre modèle de pondérations de destination (le nouveau modèle SSD300 qui détecte les objets du 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 prédiction de confiance supérieure, SSD évalue si les boîtes de limites prédites précédemment ont une IoU supérieure à 0,45 avec la prédiction actuelle pour la même classe. Si trouvé, la prédiction actuelle sera ignorée.

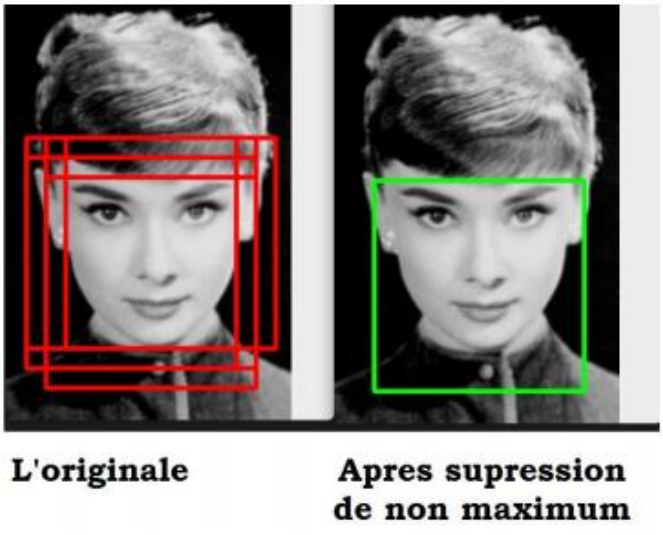

Figure 3. 12. La suppression de non maximum [54].

3.4.2. L'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 boite englobante de l'objet détecté et l'objectif de la caméra. Dans ce qui suit le principe de cette méthode.

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

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

Distance = (2 * 3.14 * 180) / (la hauteur du boite + la largeur du boite * 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.

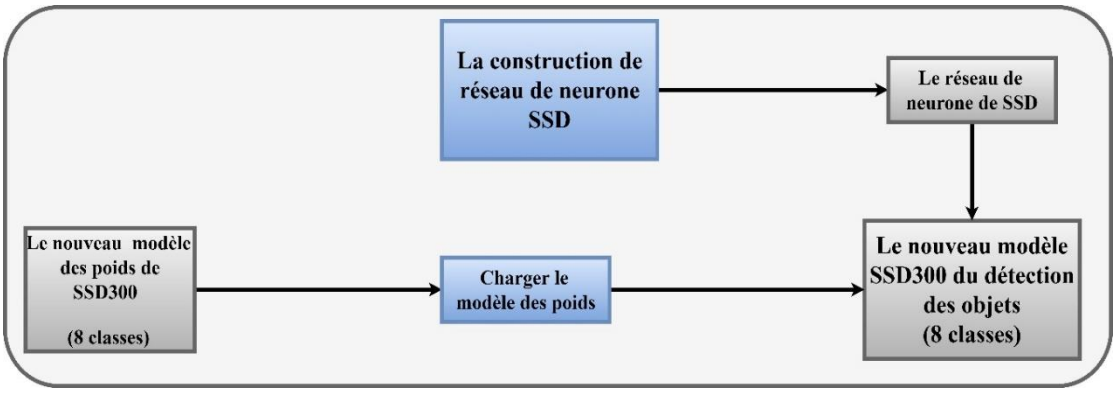

Figure 3. 13. Le diagramme représente les étapes de la construction.

3.4.3.2. L'acquisition d'image et le pré-traitement

Ensuite, nous allons extraire les frames à partir un enregistrement vidéo ou une vidéo en temps réel (une webcam). Puis, nous allons appliquer des algorithmes de pré-traitement pour changer de la taille d'image en 300*300, car la taille d'image d'entrée dans notre modèle SSD300 doit être 300*300. La figure 3.12 représente le diagramme qui représente cette opération.

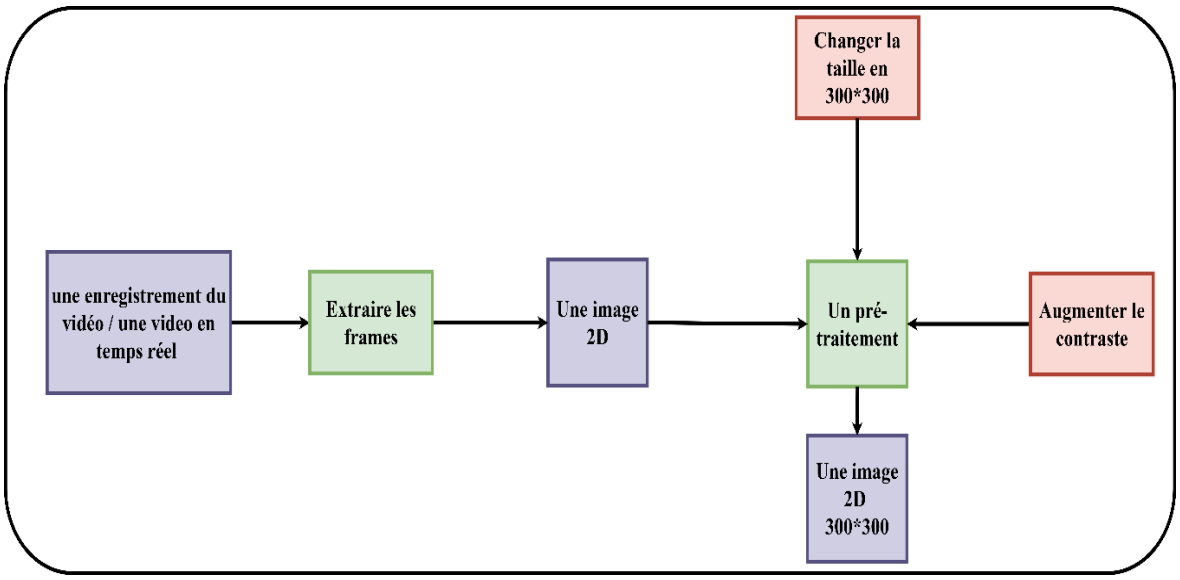

Figure 3. 14. Le diagramme représente lecture du vidéo et le pré-traitement.

.

3.4.3.3. La 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 qui au alentour de 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.

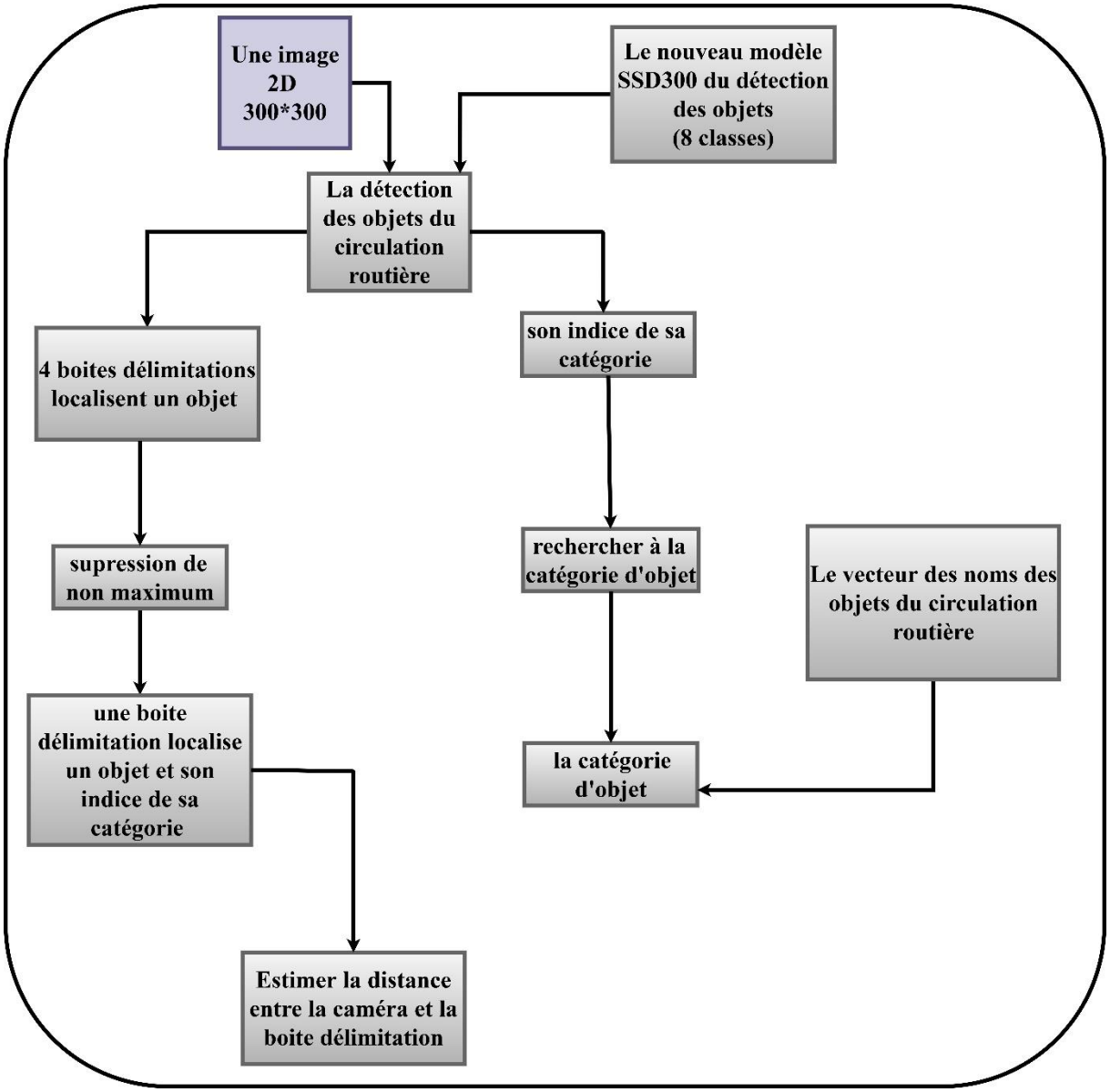

Figure 3. 15. Le diagramme représente le processus du détection des objets et l'estimation de la distance.

3.4.3.4. L'affichage du système

Notre système est sensé détecté les objets suivants : les voitures, les camions, les bus, les motos, 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 du 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 aux 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 détectés. Nous avons donc 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

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 utilisé dans le développement de notre application, ensuite nous présentons les algorithmes utilisés, ainsi que les résultats obtenus dans diverses situations.

4.2. Les outils utilisés

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

4.2.1. Le matériel utilisé

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.
- Ecran : 17.3 inch.
- Système d'exploitation : Windows 10, 64 bits.

4.2.2. Environnements et outils de développement utilisés

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

- **Python[1]**

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 sollicité par une large communauté de développeurs et de programmeurs. Les packages (bibliothèques) de python encouragent la modularité et la réutilisabilité des codes.

- **OpenCV [2]**

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 au travers d'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 [3]**

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.[1]

- **TensorFlow[4]**

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, Reconnaissance de voix,… .

- **Keras [5]**

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. Il a été développé 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. Il a été développé 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 à Google.

[1] https://www.python.org/
[2] https://opencv.org/
[3] http://www.numpy.org/
[4] https://www.tensorflow.org/
[5] https://keras.io/
[6] https://riverbankcomputing.com/software/pyqt/
[7] https://www.sublimetext.com

- **PyQt5** [6]

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** [7]

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. Il 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-entrainé et la charger**

Tout d'abord, nous allons faire une copie du modèle VGG_SSD300 pré-entrainé 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')
```

Figure 4. 1. La création de la copie.

La boucle ci-dessous est divisé en deux parties. La première partie sélectionne les couches de classification dans le modèle source puis en utilisant le vecteur des indices qui indique aux éléments qui classifient les objets de la circulation routière, elle extrait les filtres et biaises 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 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 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 biases**

```
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 biaises.

- **Copier les filtres et les biaises et créer un nouveau modèle des poids**

Dans la deuxième partie de la boucle, nous créons les nouveaux tenseurs des biaises et les filtres par la fonction de **sample_tensors()**, ensuite nous devons remplacer

les anciens tenseurs des biaises et les filtres par les nouveaux biaises 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 les objets du

```
    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-entrainé 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[0][0]
fc7_mbox_conf (Conv2D)	(None, 19, 19, 486)	4479462	fc7[0][0]
conv6_2_mbox_conf (Conv2D)	(None, 10, 10, 486)	2239974	conv6_2[0][0]
conv7_2_mbox_conf (Conv2D)	(None, 5, 5, 486)	1120230	conv7_2[0][0]
conv8_2_mbox_conf (Conv2D)	(None, 3, 3, 324)	746820	conv8_2[0][0]
conv9_2_mbox_conf (Conv2D)	(None, 1, 1, 324)	746820	conv9_2[0][0]

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

conv4_3_norm_mbox_conf (Conv2D)	(None, 38, 38, 36)	165924	conv4_3_norm[0][0]
fc7_mbox_conf (Conv2D)	(None, 19, 19, 54)	497718	fc7[0][0]
conv6_2_mbox_conf (Conv2D)	(None, 10, 10, 54)	248886	conv6_2[0][0]
conv7_2_mbox_conf (Conv2D)	(None, 5, 5, 54)	124470	conv7_2[0][0]
conv8_2_mbox_conf (Conv2D)	(None, 3, 3, 36)	82980	conv8_2[0][0]
conv9_2_mbox_conf (Conv2D)	(None, 1, 1, 36)	82980	conv9_2[0][0]

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 du détection), puis il charge le modèle des poids pour mettre à jour les valeurs des filtres et des biaises. 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.

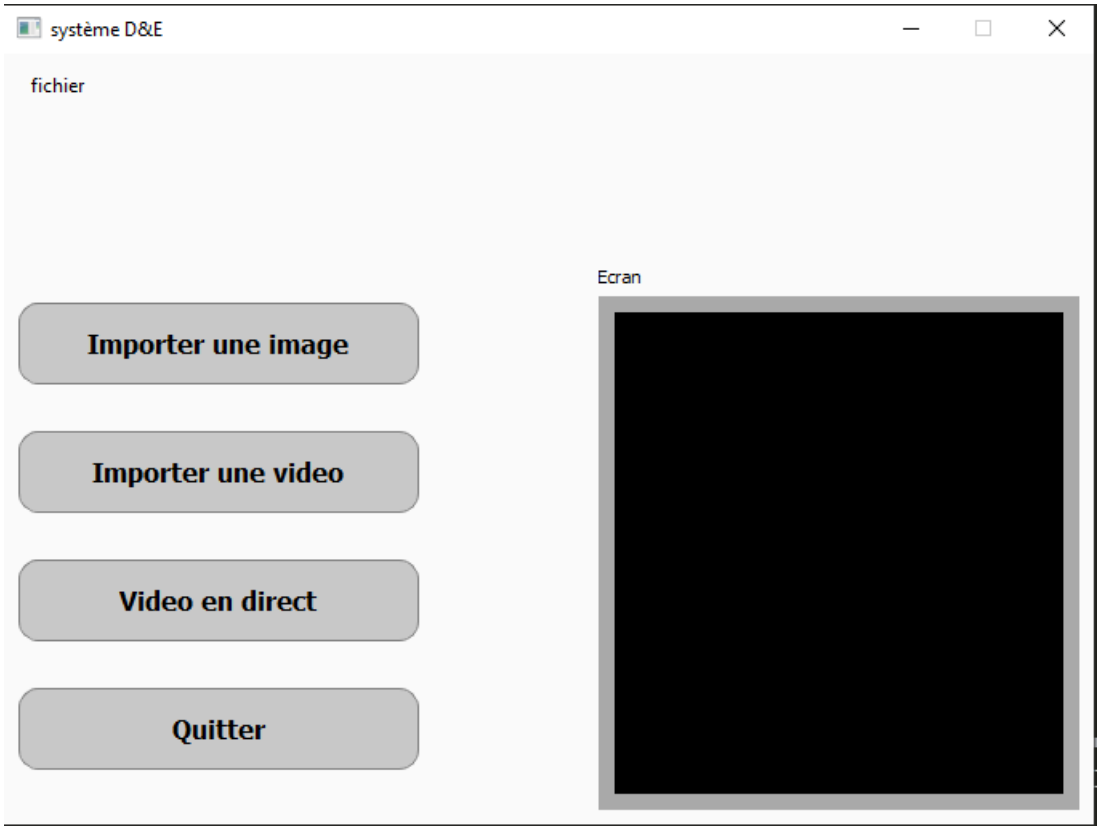

Figure 4. 7. Les taches de système

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

Pour que le système peut 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.

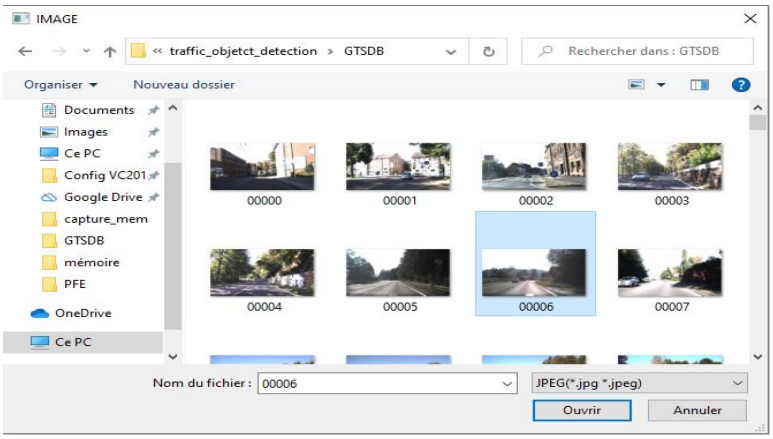

Figure 4. 8. Le choix d'une image.

Puis, le système traite l'image (il applique un changement sur la taille d'image pour la transformer en 300*300), pour qu'on puisse utiliser le modèle SSD300 obtenu et la méthode de

suppression non maximal pour localiser les objets du circulation routière dans l'image (il dessine une boite 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).

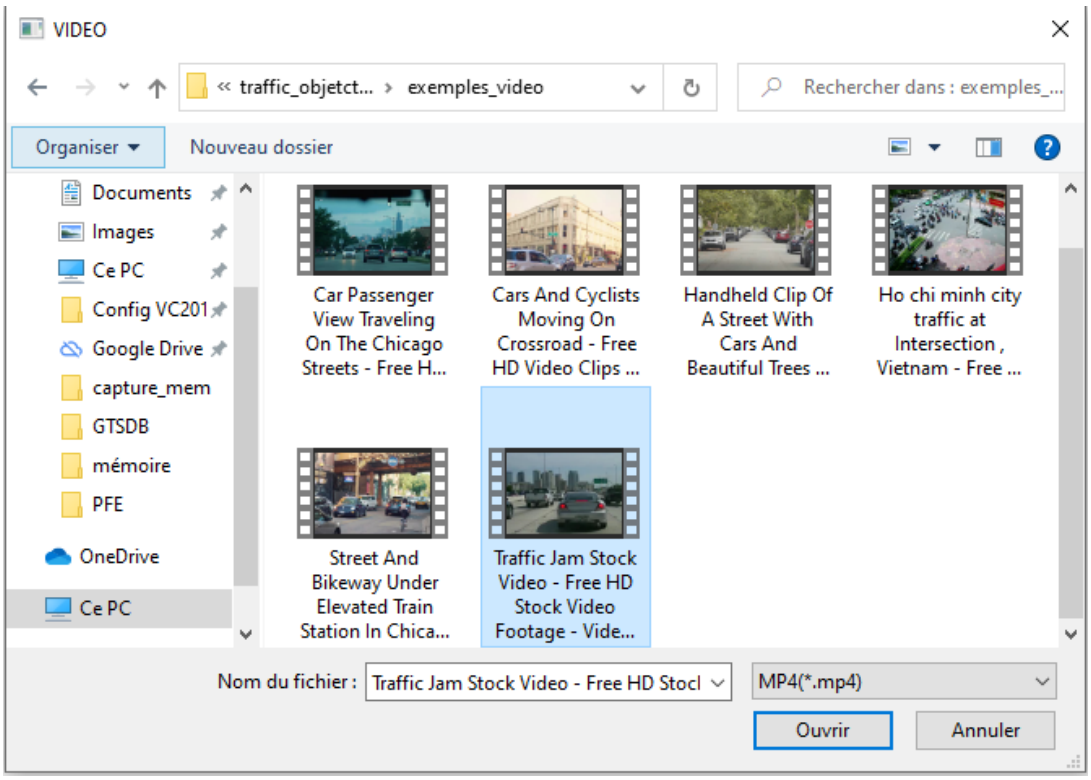

Figure 4. 9. Le choix d'un enregistrement vidéo.

- **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), puis il applique les mêmes taches qu'ont a appliqué 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 boite rouge autour de ces objets, et il 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 boite délimitation qui autour l'objet). Cette fonction récupère les coordonnées prédites par le modèle SSD300 et la méthode de suppression non maximal pour calculer les valeurs de hauteur et de largeur de la boite, puis il utilise la formule que nous avons expliqué 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, ou 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**

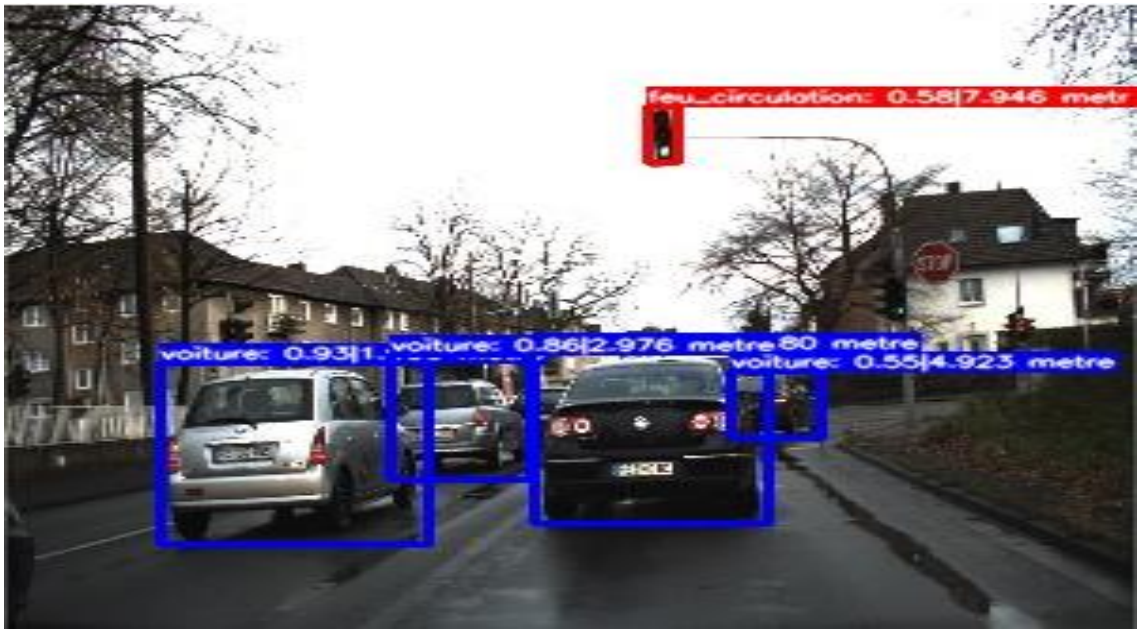

Figure 4. 12. La détection des objets et l'estimation de distance dans la première situation.

Notre système a localisé les feux du circulation et les panneaux STOP et les repère par une boite rouge, les autres objets par une boite bleu (voir les figures 4.12 et la figure 4.13), et il 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**

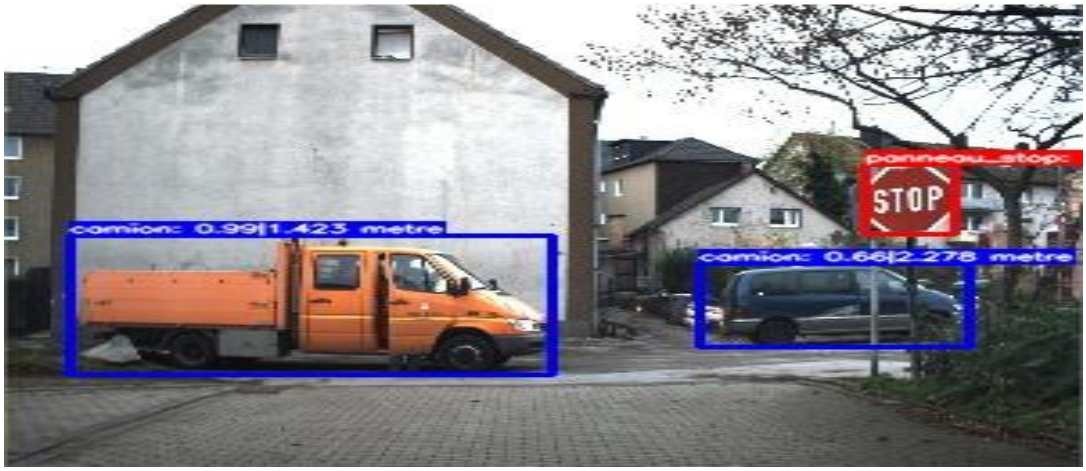

Figure 4. 13. La détection des objets dans la deuxième situation.

Voici d'autres résultats présentés dans la figure 4.13 ci-dessous, nous remarquons que le modèle a fait une confusion entre le camion et la voiture, c'est car elles ont une forme similaire. Donc nous constatons que notre système est moins robuste dans 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 voir 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 limitation 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 de système de détection des objets du circulation routières 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 du circulation routière tels que les voitures, les camions, les feux de la circulation, les piétons, les panneaux STOP, les bicyclettes, les motocycles, les buses.
- 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 de 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 distances (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, les calculs exacts de la distance, etc.

En ce qui concerne nos prochaines perspectives de travail, l'objectif serait d'améliorer l'ensemble de 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 de feu du circulation (la lumière rouge ou la lumière verte), et avertir le conducteur.
- Essayer d'utiliser une caméra stéréoscopie pour estimer la distance entre les caméras et l'objet détecté.
- Faire une amélioration au niveau d'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. Nourria Keddar, Leila Ahmed Balkacem. « Détection et Reconnaissance Des Panneaux de signalisation Routière ». Thèse de doctorat. Centre Universitaire Belhadj Bouchaib d'Aïn-Témouchent, 2019.
2. MEDJAOUI Amina, FARES Fadia.« Segmentation des Images par Contours Actifs : Application sur les Images Satellitaires à Haute Résolutions ». Mémoire de master. L'université Abou Bakr Belkaid– Tlemcen Faculté des sciences : Département d'Informatique, 2012.
3. Moustafa BENALI. « Reconnaissance Automatique des Chiffres Manuscrits ». Thèse de Doctorat. Université Abou Bakr Belkaid– Tlemcen, 2017.
4. BOUCETTA ALDJIA. « Etude de l'effet des Transformées de Décorrélation en Compression des Images Couleurs RGB ». Thèse de Magister. L'université de Batna .2010.
5. Belmerabet Sarra & Bardjak Nawal. « Segmentation d'image ». Mémoire de master. Université Larbi Ben M'hidi Oum El Bouaghi, 2017.
6. Lamri Laouamer. « APPROCHE EXPLORATOIRE SUR LA CLASSIFICATION APPLIQUÉE AUX IMAGES ». thèse de doctorat. L'université de quebec. 2006.
7. BERKAT Yaakoub , TAHIAT Abderrahim. « Détection de voie et reconnaissance des objets dans les systèmes de transport intelligents ». Mémoire du Licence. Ecole Nationale Supérieure de Technologie, 2018.
8. Khadraoui Djihene. « Analyse des expressions faciales pour la détection de fatigue ». Mémoire de master. L'université de biskra,2019.
9. Barkat Ishak. « Character Recognition In The Image Using Deep Learning». Mémoire de Master .L'université de Biskra.2019.
10. Tyler Elliot Bettilyon, «Introduction To Deep Learning» .Disponible sur : https://medium.com/tebs-lab/introduction-to-deep-learning. Publié le le 13 juin 2018, consultée le 1 février 2020.
11. Stefan Duffner and Christophe Garcia. « Robust face alignment using Convulational Neural Network». 2008. Orange Labs France .
12. A. K. Justin Johnson, CS231n Convolutional Neural Networks for Visual Recognition, page web, http://cs231n.github.io/convolutional-networks/, publié en 2018, consultée le 10 février 2020.
13. 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 1 mars 2020.
14. Mohamed Zakari. « Classification des images avec les réseaux de neurones ». Mémoire de master. l'université de telemcen. 2017.
15. Shadman Sakib, et al.« An Overview of Convolutional Neural Network: Its Architecture and Applications ». Novembre 2018. Independent University of Bangladesh, Bangladesh.
16. Siddhartha Sankar Nath ,al. «A Survey of Image Classification Methods and Techniques. ».2014. International Conference on Control, Instrumentation, Communication and Computational Technologies ».
17. 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 2020.

18. 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 2020.
19. H. Bay, T. Tuytelaars, et L. Van Goo, « SURF: Speeded Up Robust Features. » ,2006.
20. Darshan Adakane , « 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 novombre 2019, consultée le 17 mars 2020.
21. Jose Luis Masache Narvaez. « Adaptation of a Deep Learning Algorithm for Traffic Sign Detection ». Mémoire de master. Canada : l'université du Western Ontario, 2019.
22. Kang S., Byun H., Lee SW. « Real-Time Pedestrian Detection Using Support Vector Machines. ».2002. vol 2388.
23. Richa Agrawal , « Person Detection in Various Posture using HOG Feature And SVM Classifier. » ,Disponible sur : https://medium.com/@richa.agrawal228/person-detection-in-various-posture-using-hog-feature-and-svm-classifier, publié le 31 Octobre 2018, consultée le 20 mars 2020.
24. Ren, Shaoqing, et al. « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks ». Janvier 2017.IEEE Transactions on Pattern Analysis and Machine Intelligence.
25. He, Kaiming, et al. « Mask R-CNN. ». Septembre 2017. IEEE International Conference on Computer Vision (ICCV).
26. Redmon, Joseph, et al. « You Only Look Once: Unified, Real-Time Object Detection. ».2016. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
27. Liu, Wei, et al. « SSD: Single Shot MultiBox Detector. ».2016. Computer Vision – ECCV. pp. 21–37.
28. Gauen, Kent, et al. « Comparison of Visual Datasets for Machine Learning. ».2017. IEEE International Conference on Information Reuse and Integration (IRI).
29. Hua, Xia, et al. « Military Object Real-Time Detection Technology Combined with Visual Salience and Psychology. ».Janvier 2020. La conference d' électronique en la chine.
30. Gao, Junfeng, et al. « Computer Vision in Healthcare Applications. ».2018. Journal of Healthcare Engineering.
31. Safadinho, David, et al. « UAV Landing Using Computer Vision Techniques for Human Detection. ». Janvier 2020. Journal of Sensors Engineering.
32. Janai, Joel, et al.« Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. ».2020. Foundations and Trends in Computer Graphics and Vision.
33. Lu, Meng, et al. « Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety ». Juin 2005.Transportation Planning and Technology, pp. 167-187.
34. 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.
35. 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.
36. FOAHOM GOUABOU Arthur Cartel, NGWA Leslie NGWA, mémoire Master, « Conception d'un système d'aide à la conduite pour véhicule tourisme(Anticollision)», l'université de Douala, guénie, 2017.
37. Kreuzig, Robin,et al. « DistanceNet: Estimating Traveled Distance from Monocular Images using a Recurrent Convolutional Neural Network », 2019.
38. Abdul Haseeb Muhammad, et al. « Multi-DisNet: Machine Learning-Based Object Distance Estimation from Multiple Cameras. ».2019. Lecture Notes in Computer Science, , pp. 457– 469.

39. 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 2020.

40. 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. consultee le en 05 avril 2020.

41. Szarvas, M., et al. « Pedestrian Detection with Convolutional Neural Networks ».2002. IEEE Proceedings. Intelligent Vehicles Symposium.

42. Cai, Zhaowei, et al. « Learning Complexity-Aware Cascades for Pedestrian Detection. ».2002. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 9, 1, pp. 2195–2211.

43. Wang, Chunxiang, et al. « Automatic Parking Based on a Bird's Eye View Vision System ». Janvier 2014.. Advances in Mechanical Engineering, vol. 6.

44. Eyal Katz, et al. «Automatic Parking Identification and Vehicle Guidance with Road Awareness. ». Novembre 2016. IEEE International Conference on the Science of Electrical Engineering (ICSEE).

45. Liu, Kaizhan, et al. « A Real-Time Method to Estimate Speed of Object Based on Object Detection and Optical Flow Calculation. ». Avril 2018. Journal of Physics: Conference Series, vol. 1004.

46. 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.

47. de Charette, Raoul, and Fawzi Nashashibi. « Traffic Light Recognition Using Image Processing Compared to Learning Processes. ». Octobore 2009.IEEE/RSJ International Conference on Intelligent Robots and Systems.

48. Lee, Gwang-Gook, and Byung Kwan Park. « Traffic Light Recognition Using Deep Neural Networks. ». juin 2017. IEEE International Conference on Consumer Electronics (ICCE).

49. Byeonghak Lim ,et al. « Integration of Vehicle Detection and Distance Estimation using Stereo Vision for Real-Time AEB System ».2017. In Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems. les pages 211-216.

50. Too, Edna Chebet, et al. « A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification. ». Mars 2018. Computers and Electronics in Agriculture.

51. M. Hansard, et al. « Time-of-Flight Cameras: Principles,Methods and Applications. ». 2012. Springer Science & Business Media.

52. Hoi-Kok Cheung, et al. « Accurate Distance Estimation Using Camera Orientation Compensation Technique for Vehicle Driver Assistance System ».2012. International Conference on Consumer Electronics.

53. J. Hui, « SSD object detection: Single Shot MultiBox Detector for real-time processing », Disponible sur : https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multiboxdetector-for-real-time-processing-9bd8deac0e06. Publiée le 14 mars 2018, consultées le 21 juin 2020

54. Zeiler, Matthew D. et Rob Fergus. "Visualiser et comprendre les réseaux convolutifs." Dans Conférence européenne sur la vision par ordinateur, ,2014, pp. 818-833.

55. Eddie Forson, « Understanding SSD MultiBox—Real-Time Object Detection In Deep Learning », Disponible sur : https://towardsdatascience.com/understanding-ssd-multibox-real-time-objectdetection-in-deep-learning-495ef744fab. Publiée le 18 octobre 2017, consultée le 1 aout 2020.

56. Too, Edna Chebet, et al. « A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification. », Computers and Electronics in Agriculture, Mars 2018.

57. Paul pias, « Object detection and distance measurement», Disponible sur : https://github.com/paul-pias/Object-Detection-and-Distance-Measurement. Publiée en 28 juillets 2020, consultées en 14 septembre 2020.

58. Pierluigiferrari, Disponible sur : https://drive.google.com/file/. Publiée le 7 mai 2020, consultée le 5 septembre 2020.

Résumé

Dans ces dernières années, le développement du domaine de la vision par ordinateur a poussé les industries automobiles de créé des systèmes d'aide à la conduite, qui peuvent permettre de réduire les accidents de la route d'une manière significative. Dans ce mémoire, nous avons proposé un système de détection des objets de la circulation routière et d'estimer leur distance par rapport à la caméra. Notre système localise et reconnait 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'existant des feux de la circulation et les panneaux STOP. Puis 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 monde.

Les mots clé : La vision par ordinateur, la détection des objets, l'estimation de distance, les systèmes d'aide à la conduite.

Abstract

In recent years, the development of the field of computer vision helps the vehicle industry to create driver assistance system (ADAS) that helps to reduce the number of accidents around the world. In this dissertation, we used a system to detect objects in road traffic and estimate its distance from the camera. The system localizes and recognizes objects (cars, trucks, pedestrians, etc.) located in the vehicle. In order to accurately determine the nature of each object, a deep learning algorithm has been used; it also warns the driver of the existing traffic lights and STOP signs around the vehicle. Then, it estimates the distance between the integrated camera and the detected object. The combination of these different pieces of information allows the driver assistance system to better perceive the world.

Keyword: Computer vision, object detection, distance estimation, driver assistance system.