RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MHAMED BOUGARA BOUMERDES

FACULTÉ DES SCIENCES DE L'INGÉNIEUR
DÉPARTEMENTINGÉNIERIE DES SYSTÈMES ÉLECTRIQUES

MÉMOIRE DE FIN D'ÉTUDE POUR L'OBTENTION DU DIPLÔME DE MASTER
SPÉCIALITÉ RÉSEAUX ET TÉLÉCOMMUNICATIONS

THÈME

Mise au point d'un système intelligent pour la détection et la mesure de vitesse des véhicules

Réalisé par :

- STITI Saad Abdeslam
- HAMIDOUCHE Mohamed Abdelouafi

Promoteur : Dr.RIAHLA Mohamed Amine

Encadreurs

- Dr.HIMEUR Yassine
- Dr.AMOURI Bilal

Jury :

-Dr.MECHID Samira

-Dr.MERAIHI Yacine

2018/2019

Remerciements

Au terme de ce modeste travail, nous remercions Dieu le tout Puissant et le tout Miséricordieux de nous avoir donné tout au long de ce parcours, le courage, l'abnégation, la santé et la patience nécessaire pour l'accomplissement et la finalisation de ce mémoire de fin d'études de Master.

Nos vifs remerciements vont également :

À nos encadreurs Yassine Himeur et Bilal Amouri qui nous ont permis de travailler avec aux. Au demeurant, on a eu a apprécié vos qualités et vos valeurs, nonobstant vos compétences. Merci pour votre précieux aide et de nous avoir guidée à chaque étape de la réalisation de ce travail. Votre sens du devoir nous a énormément marqués, Veuillez trouver l'expression de notre grand respect et notre parfaite gratitude.

À notre promoteur RIAHLA Mohamed Amine et un remerciement particulier, pour votre précieuse aide, Nous saisissons cette occasion pour vous exprimer notre profonde gratitude et notre reconnaissance tout en vous témoignant notre grand respect.

Aux membres du jury qui nous honorent de leur présence en participant au jury de ce mémoire.

À tous nos enseignants qui nous ont formés et cela de l'école primaire jusqu'à l'université.
Que tous ceux qui nous ont aidés, de près ou de loin, à mener à bout ce travail, trouvent ici l'expression de notre reconnaissance et notre profonde gratitude.

Nous dédions ce travail à nos chers parents et frères, à nos familles et nos amis.

Merci à Tous

TABLE DES MATIERES

Table des Figures

INTRODUCTION GENERALE

Au cours des dernières années, le parc national automobile a connu une croissance très élevée, aujourd'hui, plus de 6 millions de véhicules circulent en Algérie [1]. Cependant, avec cette croissance le risque d'avoir des accidents routiers devient plus grand. De cela la recherche d'une solution pour diminuer ces risques devient nécessaire.

La surveillance du trafic routier fournie des informations précises en temps réel des véhicules, diverses sources peuvent être mises en place comme les détecteurs de boucle, les capteurs radar ou les caméras vidéo.

Les développements récents technologiques en vision par ordinateur et traitement des images ont révélé que les caméras vidéo sont un moyen efficace de collecte et d'analyse des données du trafic [2].

Les systèmes de surveillance vidéo sont plus sophistiqués et robustes parce que les informations associées aux séquences d'images présentées dans une vidéo nous permettent d'identifier et classer les véhicules d'une manière plus efficace.

L'objectif principal de notre travail et de mettre en place un système intelligent capable de détecter tout type de véhicules, fournir des informations sur ces derniers (couleur, types…etc.), puis mesurer leurs vitesses, et enfin identifier ceux qui ont été en excès de vitesse (en récupérant leurs plaques d'immatriculation).

Pour cela, notre mémoire est organisé comme suit :

- Dans le premier chapitre nous aborderons les types, les caractéristiques, et les différents formats de l'image et de la vidéo, ainsi que leurs traitements. Ensuite, nous décrirons les différents types du système de vidéo surveillance.
- Dans le second, nous présenterons l'état de l'art de la détection d'objets et les techniques sur les quelles s'appuie cette détection (segmentation des régions et détection des contours).
- Nous consacrerons le troisième chapitre pour la détection de véhicules en développant la contribution des chercheurs dans cette partie, le suivi (tracking), ainsi que les techniques de classification et enfin, nous divulguerons la technique que nous avons choisie dans le but d'atteindre notre objectif.
- Dans le dernier chapitre nous dénoncerons les outils, les logiciels et la méthode d'implémentations sur le matériel utilisé. Et pour finaliser, nous détaillerons les scripts et applications des parties de notre système.

I. CHAPITRE I

Notion de base sur le traitement d'images et vidéos

I.1. Introduction

Le traitement d'image est la discipline qui étudie l'amélioration et la transformation des images numériques, et qui a connu un développement important depuis quelques dizaines d'années. [3]
Cette dernière consiste à appliquer des transformations mathématiques sur des images qui nous permettront par la suite d'améliorer leurs qualités ou d'en extraire des informations issues du monde réel.
Ainsi, puisque la vidéo est par définition une succession d'images, alors le traitement de la vidéo n'est rien d'autre que le traitement des images de cette vidéo (autrement dites frames) indépendamment.
Dans ce chapitre, nous allons parler brièvement sur les notions de base nécessaires à la compréhension des techniques de traitement d'images, et de la vidéo. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

I.2. Représentation de l'image

L'image est la reproduction visuelle d'un objet réel, définit comme étant une représentation bidimensionnelle construite à partir d'une matrice binaire, chaque élément de cette matrice (appelé pixel) est codé selon le type d'image. [4]

I.2.1. Types d'images

On distingue trois types d'images :

- Image en noir et blanc :
  Un seul bit suffit pour coder l'information, par exemple 0 pour le noir et 1 pour le blanc. [5]
- Image en niveaux de gris :
  On utilise 8 bits, ce qui donne 256 nuances de gris possibles pour le pixel, de 0 (noir) à 255 (blanc). [6]

- Image en couleurs :

  La couleur de chaque pixel est définie par 3 composantes : Rouge, Vert et Bleu (système RVB où RGB en anglais). L'intensité de chaque composante est codée sur 8 bits, donc chaque composante a une valeur comprise entre 0 (absence de couleur) et 255 (intensité maximale de la couleur). Ainsi la couleur d'un pixel nécessite 24 bits (3 octets) pour être codée. [6]

  La couleur du pixel est obtenue par synthèse additive, en particulier :

  - si les 3 composantes sont à 0, on obtient du noir,
  - si les 3composantes sont identiques on obtient une nuance de gris,
  - si les 3 composantes sont à 255, on obtient le blanc.

I.3. Caractéristiques de l'image

I.3.1. Pixel

Une image est constituée d'un nombre fini de points appelés pixels qui représentent les plus petits éléments constitutifs d'une image (contraction des mots anglais "picture element", c'est à dire élément d'image).

Ces pixels ont une forme rectangulaire, et chacun d'entre eux possède une couleur quelconque, et en rassemblant cespixels on obtient notre image.

La taille du pixel définit la résolution par rapport à l'image analogique originale, c'est-à-dire la finesse de la grille. Plus la résolution baisse, plus le nombre de pixels dans l'image diminue, et plus la qualité de l'image numérique se dégrade. (Voir figure 1)

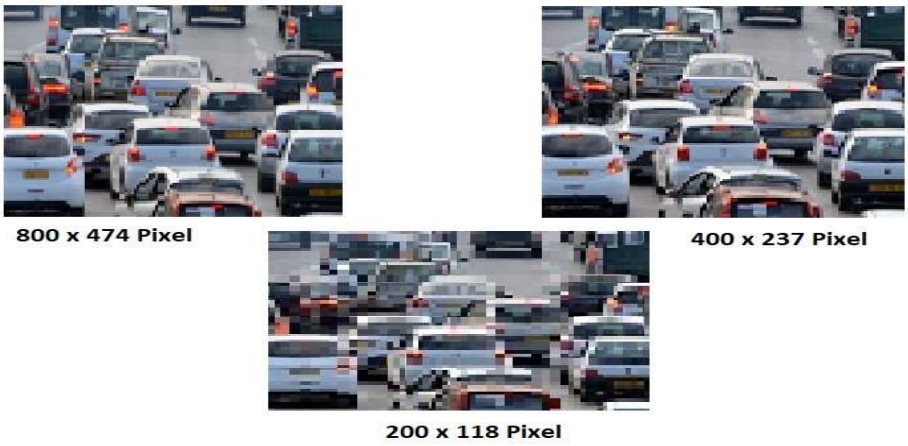


Figure 1: La relation entre la résolution et le nombre des pixels d'une image.

I.3.2. Poids de l'image

C'est la taille de l'image. Exprimé en en octets, le poids de l'image est le résultat de la multiplication du nombre de pixels fois le poids de chaque pixel.

Par exemple : si on a une image d'une résolution de 640x480 en couleur, sachant que le poids de chaque pixel = 3 octets, alors :

- ✓ Nombre de pixel = 640x480 = 307200
- ✓ Le poids de l'image = 307200x3 = 921600 octets = 900 Ko.

I.3.3. Bruit

le bruit sur une image n'a bien évidemment rien d'audible. C'est un phénomène causé par un éclairage insuffisant lors de la prise de vue. Il se caractérise par l'apparition de grains colorés particulièrement visibles dans les zones sombres**.** [7]

I.3.4. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie ainsi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface.La luminance s'exprime en candela par mètre carré (cd/m^2).

Ce terme est souvent utilisé en transmission vidéo pour désigner le signal qui permet la reproduction d'une image noir et blanc. [8]

I.3.5. Transparence

La transparence est une caractéristique définissant le niveau d'opacité des éléments de l'image, c'est la possibilité de voir à travers l'image des éléments graphiques situés derrière celle-ci.

I.3.6. Contraste

Est une propriété intrinsèque d'une image qui désigne et quantifie la différence entre les parties claires et foncées d'une image (elle différencie les couleurs claires des couleurs foncées).

En photographie on le définit le contraste comme la différence entre la densité la plus forte et la plus faible d'une image. Le contrôle du contraste est un élément important de la pratique photographique. Le contraste final de l'image dépend à la fois du sujet, de la nature et du traitement du négatif et du positif.

I.3.7. Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image. Il fournit diverses informations comme les statistiques d'ordre (moyenne, variance,...), l'entropie, et peut permettre d'isoler des objets. [9]

I.3.8. Définition

On appelle "définition" le nombre de pixel constituant l'image, c'est-à-dire sa «dimension informatique» (le nombre de colonnes de l'image que multiplie son nombre de lignes). Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée 640x480.

I.3.9. Résolution

La résolution d'une image est le nombre de pixels par pouce qu'elle contient (1 pouce = 2.54 centimètres). Elle est exprimée en "PPP" (points par pouce) ou DPI (dots per inch). Plus il y a de pixels (ou points) par pouce et plus il y aura d'information dans l'image (plus précise). Par exemple, une résolution de 300dpi signifie que l'image comporte 300 pixels dans sa largeur et 300 pixels dans sa hauteur.

I.4. Différentes formats de l'image

Classées en deux types de format : [10]

I.4.1. Format vectorielle

Dans une image vectorielle les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique (donc pas de pixels).

Par exemple, un cercle est décrit par une information du type (cercle, position du centre, rayon), un rectangle est définie par deux points, une courbe par plusieurs points et une équation. Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans.

L'intérêt avec ce typed'images c'est qu'on peut les agrandir sans perte ou dégradation de qualité.

I.4.2. Format matricielle

Une image matricielle est formée d'un tableau de points ou pixels. Plus la densité des points sont élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée. Corrélativement la place occupée en mémoire et la durée de traitement seront d'autant plus grandes.

Les images vues sur un écran de télévision ou une photographie sont des images matricielles.

On obtient également des images matricielles à l'aide d'un appareil photo numérique,d'une caméra vidéo numérique ou d'un scanner. Parmi ces formats on peut citer :

- BMP (BitMap) : Le format BMP est le format par défaut du logiciel Windows.C'est un format matriciel. Les images ne sont pas compressées. Son logiciel d'origine.

- Le format EPS : matriciel n'est pas très différent du EPS vectoriel. En fait seules les données contenues dans le fichier sont différentes. Ainsi un logiciel de retouche de photos tel que Photoshop permet l'importation, la modification et l'exportation de fichiers en format EPS.

- GIF (GraphicalInterchange Format) : Le format GIF est un format qui a ouvert la voie à l'image sur le World Wide Web. C'est un format de compression qui n'accepte que les images en couleurs indexés codé sur 8 bits, C'est un format qui perd beaucoup de son marché suite à une bataille juridique concernant les droits d'utilisation sur Internet.

- JPEG (Joint Photographique Experts Group) : Les images JPEG sont des images de 24 bits. C'est-à dire qu'elles peuvent afficher un spectre de 16 millions de couleurs. C'est la meilleure qualité d'images disponible.

I.5. Régions d’intérêt

Une région d'intérêt (en anglais: Region Of interest «ROI») est un sous-ensemble d'une image ou d'un jeu de données identifié dans un but particulier. Le jeu de données peut être l'un des éléments suivants:

- Jeu de données waveform ou 1D: la ROI est un intervalle de temps ou de fréquence sur le signal (graphique d'une quantité représentée par rapport au temps).
- Image ou jeu de données 2D: la ROI est définie par des limites données sur une image d'un objet ou sur un dessin.
- Volume ou jeu de données 3D: la ROI correspond aux contours ou aux surfaces définissant un objet physique.
- Jeu de données Time-Volume ou 4D: En ce qui concerne le jeu de données 3D changeant d'un objet dont la forme change avec le temps, la ROI est le jeu de données 3D au cours d'une période ou d'une période donnée.

En vision par ordinateur et en reconnaissance optique de caractères, le ROI définit les limites d'un objet considéré. Définie par une boîte rectangulaire située à l’intérieur d’une trame.

I.6. Seuillage

Le seuillage d'image est la méthode la plus simple de segmentation d'image. À partir d'une image en niveau de gris, le seuillage d'image peut être utilisé pour créer une image comportant uniquement deux valeurs, noir ou blanc (monochrome)1.

Le seuillage d'image remplace un à un les pixels d'une image à l'aide d'une valeur seuil fixée. Ainsi, si un pixel à une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir).

La valeur du seuil peut être déterminée manuellement ou bien automatiquement à partir de l'histogramme. [11]

I.7. Notion de la vidéo

I.7.1. Définition de la vidéo

Le mot vidéo vient du latin vidéo qui signifie «je vois». La vidéo est une succession d'images animées défilant à une certaine cadence afin de créer une illusion de mouvement pour l'œil humain qui peut distinguer environ 20 images par secondes.

D'autre part la vidéo au sens multimédia du terme est généralement accompagnée de son, c'est-à-dire de données audio.

I.7.2. Types de vidéo

Il existe deux grandes famille des systèmes vidéo: Ceux qui sont analogiques et autres qui sont numériques

- **La vidéo analogique**

La vidéo analogique représentant l'information comme un flux continu de données analogiques, destiné à être affiché sur un écran de télévision basé sur le principe du balayage (ligne par ligne, en commençant par le haut à gauche pour finir en bas à droite). Afin d'assurer une uniformité dans la luminosité de cet affichage, en d'autres termes afin que l'intensité lumineuse des pixels du haut de l'écran ne commence pas à diminuer alors que ceux du bas viennent d'être activés, une technique spécifique a été développée. Il s'agit de la technique d'entrelacement Un premier balayage est effectué sur les lignes impaires, puis un deuxième sur les lignes paires.[12]

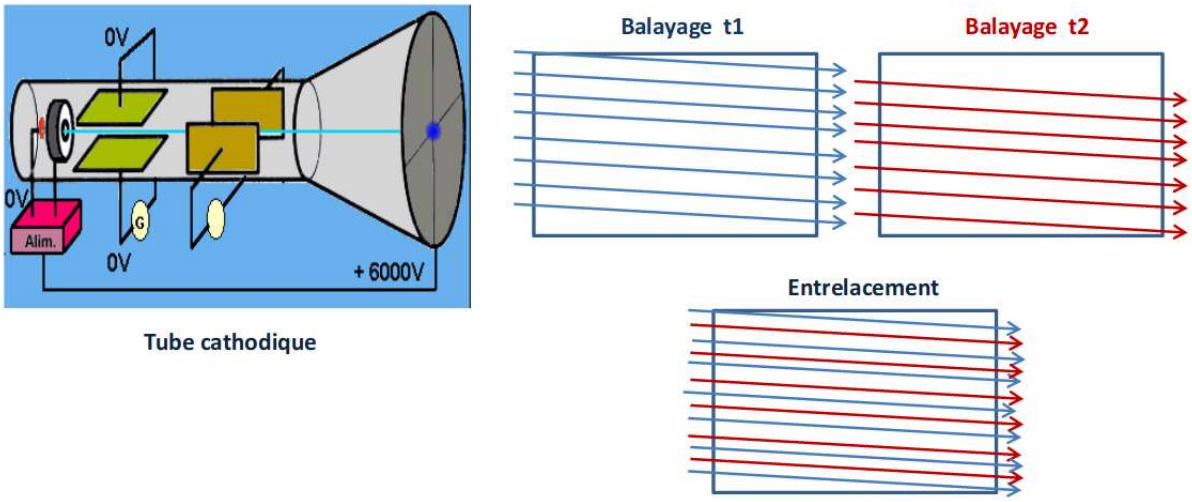


Figure 2: Technique d'entrelacement.

Il existe plusieurs normes pour la vidéo analogique. Les trois principales sont : PAL, SECAM et NTSC.

- **PAL/SECAM**: (*Phase Alternating Line*/*Séquentiel Couleur avec Mémoire*), utilisée en Europe pour la télévision hertzienne, permet de coder les vidéos sur 625 lignes (576 seulement sont affichées car 8% des lignes servent à la synchronisation). à raison de 25 images par seconde à un format 4:3 (c'est-à-dire que le rapport largeur sur hauteur vaut 4/3).
- **NTSC**: La norme *NTSC* (*National Television Standards Committee*), utilisée aux Etats-Unis et au Japon, utilise un système de 525 lignes entrelacées à 30 images/sec (donc à une fréquence de 60Hz). Comme dans le cas du PAL/SECAM, 8% des lignes servent à synchroniser le récepteur. Ainsi, étant donné que le NTSC affiche un format d'image 4:3, la résolution réellement affichée est de 640x480.

- **La vidéo numérique**

La vidéo numérique consiste à afficher une succession d'images numériques à une certaine cadence. C'est un fichier qui contient les flux vidéos multiples, le son et le texte (méta données), les sous-titres, l'information de chapitre, et l'information de synchronisation pour restituer ces différents flux placés dans un conteneur.

I.7.3. Composition d'un fichier vidéo

Un fichier vidéo se compose généralement de 2 éléments :

- **Le conteneur**: Il correspond généralement au format du fichier. Son rôle est de rassembler et d'organiser dans un fichier, différents types de données (flux audio, vidéos, sous-titres, meta-données). Chaque conteneur possède ses spécificités en termes de nombre de pistes acceptées pour la vidéo et l'audio des codecs reconnus.

Il existe plusieurs conteneurs on distingue : AVI (Audio Video Interleave), MP4, MKV (Matroska), WMV (Windows Media Video).

- **Le contenu**: Ils se composent essentiellement de flux audios et/ou vidéos. Ceux-ci sont généralement compressés à l'aide d'un codec (algorithme de compression/décompression) comme le Divx, le H264, le mp3.

I.7.4. Formats d'un fichier vidéo

Décrit l'ordre et la structure de ces images. Les données du flux vidéo, qui peuvent être accompagnées de sons sous la forme de flux audio, sont très volumineuses : elles doivent impérativement être compressées (codées) à l'aide d'un **codec** (qui est un algorithme de compression / décompression d'un signal audiovisuel numérique.) pour être stockées ou/et transmises (et donc être adaptées au débit des réseaux). [13]

Les flux vidéo (et les flux audio éventuellement associés), une fois encodés, sont généralement encapsulés dans des fichiers conteneurs : ces derniers permettent, notamment, leur lecture simultanée.

Il existe de nombreux codecs permettant de compresser et de décompresser une vidéo dans son conteneur.

- MJPEG (Motion JPEG): Qui compresse la vidéo image par image, en utilisant la technologie JPEG appliquée à l'image fixe, et réunit ces images en mouvement et le son dans un même format de fichier. C'est le codec le plus utilisé pour les captures vidéo des ensembles cartes d'acquisition et logiciels d'édition vidéo. La conservation d'une bonne qualité d'image produit toutefois de gros fichiers.

- MPEG (Moving Picture Experts Group): Les formats MPEG sont des formats de compression avec pertes pour les séquences vidéo.

- DivX (Digital Video Express) : Codec vidéo propriétaire et fermé proposé par DivX Inc., conçu à partir de MPEG-4 part 2, ce dernier ayant étant modifié afin d'y ajouter la possibilité de compresser le son au format MP3. Cela permet ainsi d'obtenir des vidéos compressées très peu volumineuses avec une perte de qualité raisonnable. Ainsi le format DivX permet de stocker un film complet de plusieurs heures sur un CD-ROM de 650 ou 700 Mo.

I.7.5. Représentation d'une séquence vidéo

Une séquence vidéo brute est une suite d'images fixes, qui peut être caractérisée par trois principaux paramètres :

- **La résolution en luminance**: Détermine le nombre de nuances ou de couleurs possibles pour un pixel. Celle-ci est généralement de 8 bits pour les niveaux de gris et de 24 bits pour les séquences en couleurs.

- **La résolution spatiale**: Définit le nombre de lignes et de colonnes de la matrice de pixels.

- **La résolution temporelle**: Est le nombre d'images par seconde. La valeur de ces trois paramètres détermine l'espace mémoire nécessaire pour stocker chaque image de la séquence. Cet espace mémoire est caractérisé par le débit, qui est le coût de stockage pour une seconde (capacité mémoire nécessaire pour stocker une seconde de vidéo).

I.8. La vidéosurveillance

La vidéosurveillance consiste à placer des caméras de surveillance dans un lieu public ou privé et de recevoir le flux vidéo sur un PC localement ou à distance en vue d'augmenter le niveau de sécurité. Les causes de l'installation de systèmes de vidéosurveillance sont diverses, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments phares dans la justification de la vidéosurveillance.

I.8.1. Types de caméras

Le choix très vaste de caméras réseau que l'on trouve à l'heure actuelle permet de répondre aux besoins de tous les secteurs, quelle que soit la taille du système requis. Tout comme les caméras analogiques, les caméras réseaux se déclinent en différents modèles, telles-que:

- Fixe: Pointée dans une direction unique, elle couvre une zone définie.
- PTZ (Pan Tilt Zoom): Motorisée, elle peut être actionnée, manuellement ou automatiquement, dans des mouvements panoramique/inclinaison/zoom.
- Dôme: Recouverte d'un caisson hémisphérique, ce qui la rend discrète et, dans certains modèles, résistante au vandalisme et aux intempéries. Elle peut être fixe ou mobile.
- Mégapixel: Offre une résolution plus élevée que les caméras standards, allant de 1 à 16 mégapixels17. Elle permet soit de capter une image plus détaillée, soit de couvrir un plus large champ visuel, réduisant le nombre de caméras nécessaires pour couvrir une aire à surveiller.

- Infrarouge et thermique: Sensible au rayonnement infrarouge (IR), elle est capable de produire une image de bonne qualité dans le noir pour une surveillance nocturne.
- Panoramique: Grâce à une optique spéciale, elle offre 360° de visibilité avec une seule caméra.

I.8.2. Les systèmes de vidéosurveillance

- **Système de vidéosurveillance analogique**

A leur début les systèmes de vidéosurveillance étaient entièrement analogiques c.à.d. que la transmission se faisait comme celle des signaux de téléphoniques.(figure 3)

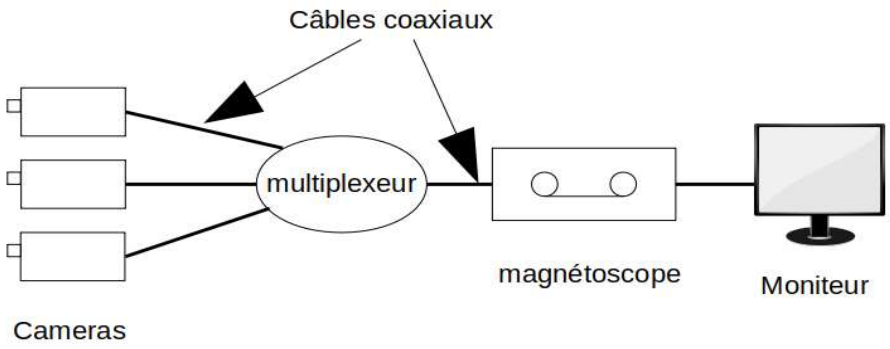


Figure 3: Schéma représentant un système de vidéosurveillance analogique.

- **La vidéosurveillance sur IP**

La vidéo sur IP ou la IP-Surveillance, un réseau qui utilise les outils et les protocoles de l'internet, se distingue des systèmes de vidéo surveillance analogiques en permettant aux utilisateurs de visualiser ou d'enregistrer des images vidéos via un réseau IP.

En notion de sécurité, la vidéosurveillance sur IP est plus sécurisée (que l'analogique) et peut être accessible à tout instant et en tout lieu afin d'obtenir des informations sur un fait qui passe en cours et le suivre en temps réel, en étant connecté au même réseau informatique et en ayant le droit ou l'autorisation d'y accéder.

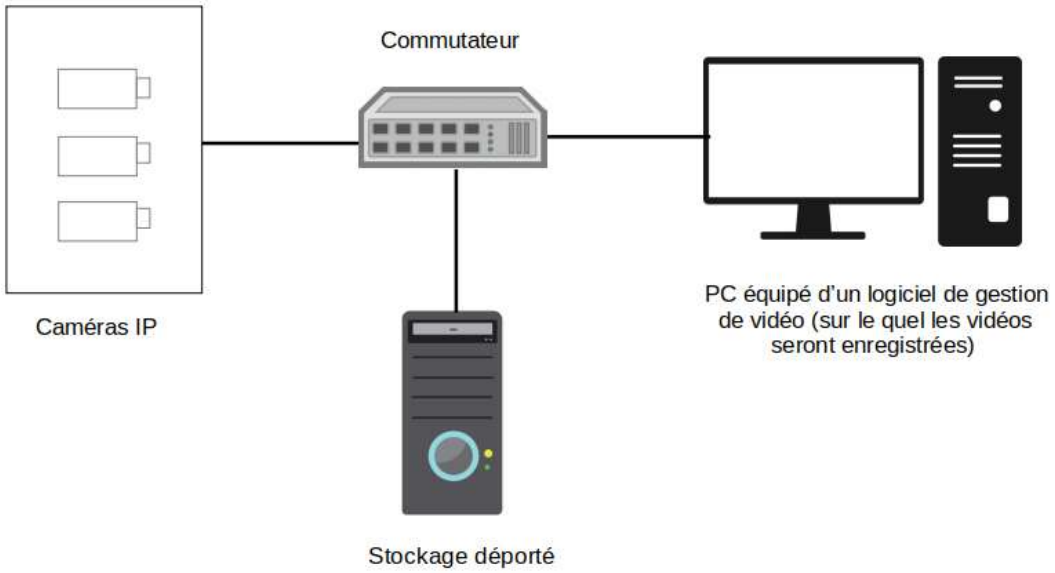


Figure 4: système représentant un système de vidéo surveillance sur IP.

- **Les systèmes analogiques/IP (hybrides)**

Les systèmes analogiques sont préférés pour leurs coûts plus compétitifs, tandis que les systèmes IP proposent une quantité d'image importante à stocker grâce a leurs systèmes d'enregistrement numérique.

Les systèmes hybrides intègrent à la fois les deux systèmes, afin de rendre les systèmes analogiques ouverts à l'extérieur en utilisant des serveurs vidéo.

Ces serveurs vidéo permettent aux utilisateurs de bénéficier des avantages des systèmes de vidéo surveillance sur IP tout en conservant les installations analogiques a part certains équipements spécifiques comme les câbles coaxiaux et les moniteurs

Un serveur vidéo possède un serveur web intégré, une puce de compression et un système d'exploitation permettant la conversion des flux entrants en images vidéo numériques, ainsi que leur transmission et leur enregistrement sur le réseau informatique où elles pourront être visualisées et consultées plus facilement.

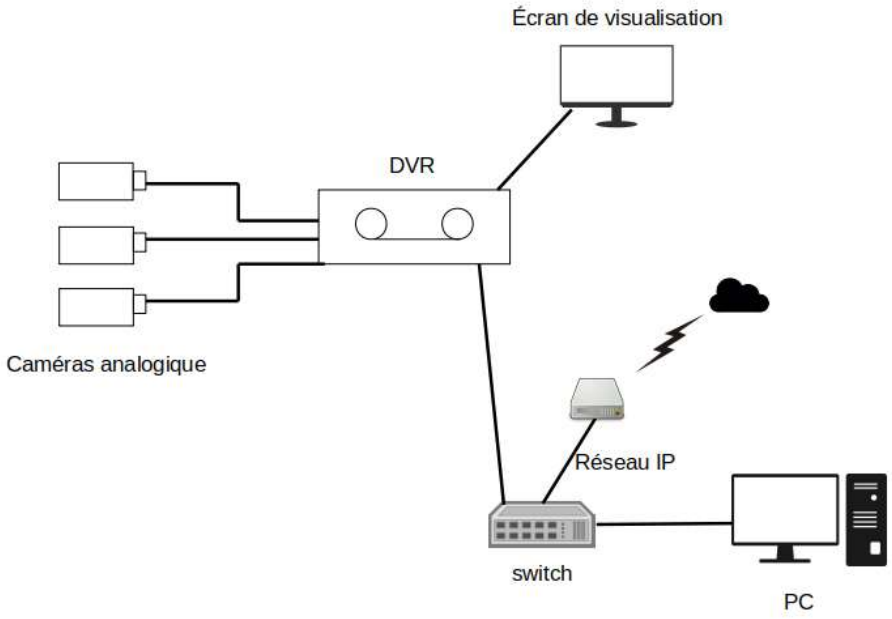


Figure 5: Système représentant un système de vidéosurveillance hybride.

I.9. Conclusion

Ce chapitre a été consacré à la présentation des notions de bases sur les images et les vidéos dans le but d'extraire des informations en se basant sur un système de vidéosurveillance. L'objectif de notre travail est la détection et suivi de véhicules en mouvement dans une vidéo, les prochains chapitres sont consacrés à la présentation des différentes méthodes de la détection de ces derniers.

II. CHAPITRE II

États de l'art de la détection d'objet

II.1. Introduction

La détection d'objets en mouvement est la première étape d'un système de vidéosurveillance. Elle permet l'extraction des objets mobiles présents sur les images de la séquence vidéo en les séparant de l'arrière-plan. C'est une étape de traitement de bas niveau, critique et difficile car elle doit être robuste aux changements dynamiques de la scène.

Dans ce chapitre nous allons définir la détection d'objet et présenter les techniques sur la quelle se base cette détection, afin de les utiliser pour la détection des véhicules qui est le but de notre projet de Master.

II.2. Détection d'objets

La détection d'objets est un domaine très actif de la recherche qui cherche a classer et localiser des régions/zones d'une image ou d'un flux vidéo. Elle facilite l'estimation de la pose, la détection des véhicules, la surveillance, etc.

En effet, le principe de la détection d'objets est le suivant: pour une image donnée, on recherche les régions de celle-ci qui pourrait contenir un objet puis pour chacune de ces régions découvertes, on l'extrait et on la classe à l'aide d'un modèle de classification d'image.

II.2.1. Techniques de détection des contours

La détection de contours est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes. Elle est très utilisée comme étape de prétraitement pour la détection d'objets, pour trouver les limites de régions. En effet, un objet peut être localisé à partir de l'ensemble des pixels de son contour. De plus, trouver cet ensemble permet d'obtenir une information sur la forme de l'objet. Du point de vue théorique, un contour est défini par un changement marqué de l'intensité d'un pixel à l'autre, autrement dit, une rupture d'intensité dans l'image suivant une direction donnée. Plusieurs méthodes existent pour détecter cette rupture, les unes plus ou moins complexes, les autres plus ou moins gourmandes en calculs. Dans la plupart des cas, Elle s'applique en deux étapes : la première permet de localiser les contours à partir d'un

calcul de Gradient ou de Laplacien dans des directions privilégiées. La seconde étape va permettre d'isoler les contours du reste de l'image à partir d'un seuillage judicieux.

Plusieurs méthodes permettent de déterminer le Gradient ou le Laplacien d'une image. Il en est de même des techniques de seuillage. L'application de détecteurs de contours sous la forme de filtres dérivateurs permet d'obtenir les contours des objets présents dans la scène.

Nous pouvons citer les approches se basant sur les différences finies comme l'opérateur de gradient, l'opérateur Laplacien, les filtres de Sobel, Prewitt, Roberts ou bien des approches reposant sur des critères d'optimalité comme les filtres de Canny-Deriche. Mais ce genre de techniques est peu exploitable car elles donnent des contours non fermés, bruités ou des contours non détectés.

- **Approches par convolution**: qui sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique (Première et deuxième dérivé). A chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité choisi. Le résultat est une image binaire constituée de points de contours et de points non-contours.

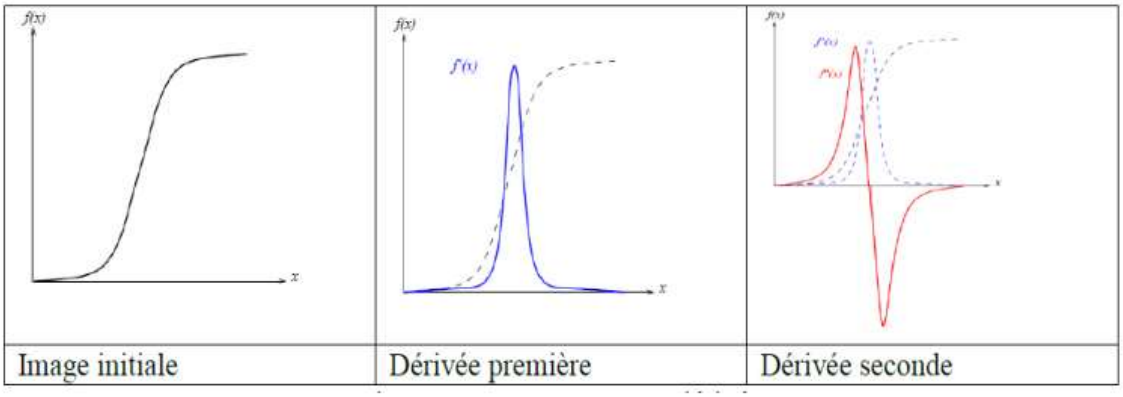


Figure 6: Contours et ses dérivées.

- Détection des contours par dérivée première:

Les filtres utilisés ici sont les filtres de dérivée première (appelés aussi filtres étroits) et l'on cherche alors le maximum de leur réponse. Leur prototype est le filtre de gradient mais la dérivation accentuant le bruit (pixels parasites de répartition aléatoire). Le gradient est une dérivation au premier ordre et est donné par la formule :

$$\nabla I(x,y) = \frac{\partial I}{\partial x}\hat{I}x + \frac{\partial I}{\partial y}\hat{I}y \qquad \text{..... Formule (1)}$$

Ix (Iy) est un vecteur unitaire suivant x (suivant y).

Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction. L'amplitude est directement liée à la quantité de variation locale des niveaux de gris. La direction du gradient est orthogonale à la frontière qui passe au point considéré. La méthode la plus simple pour estimer un gradient est donc de faire un calcul de variation monodimensionnelle c'est-à-dire en ayant choisi une direction donnée. Le gradient étant un vecteur, l'approche la plus classique pour estimer le gradient consiste à choisir deux directions privilégiées (naturellement celles associées au maillage, ie : ligne et colonne) orthogonales sur lesquelles on projette le gradient. On peut donc obtenir une connaissance parfaite du gradient de l'image qui se calcule comme suit :

$$\nabla x = \frac{\partial I(x,y)}{\partial x} \qquad \text{……Formule (2)}$$

$$\nabla y = \frac{\partial I(x,y)}{\partial y} \qquad \text{……Formule (3)}$$

Ainsi, en chaque point (x,y) de l'image, on définit deux dérivées partielles, suivant x et suivant y. La direction du vecteur gradient maximise la dérivée directionnelle et sa norme est la valeur de cette dérivée. Le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.

Il existe plusieurs opérateurs de gradient Parmi ses operateurs, il y a les masques de Roberts, de Prewitt et de Sobel ...etc.

a) Opérateurs de Sobel et Prewitt

Les opérateurs de'' Sobel'' et de ''Prewitt'' permettent d'estimer la norme du gradient bidimensionnel d'une image en niveau de gris. Ces opérateurs consistent en une paire de masques de convolution 3×3.

Pour ces opérateurs les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$\frac{\Delta I}{\Delta j} = h_j * I(i,j) \text{et} \frac{\Delta I}{\Delta J} = h_i * I(i,j) \quad \text{……Formule (4)}$$

Avec:

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \text{et} : h_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -c \end{bmatrix} \quad \text{…...……Formule (5)}$$

Les matrices h1 et h2 sont appelées masques, Les masques de Prewitt sont définis par c=1 et les masques de Sobel par c=2.

b) Opérateur de Roberts (1965)

Le détecteur de Roberts permet de calculer le gradient bidimensionnel d'une image de manière simple et rapide. Ce principe ne diffère pas beaucoup de celui des opérateurs de ''Prewitt ''et ''Sobel''.

Les masques de convolution de Robert sont :

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{et} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{……..Formule (6)}$$

Dans les figures suivantes nous allons voir les contours détectés par ces filtres:

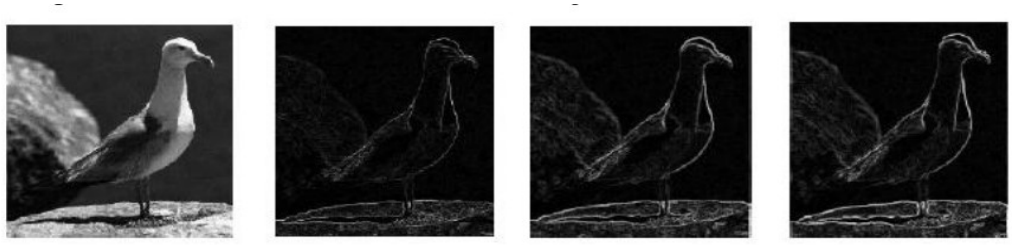
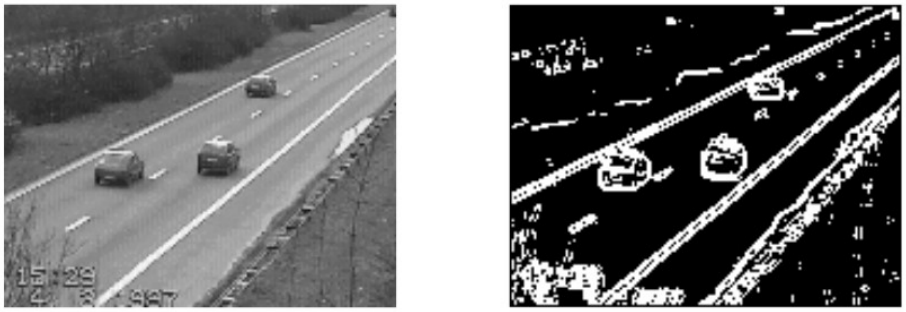


Figure 7: Contours détectés en utilisant les filtres de Roberts et PerwittSobel.[14]



(a) (b)

Figure 8: Exemple d'application du filtre Robert/PerwittSobel, (a) : image d'origine, (b) : contour de Robert/PerwittSobel.[15]

- Détection des contours par dérivée deuxième:

Les filtres larges que nous allons maintenant étudier recherchent les zéros de la dérivée seconde ou, plus précisément, du Laplacien qui est une dérivation au deuxième ordre, On décrit ici les opérateurs de type Laplacien définient comme étant :

$$\Delta I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \quad \text{…….Formule (7)}$$

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y-1) + f(x,y+1) - 4f(x,y)$$

Formule (8)

Les points de contours correspondent alors aux passages par zéro de l'image obtenue par convolution avec l'opérateur Laplacien. Une opération de seuillage sur la norme du gradient est généralement nécessaire afin d'éliminer les contours correspondants au bruit. Les considérations concernant le bruit dans la dérivée première sont encore plus importantes dans les calculs de dérivée seconde. On utilise donc couramment une combinaison de lissage et laplacien ce qui correspond au laplacien d'une gaussienne. L'estimation de la dérivée seconde étant très sensible aux bruits, il convient de filtrer très fortement l'image avant d'en mesurer le laplacien. Ainsi, afin de limiter les réponses dues au bruit de l'image, le plus souvent, on fait appel à un filtrage gaussien dont le laplacien est plus connu sous le nom de "chapeau mexicain".

Les plus simples opérateurs du Laplacien sont données par l'application des masques suivants :

- Laplacien en connexité 4 :

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Laplacien en connexité 8:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

La figure suivante nous montre les contours détectés en utilisant le Laplacien:

Figure 9: Image originale, contours détectés par lelaplacien.

- **Approche par filtrage optimal**: Cette approche a émergé dans les années quatre-vingt-cinq. Elle a permis une meilleure compréhension des conditions d'une bonne détection de contours et a ainsi conduit à des détecteurs de très bonne qualité. Les qualités attendues d'un filtre de détection de contours est un filtre de réponse impulsionnelle h(x) qui permet une bonne détection des contours c'est-à-dire une réponse forte même à de faibles contours, une bonne localisation de ceux-ci et une faible multiplicité des maxima dus au bruit (assurer que pour un contour, il n'y aura qu'une seule détection). Ces trois critères proposés par Canny (A computational approach to edge detection, 1986) s'expriment par l'optimisation conjointe de trois fonctionnelles qui permettent ainsi de définir le filtre linéaire optimal. Ces trois critères d'optimalités sont :

✓ Une bonne détection: Plus le filtre lisse le bruit, plus la détection est bonne : on cherche à maximiser le rapport signal sur bruit, par conséquent, minimiser les fausses réponses.

✓ Une bonne localisation: Moins le filtre lisse l'image, meilleure est la localisation : il s'agit de minimiser la distance entre les points détectés et le vrai contour.

✓ Réponse unique: On veut une réponse unique par contour. Il existe des cas où il est difficile de savoir si on est en présence de deux contours distincts ou un seul contour bruité : il s'agit de minimiser le nombre de réponse pour un seul contour. En peut citer comme filtres utilisant : le filtre de Canny et le filtre de Deriche.

II.2.2. Techniques de segmentation des régions

Les images sont composées de régions possédant des propriétés locales qui peuvent être la répartition des niveaux de gris. En regroupant des points de l'image qui possèdent une même propriété donnée, on obtient des régions uniformes. Cette opération s'appelle «segmentation».

La segmentation d'image est l'opération la plus importante dans un système de traitement d'images, car elle est située à l'articulation entre le traitement et l'analyse des images. L'intérêt de la segmentation est de partitionner une image en plusieurs régions homogènes, au sens d'un critère fixé a priori. L'intérêt de disposer de régions homogènes est de fournir des données simplifiées qui facilitent la tâche d'un système de reconnaissance de formes, ou autre système d'extraction des objets contenus dans l'image.

Une bonne méthode de segmentation sera celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu.

la segmentation peut être définie comme une partition d'une image I en une ou plusieurs régions R 1 , ..., R n telles que :

$$I = \bigcup_{i=1}^{n} R_i \; et \; R_i \cap R_j = \emptyset \; pour \; tout \; i \neq j \quad \text{..... Formule (9)}$$

Il existe plusieurs méthodes telles que la segmentation par croissance de région, par division de région, et par fusion de région que nous présentons ci-dessous:

a) Croissance de région (région growing):

Cette technique consiste à faire progressivement accroître les régions autour de leur point de départ. L'initialisation de cette méthode consiste à considérer chaque pixel comme une région.

Le principe de l'agrégation de pixel est le suivant : on choisit un germe (Le point de départ est le choix d'un ensemble de pixels appelés « germes ») et on fait croître ce germe tant que des pixels de son voisinage vérifient le test d'homogénéité. Lorsqu'il n'y a plus de pixels candidats dans le voisinage, on choisit un nouveau germe et on itère le processus.

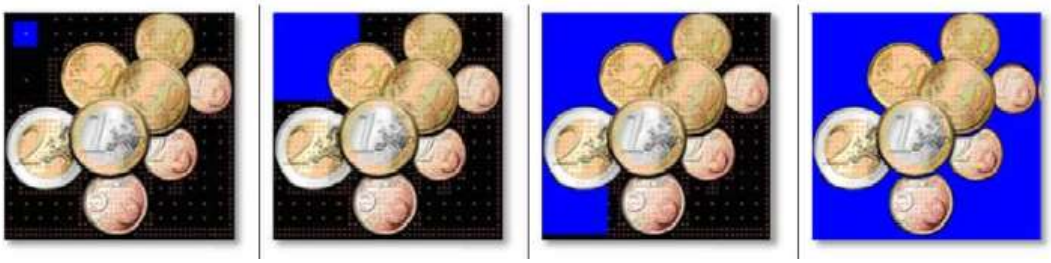


Figure 10: Croissance progressive des régions. [14]

Comme avantages de cette technique on peut citer:

- La simplicité et la rapidité de la méthode.
- La segmentation d'objet à topologie complexe.
- La préservation de la forme de chaque région de l'image.

Cependant, il existe plusieurs inconvénients, on cite:

- L'influence du choix des germes initiaux et du critère d'homogénéité sur le résultat de la segmentation.
- Une mauvaise sélection des germes ou un choix du critère de similarité mal adapté peuvent entraîner des phénomènes de sous-segmentation 1 ou de sur-segmentation.
- Il peut y avoir des pixels qui ne peuvent pas être classés.

b) Segmentation par fusion de régions (Merge):

Cette technique consiste à faire progressivement accroître les régions autour de leur point de départ. L'initialisation de cette méthode consiste à

Les techniques de réunion (region merging) sont des méthodes ascendantes où tous les pixels sont visités. Pour chaque voisinage de pixel, un prédicat P est testé. S'il est vérifié les pixels correspondants sont regroupés dans une région.

Les inconvénients de cette méthode se situent à deux niveaux :

- Cette méthode dépend du critère de fusion qui peut influer sur le résultat final de la segmentation.
- Elle peut introduire l'effet de sous-segmentation.

c) Segmentation par division de régions (Split):

La division consiste à partitionner l'image en régions homogènes selon un critère donné.Le principe de cette technique est de considérer l'image elle-même comme région initiale, qui par la suite est divisée en régions. Le processus de division est réitéré sur chaque nouvelle région (issue de la division) jusqu'à l'obtention de classes homogènes.

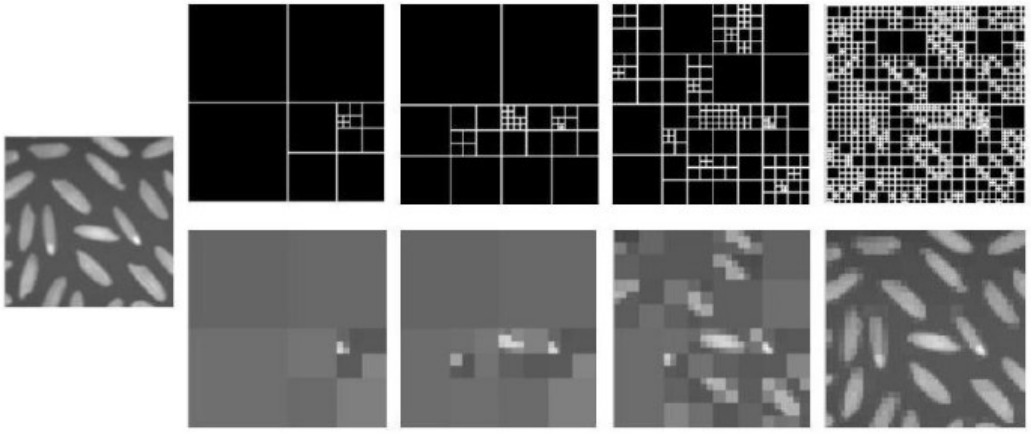


Figure 11: Décompositions successives des blocs.

Cette méthode présente un inconvénient majeur qui est la sur-segmentation. Ce dernier peut être résolu en utilisant la méthode de diffusion-fusion que nous allons présenter dans ce qui suit.

d) Segmentation par division-fusion (Split and Merge):

Cette méthode combine les deux méthodes décrites précédemment, la division de l'image en de petites régions homogènes, puis la fusion des régions connexes et similaires au sens d'un prédicat de regroupement. On part du principe que chaque pixel représente à lui seul une région.

Deux régions seront fusionnées si elles répondent aux critères de similarité des niveaux de gris et d'adjacence de régions .On s'arrête quand le critère de fusion n'est plus vérifié.

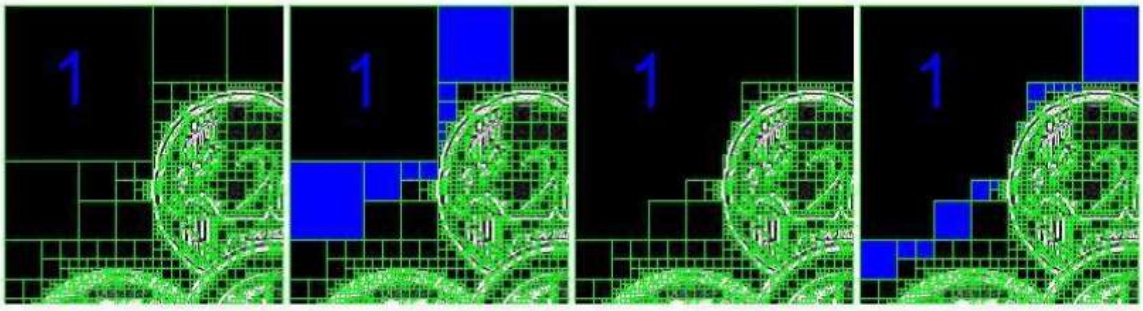


Figure 12: Agrégation itératives des blocs similaires.

Comme inconvénients de cette technique, il existe:

- Les régions obtenues ne correspondent pas, dans tous les cas, aux objets représentés dans l'image.
- Les limites des régions obtenues sont habituellement imprécises et ne coïncident pas exactement aux limites des objets de l'image.
- La difficulté d'identifier les critères pour agréger les pixels ou pour fusionner et diviser les régions.

II.3. Classification par cascade

Un classificateur est formé à l'aide de quelques centaines de vues d'un objet particulier (un visage ou une voiture), appelées exemples positifs, qui sont redimensionnées de la même manière. taille (par exemple, 20x20), et des exemples négatifs - images arbitraires de la même taille.

Une fois qu'un classificateur est formé, il peut être appliqué à une région d'intérêt (de la même taille que celle utilisée pendant la formation) dans une image d'entrée. Le classificateur génère un «1» si la région est susceptible d'indiquer l'objet (c'est-à-dire visage / voiture) et «0» dans les autres cas. Pour rechercher l'objet dans l'image entière, vous pouvez déplacer la fenêtre de recherche sur l'image et vérifier chaque emplacement à l'aide du classificateur. Le classificateur est conçu pour pouvoir être facilement «redimensionné» afin de pouvoir trouver les objets d'intérêt de différentes tailles, ce qui est plus efficace que de redimensionner l'image elle-même. Ainsi, pour trouver un objet de taille inconnue dans l'image, la procédure de numérisation doit être effectuée plusieurs fois à différentes échelles.Le mot "cascade" dans le nom du classificateur signifie que le classificateur résultant est composé de plusieurs classificateurs plus simples (étapes) qui sont

appliqués ultérieurement à une région d'intérêt jusqu'à ce que le candidat soit rejeté à une certaine étape ou que toutes les étapes soient franchies. [16]

La caractéristique utilisée dans un classificateur particulier est spécifiée par sa forme, sa position dans la région d'intérêt et son échelle (cette échelle n'est pas la même que celle utilisée à l'étape de la détection, bien que ces deux échelles soient multiplié).

II.4. Conclusion

Dans ce chapitre nous avons présenté les différentes techniques de prétraitement et de segmentation des images que nous allons les appliquées dans le chapitre suivant pour détecter le contour des véhicules dans une vidéo.

III. CHAPITRE III

Détection des véhicules

III.1. Introduction

Afin d'améliorer les systèmes de surveillance routière pour éliminer des problèmes tels que les accidents de circulation, on fait appel a des techniques de traitement d'images pour la détection des véhicules et la mesure de leurs vitesses.

Cette détection automatique du mouvement dans les séquences vidéo est un sujet très actif depuis le début des années 1980. Depuis cette date, de nombreux travaux ont été menés et de nombreuses approches ont été mises en œuvre, et en particulier depuis le milieu des années 1990, lorsque la puissance des ordinateurs grand public a permis d'envisager sérieusement un traitement en temps réel des données vidéo, toutefois la plupart d'entre elles ne sont valables que dans certaines conditions spécifiques.

Dans les chapitres précédents, nous avons parlé des notions de base du traitement d'images et de vidéos (en premier chapitre), des techniques qui ont été développées pour une meilleure détection d'objets(dans le chapitre2).

Dans ce chapitre, nous nous focaliserons sur la détection de véhicules, le tracking, des contributions de quelques chercheurs dans ce domaine, puis nous décrirons la technique qu'on va implémenter.

III.2. La détection des véhicules

La détection des véhicules en mouvement dans une séquence d'images est actuellement un sujet de recherche très actif en vision par ordinateur, le but est de reconnaître les véhicules d'une image à une autre,et de reconstruire leur trajectoire afin d'extraire les informations qui caractérisent chacun d'entre eux tel que la distance parcourue, la vitesse,...etc.

Durant la dernière décennie, de nombreuses méthodes de détection et de poursuite de mouvement de véhicules ont été proposées dans la littérature scientifique.

Le but des méthodes de détection de mouvement est de séparer les véhicules de leurs environnements afin d'extraire des informations utiles pour le traitement de poursuite, telles que la position, la taille,...etc.

Parmi les travaux qui ont traité ces problèmes,on peut citer:

- Jin-Cyuan et Shih ont construit un système de surveillance de trafic routier en autoroute capable d'estimer des paramètres de trafic tel que le nombre et les classes des véhicules,le système proposé comprend trois étapes: extraction des régions de véhicules, poursuite de mouvement et extraction des paramètres de trafic. La méthode de soustraction de fond est premièrement utilisée pour détecter les véhicules sur la route. Les deux auteurs ont utilisé quelques propriétés géométriques pour éliminer les fausses détections et un algorithme de soustraction d'ombre pour améliorer la précision de la détection. La méthode de poursuite utilisée consiste à associer les graphes(les contours) afin de trouver la correspondance entre les véhicules détectés à des instants différents. Dans ce travail les auteurs ont utilisé des vidéos avec différentes conditions d'éclairage pour prouver l'efficacité de leur système.

- Ehsan Adeli Mosabbeb et autres ont utilisé quelques paramètres statiques pour chaque trame de la vidéo tels que la moyenne et l'écart type afin de détecter et de segmenter les véhicules; cette approche a été utilisée pour la détection de véhicules en mouvement dans une scène congestionnée, où il s'avère nécessaire de trouver des solutions pour surmonter le problème de fausses détections. Dans leur travail ils ont proposé de soustraire l'ombre afin de mieux séparer les véhicules très prés les uns des autres. Pour montrer la robustesse et la précision de cette approche ils ont utilisé des scènes routières sous différentes conditions climatiques telles que le bruit, la pluie, la neige, et des conditions d'éclairage variables.

- ZheLiu et Yangzhou Chen ont proposé un système d'estimation des paramètres de trafic routier basé sur la détection de mouvement par l'algorithme de différence de trames consécutives, et la poursuite par application de l'algorithme camshift qui s'est avéré efficace pour le suivi des véhicules de formes et de tailles variables et sous différentes conditions de luminosité. Ils ont ensuite reconstruit la trajectoire individuelle de chaque véhicule, ce qui donne

la possibilité d'estimer les paramètres individuels de chaque véhicule, tels que la distance parcourue ainsi que la vitesse.

- XieLei et Guangxi ont utilisé le filtre de Kalman pour la poursuite des véhicules en mouvement en temps réel dans une scène routière, leur travail comprend deux étapes: détection de mouvement et poursuite. Ils ont testé leur algorithme sur différentes scènes et pour différentes conditions climatiques. Les résultats obtenus ont prouvé la robustesse de l'algorithme en temps réel et pour différentes conditions de luminosité.

- François Bardet et Thierry Château se sont intéressés à la poursuite d'objets multiples dans un environnement routier en utilisant une ou plusieurs caméras, pour cela ils ont employé une méthode qui combine un filtre particulaire à une Chaine de Markov.

- Vamsi Krishna etHanmandlu ont présenté une méthode d'estimation de la vitesse des véhicules, le mouvement est extrait par l'utilisation d'une équation basée sur la projection sphérique qui relie le mouvement de l'image au mouvement de l'objet. La tâche de poursuite est réalisée par l'algorithme de Kanade-Luca-Tomasi. L'équation de mouvement a été reformulée en un modèle d'espace d'état dynamique à laquelle ils ont appliqué le filtre de Kalman et sa version étendue pour l'estimation de la vitesse de l'objet et pour la prédiction de sa nouvelle position. Leur travail a été testé sur une scène en utilisant une caméra non calibrée, cet algorithme est simple et précis pour le calcul de la vitesse en temps réel.

- Daniel J. Daileyet autres ont développé un algorithme pour l'estimation de la vitesse moyenne du trafic routier en utilisant une séquence d'images à partir d'une caméra non calibrée ils ont affirmé qu'une calibration de la caméra n'est pas nécessaire pour l'estimation de la vitesse, et pour cela ils ont utilisé:

- des relations géométriques intrinsèques disponibles dans l'image.
- certaines hypothèses qui permettent de réduire le problème à une géométrie d'une seule dimension (1-D).
- Une différentiation des trames pour l'isolation des contours en mouvement et la poursuite des véhicules entre les trames.

- des paramètres à partir de la distribution de longueurs de véhicules pour l'estimation de la vitesse.

Leur méthode a présenté une alternative viable à la calibration de la caméra.

M.Burlin et autres ont proposé un système d'analyse de trafic routier capable de compter et de détecter les véhicules à l'arrêt ou en contre-sens. En définissant des pixels ambigus pour les objets détectés, le système utilise une modélisation de la scène pour améliorer la détection et le suivi des objets. La première étape de ce système consiste à extraire des informations sur la géométrie de la scène (position et caractérisation des voies).Les bordures des voies, l'estimation de la profondeur dans l'image, et les informations de mouvement sont ensuite utilisées pour aider à la segmentation des objets et à leur suivi image après image. Les résultats obtenus sont prometteurs et montrent la robustesse du système proposé pour le suivi de plusieurs objets simultanément et la correction des erreurs de segmentation.

- Xinting Pan et autres ont proposé un système de surveillance de trafic pour la détection et le comptage des véhicules, tout d'abord,ils extraient la Région d'intérêt (ROI) de la vidéo en utilisant une combinaison de la soustraction de fond et la détection de contours. Après la détection ils ont utilisé une méthode automatique de division de voies. Enfin en se basant sur la largeur des voies,ils ont employé plusieurs fenêtres d'auto-adaptation pour le comptage des véhicules au lieu de la méthode de comptage traditionnelle à base de fenêtre fixe.Les résultats expérimentaux ont prouvé la robustesse de leur système.

III.3. Le suivi (tracking)

Le tracking est un procédé de localisation d'un (ou plusieurs) objet en mouvement en temps réel en utilisant une caméra. Un algorithme analyse les photogrammes de la vidéo et localise les cibles en mouvement sur la vidéo.

La principale difficulté dans le tracking sur une vidéo est d'associer la localisation des cibles dans les photogrammes successifs, particulièrement lorsque les objets bougent rapidement par rapport au frame rate. Les systèmes de tracking sur une vidéo normalement utilisé, utilise un

modèle en mouvement qui décrit comment l'image de la cible peu changer en tenant compte du mouvement possible de l'objet traquer.

III.4. La classification

III.4.1. Définition

Tout d'abord, la classification cherche à identifier ce que représente chacune des régions segmentées. C'est un problème fondamental en vision par ordinateur, qui a de nombreuses applications concrètes. Le but est de construire un système capable d'assigner correctement une catégorie (autrement dite classe) à n'importe quelle image en entrée. Ses classes peuvent être connues à priori (classification supervisée) ou inconnue (classification non supervisée).

- **Classification supervisée**: dans ce type de classification les classes sont créées au préalable avec définition des règles permettant de classer les objets détectés dans l'images dans ces dernières à partir de variables qualitatives et quantitatives qui les décrivent. C'est à dire, on dispose au départ un échantillon dit d'apprentissage dont le classement est connu.

Par exemple: nous disposons d'éléments déjà classés: voitures, maison, arbres, ...etc. on classe les éléments dans l'une de ces classes.

- **Classification non supervisée**: cette classification ne nécessite aucun apprentissage, elle consiste à laisser l'ordinateur calculer automatiquement les classes sur la base de plusieurs (en tout cas plus d'une) bandes de fréquences de l'image. Cela nous laisse la tâche d'identifier le bon nombre et la nature réelle des classes obtenues.

Par exemple: comme exemple de cette classification, nous avons des éléments non classés (des voitures par exemples) et on veut les classer dans la même classe.

III.4.2. L’apprentissage machine (approche classique)

Cette stratégie, qui consiste à définir "à la main" des règles pour différencier les classes d'images, était en fait traditionnellement utilisée il y a 40 ans. Mais elle demande un travail fastidieux et manque de flexibilité : avec la quantité énorme d'images à notre disposition aujourd'hui, elle est extrêmement difficile à mettre en œuvre. [17]

Composés de deux blocs: un extracteur de caractéristiques (feature extractor en anglais), suivi d’un classifieur entraînable simple. L’extracteur de caractéristiques est programmé «à la main», et transforme le tableau de nombres représentant l’image en une série de nombres, un vecteur de caractéristiques, dont chacun indique la présence ou l’absence d’un motif simple dans l’image. Ce vecteur est envoyé au classifieur, dont un type commun est le classifieur linéaire. Ce dernier calcule une somme pondérée des caractéristiques: chaque nombre est multiplié par un poids (positif ou négatif) avant d’être sommé. Si la somme est supérieure à un seuil, la classe est reconnue. Les poids forment une sorte de «prototype» pour la classe à laquelle le vecteur de caractéristiques est comparé. Les poids sont différents pour les classifieurs de chaque catégorie, et ce sont eux qui sont modifiés lors de l’apprentissage. [18]

- Les algorithmes utilisés pour la classification:

➢ KNN: La méthode de k voisins les plus proches (en anglais K-Nearest Neighbor), l’un des algorithmes les plus simples d’apprentissage artificiel. C’est une méthode d’apprentissage dédiée à la classification qui a pour objectif de classer les exemples non étiquetés sur la base avec les exemples de la base d’apprentissage.
Son principe de fonctionnement est comme suit:

Nous avons: - Un ensemble de données d’apprentissage D.
- Une fonction de distance d.
- Un entier K que nous devons bien choisir pour une meilleure précision.

L’algorithme commence a tester point par point, pour tout nouveau point x, il recherche dans D, les K points les plus proches de x au sens de la distance d, et attribut x à la classe qui est la plus fréquente parmi ces K voisins, comme le représente la figure qui suit.

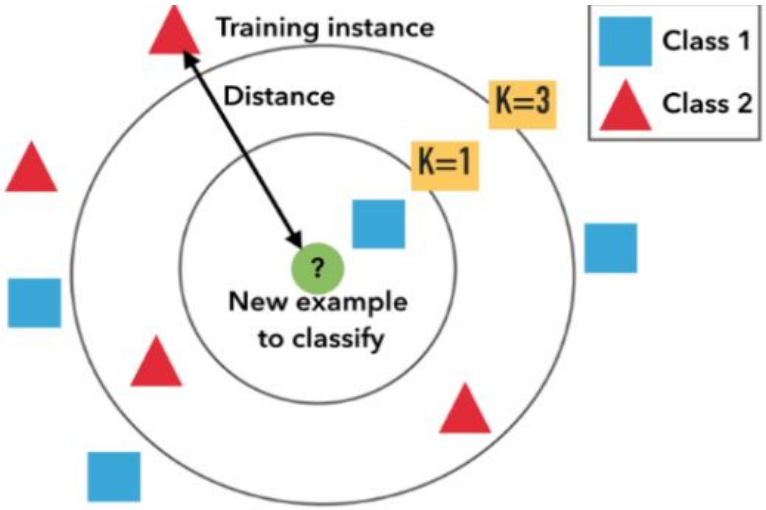


Figure 13: Schéma explicatif du fonctionnement de l'algorithme KNN.

Dans cet exemple, l'ensemble des données d'apprentissage est illustré avec les Class 1 et Class 2, si on prend la valeur de k=1 alors l'objet est assigné à la classe de son prochain voisin qui est Class 1, si on prend la valeur de k=3 alors l'objet détecté sera assigné à Class 2.

Pour choisir la valeur de k nous devons:

-calculer la racine carrée de n (sqrt (n)), tel que n est le nombre total des données.

-choisir une valeur impaire de K pour éviter toute confusion entre deux classes de données.

- SIFT: Publié par David Lowe en 1999, l'algorithme SIFT (Scale Invarient Feature Transform) comme son nom l'indique cherche à trouver des points clés (ou points caractéristiques) qui sont invariants à plusieurs transformations (rotation, echelle et illumination) puis faire une comparaison entre les deux images (la méme image avant et aprés les transformations) en utilisant ces points clés.

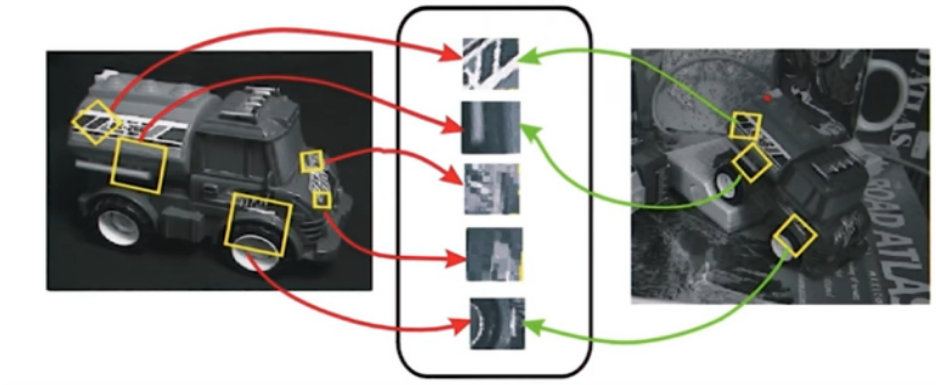


Figure 14: Principe de fonctionnement de l'algorithme SIFT.

- SURF (Speed Up Robust Features): développé par Herbert Bay, Tinne Tuytelaars, et Luc Van Gool en s'appuyant sur le SIFT, le SURF est plus rapide et plus robuste à certaines transformations.

III.4.3. La classification et les réseaux de neurones

L'approche classique consiste à définir "à la main" des règles pour différencier les classes d'images, elle demande un travail fastidieux (un bon extracteur de caractéristiques est très difficile à construire) et manque de flexibilité : avec la quantité énorme d'images à notre disposition aujourd'hui, elle est extrêmement délicate à mettre en œuvre.

C'est là qu'intervient l'apprentissage profond (en anglais deep learning). Qui est une classe de méthodes dont les principes sont connus depuis la fin des années 1980, mais dont l'utilisation ne s'est vraiment généralisée que depuis 2012, environ. [19]

L'apprentissage en profondeur est un ensemble de méthodes d'apprentissage visant à modéliser des données à l'aide d'architectures complexes combinant différentes transformations non linéaires. Les briques élémentaires de l'apprentissage en profondeur sont les réseaux de neurones, qui sont combinés aux réseaux de neurones profonds. Ces techniques ont permis des progrès significatifs dans les domaines du traitement du son et de l'image, notamment la reconnaissance faciale, la reconnaissance vocale, la vision par ordinateur, le langage automatisé traitement, classification du texte (pour la reconnaissance des exemples). Il existe plusieurs types d'architectures pour les réseaux de neurones:

• Les perceptrons multicouches, les plus anciens et les plus simples.

• Les réseaux de neurones convolutionnels (CNN), particulièrement adaptés au traitement d'images.

• Les réseaux de neurones récurrents, utilisés pour les données séquentielles telles que les séries de textes ou d'orthèses.

a) **Les réseaux de neurones**: il existe deux types de réseaux de neurones, des réseaux neurones biologiques et d'autres artificiels inspirés des RNB et configurés pour exécuter des actions spécifiques afin de reproduire certaines fonctions du cerveau comme la reconnaissance des formes, la mémorisation associative ou bien l'apprentissage. Le RNA est un ensemble de neurones interconnectés basé sur un modèle de neurones simplifié, sa connaissance est acquise par apprentissage puis stockée dans les connexions entre les neurones avec une valeur/force appelée "poids synaptique" (weight).

Un réseau de neurones artificiel de trois types de couches, une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie.

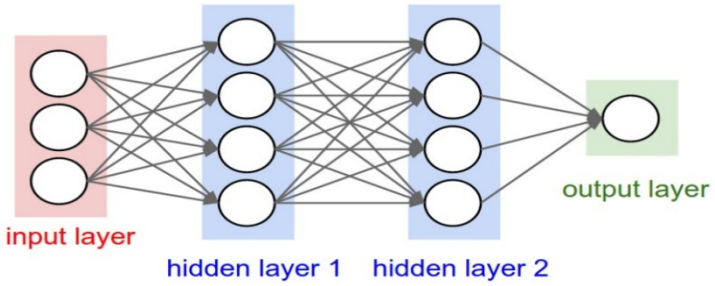


Figure 15: Architecture des réseaux de neurones artificiels.

b) Les neurones: Un neurone n'est rien d'autre qu'un nœud comme ceux qui sont présentés dans la figure précédente. Chaque nœud prend la somme pondérée de ses entrées, et passe à travers une fonction d'activation non linéaire. C'est la sortie du nœud, qui devient alors l'entrée d'un autre nœud dans la couche suivante. Entraîner un réseau neuronal signifie d'apprendre les poids associés à chaque classe.

La somme pondérée de ses entrées passe par une fonction d'activation non linéaire. Il peut être représenté comme un produit scalaire vectoriel, où n est le nombre d'entrées pour le nœud.

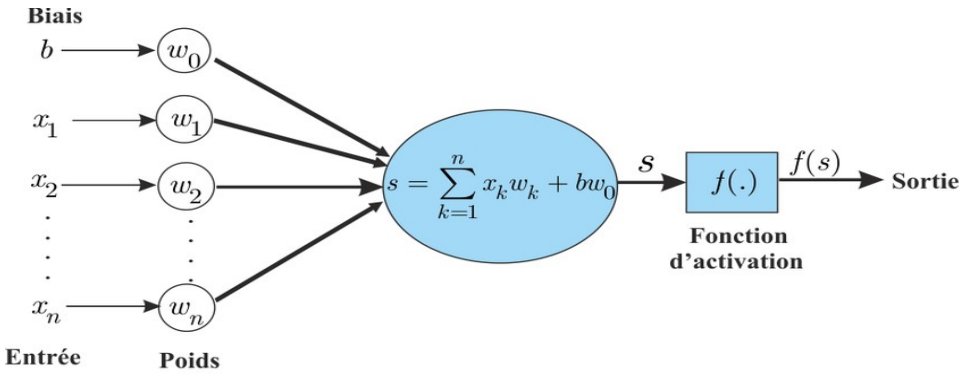


Figure 16: Schéma explicatif d'un neurone artificiel.

c) **Les réseaux de neurones convolutifs (CNN)**:

Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones : ils présentent donc toutes les caractéristiques listées ci-dessus. Cependant, les CNN sontspécialement conçus pour traiter des images en entrée.

L'idée est de passer l'image dans une succession de filtres convolutives apportant une description réduite et pertinente de l'image. Ces caractéristiques sont, par la suite, envoyées à un percéptron multicouche composé de couches cachées et d'une couche de sortie complètement connectées permettant la classification du chiffre présent dans l'image. Les filtres de convolution et les couches complètement connectée sont appris simultanément. Les CNN sont un type particulier de réseaux de neurones applicables facilement à des images pour capter spatialement de l'information. Les CNN peuvent être vus comme un assemblage de modules en série permettant l'extraction de caractéristiques de manière hiérarchique à partir des pixels d'une image. [20]

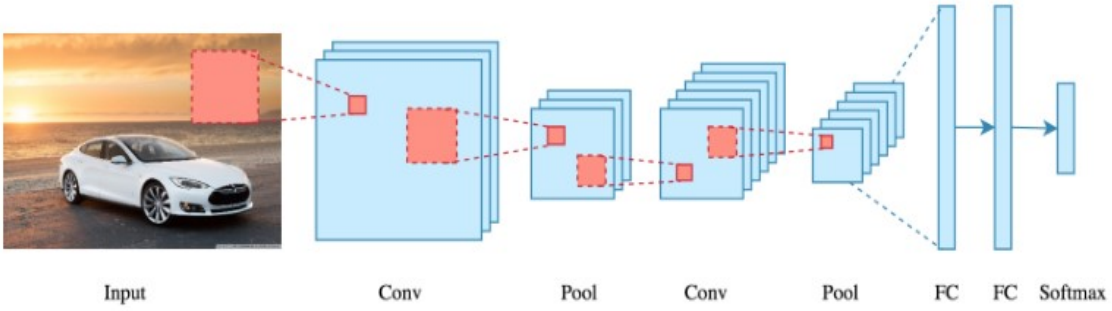


Figure 17: un CNN (convolutional neural network)

La figure ci-dessus montre l'architecture de base de CNN. Les principales composantes des architectures de CNN sont les suivantes:

- couches de convolution
- couches de pooling
- Couches entièrement connectées
- fonction Softmax

Chaque image d'entrée passe à travers plusieurs couches de convolution et pooling, suivi de quelques couches entièrement connectées. De plus, la fonction softmax est appliquée à la fin pour effectuer la classification multiclass.

- **Couches de convolution**

La couche de convolution est la première et principale composante de CNN. Couches de convoltion sont constitués de neurones qui agissent comme des filtres. Les filtres sont glissés séquentiellement sur l'image entière pour créer des cartes de caractéristiques. Ce processus de glissement est appelé convolution.Les figures 18 et 19 illustrent le processus de convolution sur une entrée 6x6 carte de fonctions utilisant un filtre 3x3 (noyau). Après avoir fait glisser le filtre sur l'ensemble de l'entrée, le résultat est une carte d'entités en sortie de taille 4x4.

3	2	5	1	8	6
1	4	6	4	7	3
8	9	1	5	3	7
2	4	5	2	9	2
3	1	6	1	5	8
9	3	7	3	2	5

(a) Carte d'entités en entrée

1	0	0
0	1	1
1	1	1

(b) Filtre

Figure 18: Carte d'entités en entrée 6x6 et filtre 3x3

3x1	2x0	5x0	1	8	6
1x0	4x1	6x1	4	7	3
8x1	9x1	1x1	5	3	7
2	4	5	2	9	2
3	1	6	1	5	8
9	3	7	3	2	5

(a) processus de convolution

31	27	25	26
22	21	30	27
27	24	24	30
28	24	23	25

(b) carte d'entités en sortie

Figure 19: Processus de convolution d'une entrée 6x6 avec un filtre 3x3

Comme indiqué dans l'architecture CNN ci-dessus, le premier bloc est un bloc de convolution couche. Chaque couche fonctionne comme un filtre différent et apprend différentes cartes de caractéristiques à partir de une image d'entrée ou une carte de caractéristiques d'entrée. Par exemple, étant donné une image en entrée, si nous avoir trois couches de convolution dans un bloc. La sortie de chaque couche est une carte de caractéristiques. pour trois caractéristiques différentes telles que les couleurs, les bords et les formes. Aussi, la sortie de chaque bloc de convolution a trois dimensions, hauteur, largeur et profondeur. Hauteur et largeur correspond à la hauteur et à la largeur des cartes de caractéristiques, et profondeur est le nombre de couches de convolution.

- **Couches de pooling**

Les couches de pooling suivent généralement des couches de convolution. La couche de pooling réduit la dimensionnalité de chaque carte de caractéristiques. La couche prend chaque entité indépendante mappe, sous-échantillonne et crée une carte d'entités compressée. Bien que la

sortie du regroupement des couches est toujours tridimensionnelle comme des couches convolutives. La profondeur reste la même, seules la hauteur et la largeur sont condensées. Il existe différents types de mise en commun tels que max-pooling, sum-pooling et average-pooling . Le pooling maximum est le plus populaire parmi eux, Max-pooling fonctionne en faisant glisser une fenêtre sur chaque carte de fonctionnalité et en sélectionnant une valeur maximale dans cette fenêtre.

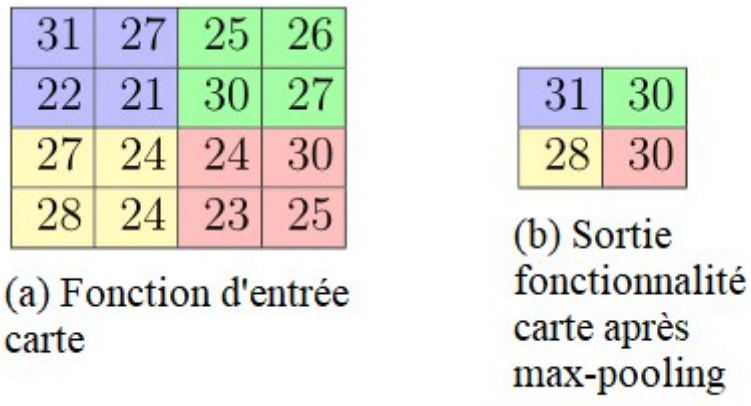


Figure 20: Fonction de la couche de pooling maximal avec une fenêtre 2x2 et une foulée 2

La figure 20 ci-dessus montre un exemple de max-pooling où la taille de la fenêtre est 2x2 et la foulée est de 2. La longueur de foulée est de combien de pixels on déplace le filtre ou fenêtre à chaque étape. Le principal avantage de pooling est qu'elle aide à réduire le nombre de paramètres ce qui réduit le temps de formation.

- **Couches entièrement connectées (FC)**

Certaines couches entièrement connectées sont ajoutées après les couches de convolution et de pooling. Un réseau entièrement connecté est un réseau où tous les neurones sont connectés à chaque neurone dans les couches voisines. Comme mentionné ci-dessus, la sortie de convolution et de pooling, les couches sont en trois dimensions, mais les couches entièrement connectées consomment un vecteur à une dimension en entrée. Donc, on aplatit la sortie finale puis on l'envoie comme entrée des couches connectées. Les couches FC combinent ensuite ces cartes de caractéristiques pour élaborer un modèle.

- **Fonction Softmax**

La fonction softmax est l'une des fonctions de perte qui peut être ajoutée comme dernière couche d'un CNN. La fonction softmax est utile pour la détection multiclass; ça nous donne la probabilité de prédire une classe sur toutes les autres classes. Si on utilise le softmax fonction de détection multiclass, la sortie de cette fonction est unidimensionnelle vecteur de probabilités de toutes les classes. Donc, la classe avec la probabilité la plus élevée est la classe cible. [21]

a) **Les réseaux de neurones convolutifs par région (RCNN, Fast RCNN et Faster RCNN)**

- **RCNN:**

Le RCNN est le premier algorithme qui fait la classification d'objets par recherche par région. D'abord, il analyse l'image d'entrée pour les objets possibles détectés en utilisant un algorithme appelé «selective search» en générant 2000 régions. Ensuite, pour chaque région, il utilise un réseau de neurones convolutifs afin d'extraire ces caractéristiques, enfin l'application d'un classifieur pour identifier les objets et afficher un rectangle au tour de l'objet détecté.

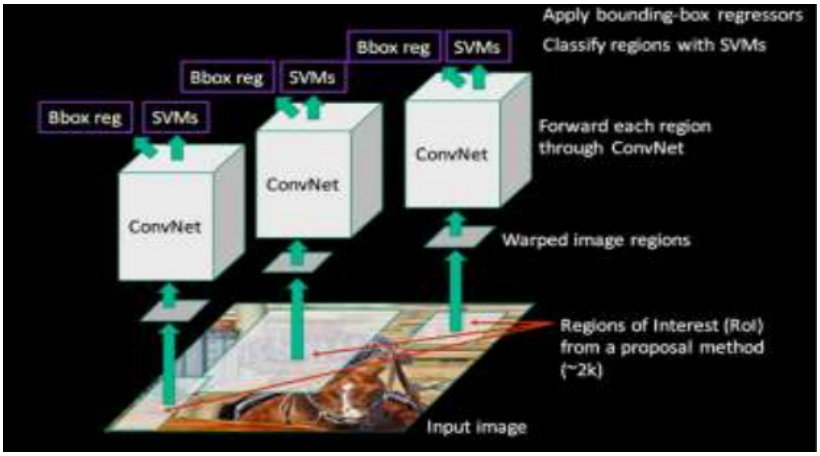


Figure 21: Méthode de fonctionnement du RCNN [13].

- **Fast-RCNN:**

Le Fast-RCNN (comme son nom l'indique) est plus rapide que le RCNN, La raison pour laquelle «Fast R-CNN» est plus rapide que R-CNN est qu'il n'est pas nécessaire d'alimenter chaque fois 2000 propositions de régions sur le réseau de neurones à convolution. Au lieu de

cela, l'opération de convolution est effectuée une seule fois par image et une carte de caractéristiques est générée à partir de celle-ci. [22]

- **Faster-RCNN:**

Le R-CNN et Fast R-CNN utilisent la recherche sélective pour rechercher les propositions de région, qui est un processus lent et fastidieux qui affecte les performances du réseau. Par conséquent, Shaoqing Ren et al. est venu avec un algorithme de détection d'objet qui élimine l'algorithme de recherche sélective et permet au réseau d'apprendre les propositions de région.

Semblable à Fast R-CNN, l'image est fournie en tant qu'entrée à un réseau de convolution qui fournit une carte de caractéristiques de convolution. Au lieu d'utiliser un algorithme de recherche sélective sur la carte des caractéristiques pour identifier les propositions de région, un réseau séparé est utilisé pour prédire les propositions de région. Les propositions de région prédites sont ensuite remodelées à l'aide d'une couche de regroupement RoI qui est ensuite utilisée pour classer l'image dans la région proposée et pour prédire les valeurs de décalage pour les boîtes de délimitation. [13]

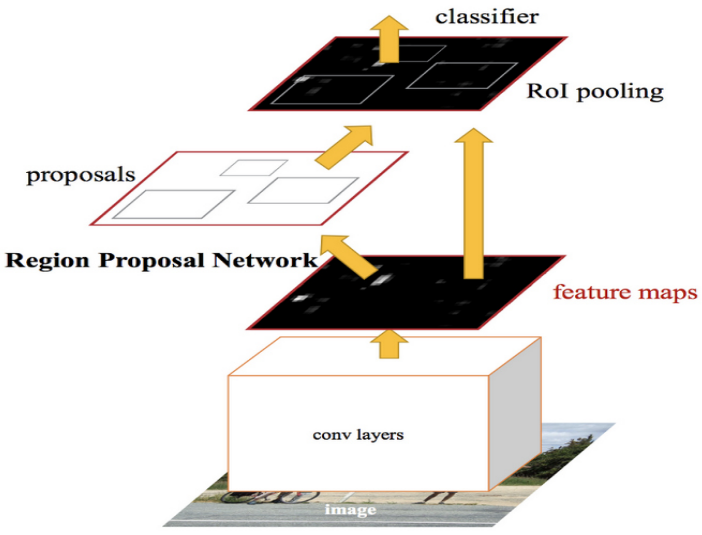


Figure22: Fonctionnement du Faster-RCNN

III.5. Technique utilisée

Comme nous avons vu au cours de ce chapitre qu'il existe plusieurs méthodes de détection et suivi de véhicules, nous allons à présent présenter dans ce qui suit les méthodes que nous avons adopté et implémenté par la suite.

III.5.1. Détection de véhicules

Notre objectif est de détecter plusieurs véhicules à partir d'une caméra, donc le réseau de neurones à convolution standard n'est pas la solution qui va résoudre notre problème car le nombre des objets d'intérêt est non fixé, et peuvent avoir différents emplacements spatiaux dans l'image.

Le Faster-RCNN semble être le bon choix à prendre vu que c'est la technique la plus développées des RCNN pour donner enfin un réseau plus rapide capable de détecter des objets en temps réel, mais malgré les étonnantes réalisations affichées dans ce domaines, aucune de ces architectures n'a atteint l'objectif (détection en temps réel) et les problèmes qui ont été identifiés sont:

- L'entraînement des données est lourd et trop long.
- La formation se déroule en plusieurs phases (par exemple, proposition de région d'entraînement vs classificateur).
- Le réseau est trop lent au moment de l'inférence (c'est-à-dire lorsqu'il s'agit de données non liées à l'entraînement).

Pour cela nous utiliserons une application basée sur une meilleure méthode, plus robuste et plus rapide appelée SSD (Single Shot MultiBox Detector) qui est conçue pour la détection d'objets en temps réel.

SSD comme son nom implique:

- Single Shot: Cela signifie que les tâches de localisation et de classification des objets sont effectuées en un seul passage du réseau.
- MultiBox: c'est le nom d'une technique de régression de la boîte englobante développée par Szegedy et al. [23]
- Detector: Le réseau détecte les objets et les classe en même temps.

La détection d'objets en SSD se compose de deux parties:

- Extraire des cartes de caractéristiques (en utilisant VGG16).
- Appliquer des filtres de convolution pour détecter les objets.

l'architecture de SSD s'appuie sur l'architecture VGG-16 qui a été utilisé comme réseau de base pour ses performances élevées dans les tâches de classification d'images de haute qualité et sa popularité pour les problèmes pour lesquels le transfert de l'apprentissage contribue à l'amélioration des résultats. Au lieu des couches entièrement connectées d'origine de VGG, un ensemble de couches convolutives auxiliaires (à partir de conv6) a été ajouté, permettant ainsi d'extraire des entités à plusieurs échelles et de réduire progressivement la taille de l'entrée pour chaque couche suivante. SSD effectue 8732 prévisions en utilisant 6 couches.

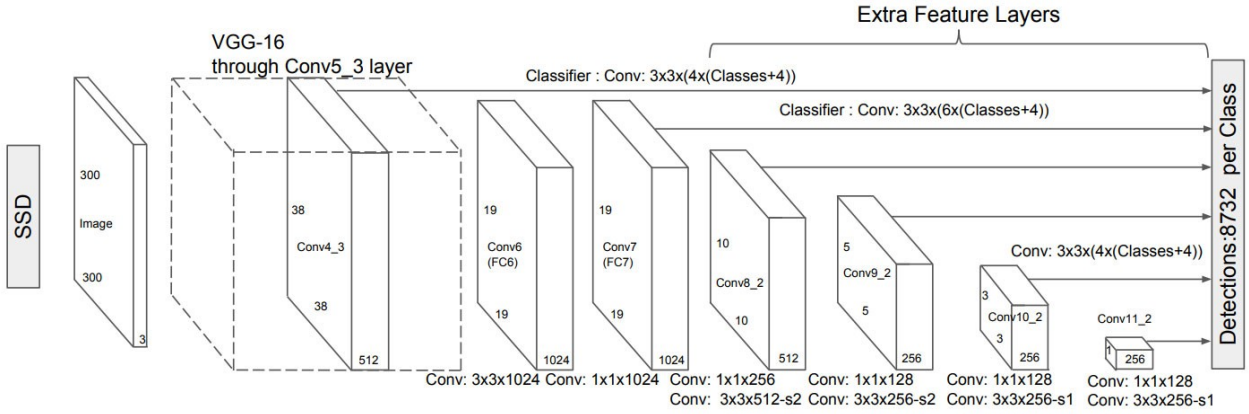


Figure23: Architecture du SSD

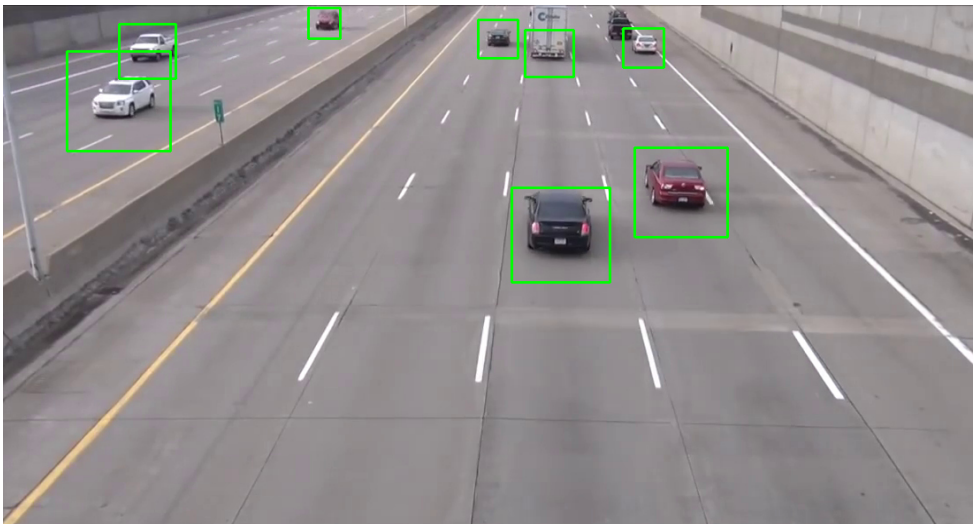


figure 24: Détection de véhicules.

III.5.2. Mesure de vitesse

Afin de mesurer la vitesse des véhicules détectés nous devons diviser la distance parcourue par le véhicule détecté par le temps du déplacement, pour cela nous avons ajouté quelques équations, le premier fait la conversion de la distance entre les pixels à la distance réelle. La deuxième, pour obtenir le temps total écoulé pour qu'un véhicule traverse la zone de retour. La troisième équation, nous effectuons une mise à l'échelle manuelle car nous n'avons pas effectué d'étalonnage de la caméra. L'équation est:

vitesse = échelle longueur réelle /échelle temps réel passé / constante d'échelle.

III.5.3. Détection et reconnaissance de plaques d'immatriculation

Après avoir détecté et enregistrer les véhicules, il va falloir les identifier d'une façon unique et pour cela on a besoin d'un caractéristique unique pour chacun d'eux, la plaque d'immatriculation est l'objet idéal pour cela. Sur ce deuxième module on a comme entrée l'image représentant le véhicule et donne en sortie la plaque d'immatriculation, cela se fait en trois étape qui sont la détection, la coupure, l'enregistrement sur disque. Un deuxième réseau neuronal convolutif est utilisé pour la détection de la plaque d'immatriculation des véhicules déjà enregistrée par le 1er module, et donne comme sortie une image de la plaque d'immatriculation.

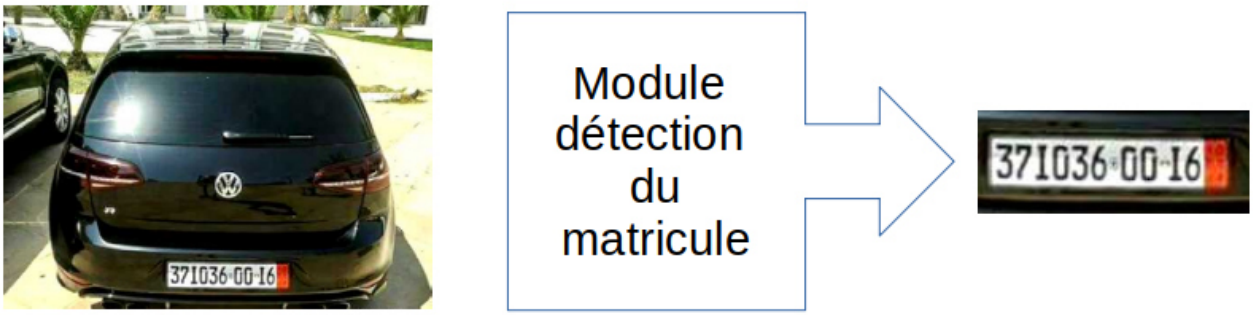


Figure 25: Détection de plaque d'immatriculation.

En suite un dernier module est responsable de la détection et la reconnaissance des chiffres sur l'image déjà enregistrées par le 2 éme module. Les caractères détectés sont désordonnés lors de la détection, un traitement d'ordonnancement est nécessaire avant de les enregistrer sur un fichier. [24]

III.6. Conclusion

Nous avons décrie dans ce chapitre les méthodes et techniques qu'on à implémentées. Dans le chapitre qui suit, nous détaillerons plus sur l'implémentation et nous afficherons les résultats obtenues.

IV. CHAPITRE IV

Implémentation de la technique utilisée

IV.1. Introduction

Après avoir pris connaissance théorique dans les chapitres précédents des principes de la détection et de suivi de véhicules, nous allons dans ce chapitre implémenter les techniques que nous avions proposé. En commençant par les outils et les logiciels utilisés arrivait a l'implémentation sur matériels et enfin nous décrirons les scripts et applications sur les quels se base notre projet.

IV.2. Outils et logiciels utilisés

IV.2.1. Tensorflow

Tensorflow est une bibliothèque opensource de machine learning, créé par l'équipe Google Brain en 2011 et appelé DistBelief à ce moment la. En modifiant son code source en 2015, cet outil est devenu une bibliothèque basée applications, Google l'a rendu opensource et l'a renommé Tensorflow.

Le Tensorflow permet de résoudre des problèmes mathématiques complexes avec aisance et de développer et d'exécuter des applications de machine learning et de deep learning.

Contrairement à la technologie machine learning qui peut être utile pour de nombreux usages mais qui est en même temps complexe à manipuler (acquisition de données, entraînement de modèles, déploiement de réseaux de neurones), le Tensorflow:

- permet l'apprentissage en profondeur.
- est un outil opensource et gratuit.
- est soutenu par Google.
- est une compétence reconnue par de nombreux employeurs.

- est surtout facile à implémenter.

✓ **Fonctionnement**

TensorFlow permet aux développeurs de **créer des graphiques de dataflow (dataflow graphs)**, à savoir des structures qui décrivent la façon dont les données se déplacent sur un graphique ou une série de noeuds de traitement.

Chaque nœud du graphique représente une opération mathématique, et chaque connexion entre ces nœuds est une **flèche de données multidimensionnelle : un tensor**.

Les nœuds et les tensors sont des objets Python, les applications TensorFlow aussi. Cependant, les opérations mathématiques en elles-mêmes ne sont pas effectuées en Python. Les bibliothèques de transformation disponibles sur TensorFlow **sont écrites en C++ haute-performance**. Ainsi, le Python se contente de diriger le trafic entre les éléments et fournit des abstractions de programmation de haut niveau pour les connecter entre eux.

La détection et le suivi de véhicules sont basés sur des API prédéveloppées sur Tensorflow qui sont:

IV.2.1.1. Tensorfow object dectection API

La création de modèles d'apprentissage automatique précis, capables de localiser et d'identifier plusieurs objets dans une seule image, reste un défi majeur en vision par ordinateur. L'API de détection d'objet TensorFlow est une infrastructure open source construite sur TensorFlow qui facilite la construction, la formation et le déploiement de modèles de détection d'objet.

IV.2.1.2. Tensorflow object counting API

Le Tensorflow object counting API est un framework open source construit sur TensorFlow et Keras qui facilite le développement de systèmes de comptage d’objets.

IV.2.2. YOLO

YOLO (YOU ONLY LOOK ONCE) est un Algorithme basé sur la régression, couramment utilisé pour la détection d'objets en temps réel grâce a sa rapidité (capable de traiter 45 images par seconde) - au lieu de sélectionner des parties intéressantes d'une image, nous prédisons des classes et des cadres de sélection pour l'ensemble de l'image en une seule exécution de l'algorithme (un seul réseau de neurones sur toute l'image).

- ✓ **Fonctionnement**

L'algorithme applique un réseau de neurones à une image entière. Le réseau divise l'image en une grille S x S et crée des cadres de sélection, qui sont des boîtes dessinées autour des images et des probabilités prédites pour chacune de ces régions.
Puis, la combinaison de ces cadres et affichage d'encadrements des objets avec la classe à laquelle chaque objet appartient qui dépasse un seuil de probabilité donné (l'utilisateur peut changer ce seuil). [25]

IV.2.3. OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque de logiciels open source de vision informatique et d'apprentissage automatique, développée initialement par intel dans le but de fournir une infrastructure commune aux applications de vision par ordinateur.

La bibliothèque contient plus de 2500 algorithmes optimisés (classiques et avancés de vision par ordinateur et d'apprentissage automatique). Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer les actions humaines dans des vidéos, suivre les mouvements d'une caméra, suivre des objets en mouvement...etc

Il possède des interfaces C ++, Python, Java et MATLAB et prend en charge Windows, Linux, Android et Mac OS. [26]

IV.3. Implémentation

IV.3.1. Description générale

Notre système tout d'abord reçoit un flux vidéo à l'entrée, une fenêtre s'ouvrira avec une vue sur la route à surveiller et des paramètres seront affichés: le nombre de véhicules détectés, le nombre des véhicules qui ont dépassé la vitesse maximale ainsi que la vitesse, la couleur et le type de chaque véhicule détecté.

Ensuite et comme deuxième étape, la sauvegarde des images de chaque véhicule détecté avec excès de vitesse.

La troisième étape est la détection de plaques d'immatriculation et la reconnaissance des digits.

Et enfin, la sauvegarde de tout les résultats: la vitesse, le type, la couleur, l'heure et le matricule dans un fichier '.txt' dans la 4éme partie.

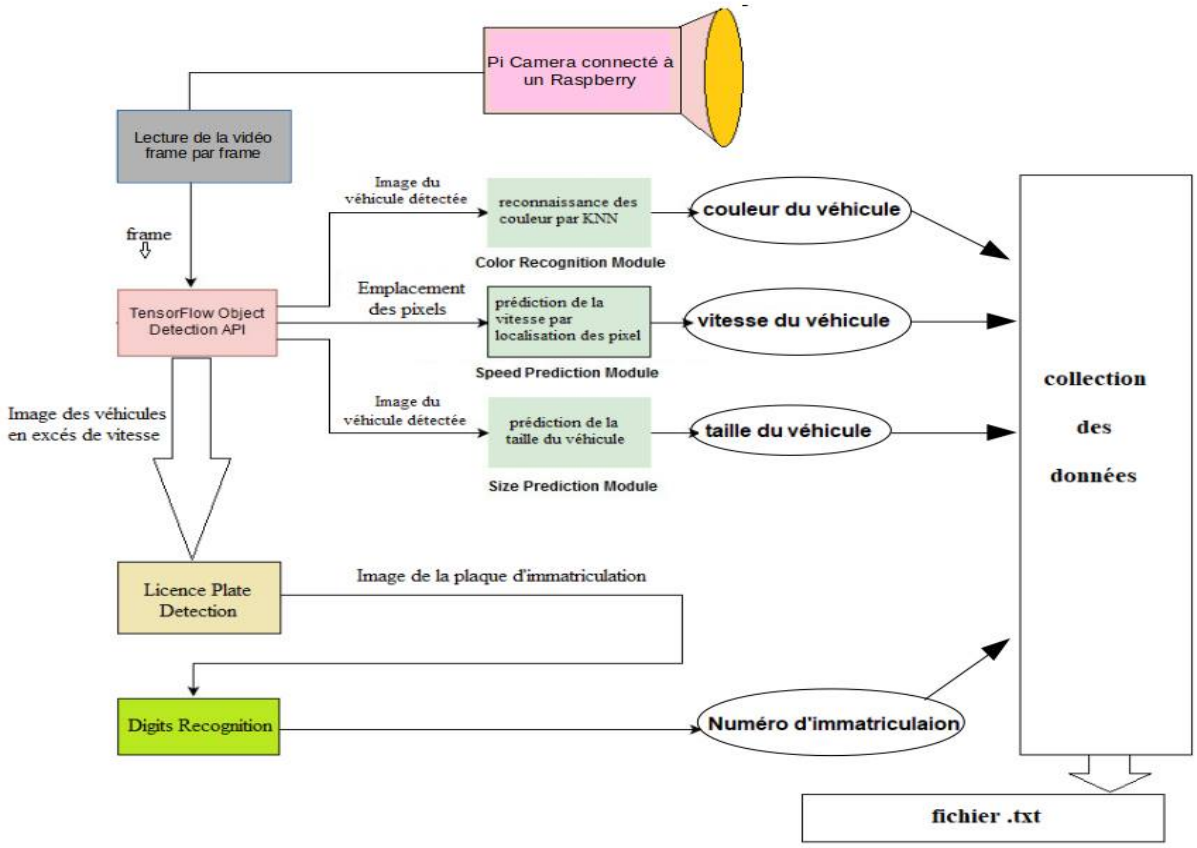


Figure 26: Schéma descriptif du système et modules implémenté

IV.3.2. Matériel utilisé

IV.3.2.1. Ordinateur portable (Lap Top)

Nous avons utilisé un ordinateur qui a comme caractéristiques:

- CPU: i3 2.53 GHz
- RAM de 4Go
- Disque dur HDD de 500Go
- OS: Linux Ubuntu 18.04 LTS

IV.3.2.2. Raspberry PI 3 B+

Comme caractéristiques de ce dernier nous distinguons:

- Un CPU 64 bit quad core ARM Cortex-A53 intégré et cadencé à 1,4 GHz
- Un contrôleur graphique Broadcom Videocore IV
- 1 Go de mémoire vive pour assurer fluidité à votre système
- 1 lecteur micro SD / SDHC
- 4 sorties USB 2.0
- 1 port RJ45 (Ethernet 10/100/300 Mbps)
- 1 port HDMI
- 1 audio Jack 3,5 mm
- GPIO 40 broches + 4 nouvelles broches pour le PoE
- Compatible Wifi 802.11 b/g/n/ac double bande 2,4 Ghz et 5 Ghz et Bluetooth 4.2 (Bluetooth Classique et LE)

IV.3.2.3. Camera PI

Les caractéristiques de cette camera sont:

- Capteur Omnivision 5647 avec objectif à focale fixe
- Capteur de 5 Mégapixels
- Résolution photo : 2592 x 1944
- Résolution vidéo maximum : 1080p
- mages par seconde maximum : 30fps
- Taille du module : 20 x 25 x 10mm
- Connexion par câble plat à l'interface 15-pin MIPI Camera Serial Interface (CSI) (Connecteur S5 du Raspberry Pi)

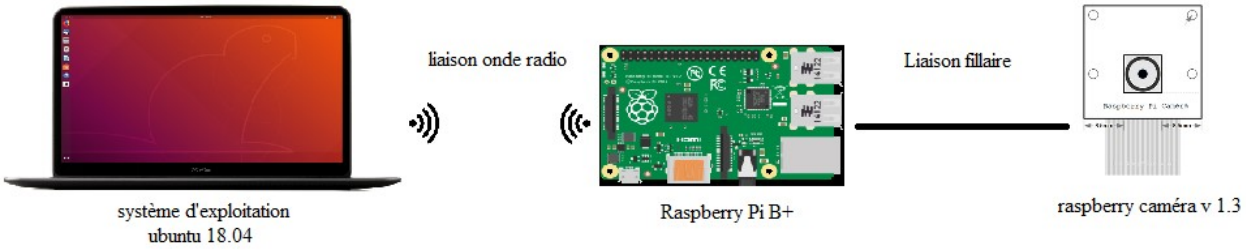


Figure 27: Matériels utilisés et leurs interconnections.

IV.3.3. Mise en place du système

IV.3.3.1. Sur ordinateur

En entrant ces commandes dans le terminal de la machine linux, chaque partie avec ses commandes:

- Python et pip (python version>3.3, pip3)

$ sudo apt-get install python3-pip python3-dev

- OpenCV

$sudo apt-get update
$sudo apt-get install -y build-essential
$sudo apt-get install -y cmake
$sudo apt-get install -y libgtk2.0-dev
$sudo apt-get install -y pkg-config
$sudo apt-get install -y python-numpy python-dev
$sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev
$sudo apt-get install -y libjpeg-dev libpng-dev libtiff-dev libjasper-dev
$sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake pkg-config yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils

- OpenCV version 2.4.11

$wget http://downloads.sourceforge.net/project/opencvlibrary/opencv-unix/2.4.11/opencv-2.4.11.zip

$unzip opencv-2.4.11.zip

$cd opencv-2.4.11

$mkdir release

$cd release

- Compilation

$cmake -G "Unix Makefiles" -D CMAKE_CXX_COMPILER=/usr/bin/g++

$ CMAKE_C_COMPILER=/usr/bin/gcc -D CMAKE_BUILD_TYPE=RELEASE -D

$ CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=ON -D

$ BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D

$ INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D

$ BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON -D

$ BUILD_FAT_JAVA_LIB=ON -D INSTALL_TO_MANGLED_PATHS=ON -D

$ INSTALL_CREATE_DISTRIB=ON -D INSTALL_TESTS=ON -D

$ ENABLE_FAST_MATH=ON -D WITH_IMAGEIO=ON -D

$ BUILD_SHARED_LIBS=OFF -D WITH_GSTREAMER=ON

$make all -j4 # 4 cores

$sudo make install

- Tensorflow

$pip3 install tensorflow-gpu (GPU support)

$pip3 install tensorflow-cpu (CPU support)

- Tensorflow ObjectDetection API

$sudo apt-get install protobuf-compiler python-pil python-lxml python-tk

$pip install --user Cython

$pip install --user contextlib2

$pip install --user jupyter

$pip install --user matplotlib

$pip install --user pillow

$pip install --user lxml

$sudo apt-get install protobuf-compiler

- Installation de COCO API

$git clone https://github.com/cocodataset/cocoapi.git

$cd cocoapi/PythonAPI

$make

$cp -r pycocotools <path_to_tensorflow>/models/research/

- Compiler protobuf

$cd tensorflow/models/research

$protoc object_detection/protos/*.proto --python_out=

- YOLO

En ce qui concerne yolo, il faut juste suivre les commandes mentionnées dans le site officiel https://pjreddie.com/darknet/yolo/ et mème apprendre à faire de l'apprentissage.

IV.3.3.2. Sur raspberry Pi

En créant un proxy avec l'outil VLC (le protocole utilisé entre le raspberry et l'ubuntu est le RTSP: Real Time Streaming Protocol).

IV.4. Scripts et Applications

IV.4.1. Partie 1: La détection et suivi des véhicules

Dans cette partie, nous allons implémenter une application qui s'appelle Vehicule_counting API tout en apportant des modifications qui peuvent nous être utile dans notre projet.

IV.4.1.1. Le scripte vehicle detection

C'est le premier module de notre système qui après avoir fait appel aux bibliothèques et aux autres scripts python et modules, il:

- démarre la détection en temps réel à partir du proxy déjà créé sur le raspberry.

```
#force cv2 to use udp protocole with rtsp
os.environ["OPENCV_FFMPEG_CAPTURE_OPTIONS"] = "rtsp_transport;udp"

# input video
cap = cv2.VideoCapture('rtsp://192.168.43.1:5554/playlist.m3u', cv2.CAP_FFMPEG)
```

- Appelle le SSD entrainé par mobilenet V1 COCO.

```
MODEL_NAME = 'ssd_mobilenet_v1_coco_2018_01_28'
MODEL_FILE = MODEL_NAME + '.tar.gz'
DOWNLOAD_BASE = \
    'http://download.tensorflow.org/models/object_detection/'
```

- Appelle la bibliothèque label map utils qui indexe les véhicules détectés au nom de la catégorie.

```
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
        max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)
```

- Ajout quelques détails à affiche (les boites englobantes, le scores de chaque détection, types de véhicules, numéro de véhicule détecté).

```
# Actual detection.
(boxes, scores, classes, num) = \
    sess.run([detection_boxes, detection_scores,
              detection_classes, num_detections],
              feed_dict={image_tensor: image_np_expanded})
```

- Visualisation des résultats: c'est la ou les informations de la détection vont être affichées.

```
# Visualization of the results of a detection.
(counter, csv_line) = \
    vis_util.visualize_boxes_and_labels_on_image_array(
    cap.get(1),
    input_frame,
    np.squeeze(boxes),
    np.squeeze(classes).astype(np.int32),
    np.squeeze(scores),
    category_index,
    use_normalized_coordinates=True,
    line_thickness=4,
    )
font = cv2.FONT_HERSHEY_SIMPLEX

total_passed_vehicle = total_passed_vehicle + counter
vehiclenum = 1
if speed != 'n.a.' :
    speedy = float(speed)
    if speedy > limite :

        speedcn = speedcn + counter

        cv2.putText(
            input_frame,
            'speed out of range: ' + str(speedcn),
            (10, 55),
            font,
            0.8,
            (0, 0xFF, 0),
            2,
            cv2.FONT_HERSHEY_SIMPLEX,
            )
        if counter != 0 :
            file = open("resultat.txt","a")
            file.write("vehicle" + str(vehiclenum) + ".jpg__speed_:_" + str(speedy) + "__color_:_" + color + "__type_:_" + size + "__heur/matricule_:_")
            file.write("\n")
            file.close()
            vehiclenum = vehiclenum + 1
```

- Ainsi, vu que c'est le principal module de cette partie, il fait appel aux autres modules (speed detection et color recognition) afin de tout visualiser sur écran.

Figure 28: Détection de véhicules sur autoroute et visualisation des résultats

IV.4.1.2. Speed and direction prediction module (estimation et comparaison de la vitesse)

Ce dernier est responsable de la prédiction de la vitesse dans la région d'intérêt (avec n'importe quelle direction) tout en se basant sur une équation mathématique simple introduite dans un script python speed_prediction.py.

```
if bottom > bottom_position_of_detected_vehicle[0]:
    direction = 'down'
else:
    direction = 'up'

if isInROI:
    pixel_length = bottom - bottom_position_of_detected_vehicle[0]
    scale_real_length = pixel_length * 44
    total_time_passed = current_frame_number - current_frame_number_list[0]
    scale_real_time_passed = total_time_passed * 24
    if scale_real_time_passed != 0:
        speed = scale_real_length / scale_real_time_passed / scale_constant
        speed = speed / 6 * 40
        current_frame_number_list.insert(0, current_frame_number)
        bottom_position_of_detected_vehicle.insert(0, bottom)
```

Ainsi, faire une comparaison entre la vitesse détectée et la vitesse limite (donnée par l'utilisateur), et si la vitesse détectée est supérieure à l'autre, ce script fait appel à deux autres scripts, le premier s'appelle crop_image.py et qui tout simplement rogne et recadre l'image en gardant seulement le véhicule détecté, le deuxième image_saver2.py va sauvegarder l'image de cette détection.

```
        if speed != 'n.a.':
            speedy = float(speed)
            if speedy > limitex and bot == 1 :
                image_saver_2.save_image(crop_img) # save detected vehicles speed out of range

    return (direction, speed, is_vehicle_detected, update_csv)
```

IV.4.1.3. Color recognition module (module de reconnaissance de couleur)

Color recognition module fait la classification des couleurs en s'appuyant sur un classificateur Machine Learning de KNN. La méthode de k voisins les plus proches, formé à l'aide de l'histogramme de couleurs R, V et B. Il peut classer le blanc, noir, rouge, vert, bleu, orange, jaune et violet.

Figure 29: photo capturée dans un essai d'application de classification des couleurs.

IV.4.2. Partie 2: La détection des plaques d'immatriculation

Dans cette partie on a utilisé le module LPD (licence plate detection), un modèle pré-entrainé qui utilise les réseau de neurone convolutif pour détecter les plaques d'immatriculation dans les images des véhicules déjà détectées dans la première partie de ce système.

Il est composé de 44 couches de convolution et 23 couches de pooling et une dernière couche entièrement connectée. [24]

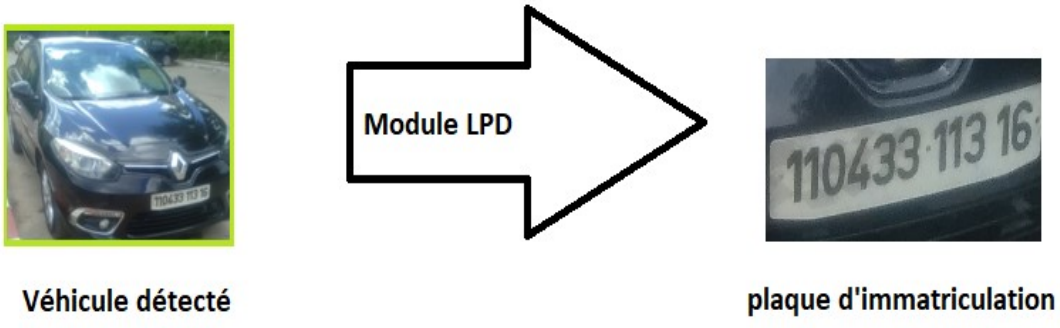


Figure 30: Fonctionnement du module LPD et détection des plaques d'immatriculation du véhicule détecté.

IV.4.3. Partie 3: Module de reconnaissance des chiffres

Ce module DR (digits recognition) est responsable de la détection et la reconnaissance des chiffres sur l'image déjà enregistré par le module de détection de plaque d'immatriculation (LPD). Les caractères détectés sont désordonnés lors de la détection, un traitement d'ordonnancement est nécessaire avant de les enregistrer sur un fichier.

Un troisième et dernier réseau neuronal convolutif est utilisé pour la détection et la reconnaissance de caractères sur la plaque d'immatriculation, il est composé de 44 couches de convolution et 23 couches de pooling et une dernière couche entièrement connectée qui donne en sortie une chaîne de chiffres présents dans la plaque d'immatriculation ainsi que l'heure de détection. [24]

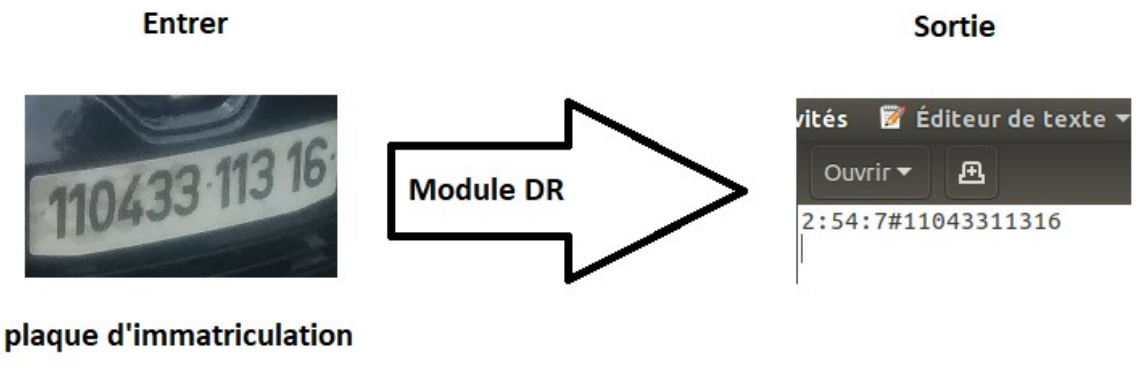


Figure 31: Reconnaissance des caractères sur la plaque d'immatriculation en utilisant le module DR (Digits Recognition).

IV.5. Conclusion

Dans ce chapitre, nous avons présenté les différents résultats que nous avons obtenus en intégrant les trois parties de notre système :

La première partie avec ses trois modules qui font la détection et la mesure de vitesse et la reconnaissance des couleurs,et qui nous a donné de bons résultats.

La deuxième partie fait la détection de la plaque d'immatriculation après la détection de véhicules qui se fait dans l'étape précédente.

La troisième et la dernière partie qui est capable de reconnaître les caractères sur la plaque d'immatriculation.

Conclusion générale et perspectives

Dès le début de notre projet, nous avons été intéressés par la mise en place d'un système intelligent pour la détection et la mesure de vitesse de véhicules.

Plusieurs techniques ont été développées dans le domaine de détection d'objets en mouvement, nous avons mentionné quelques unes, puis nous avons décrit les outils qu'on avait implémenté qui se basent sur la SSD qui est la plus réputée.

Nous avons rencontré quelques problèmes dans la partie de la détection de véhicules qui se sont manifestés sous forme de diminution du nombre de frames à traiter par secondes par rapport au FPS du flux vidéo de la caméra. Comme solution à ce problème, nous avons utilisé une plateforme plus performante qui est Google colab.

Notre système est maintenant capable de détecter des véhicules, fournir des informations sur leurs types, couleurs, et leurs directions puis, mesurer leurs vitesses, et identifier les véhicules qui étaient en excès de vitesse selon un seuil fixé par l'utilisateur.

Comme future améliorations, nous créerons une base de données pour la récolte et l'enregistrement des informations des véhicules détectés, et nous diminuerons la consommation des ressources afin de pouvoir faire l'implémentation sur des Smartphones sous système Android.

Bibliographie

[1] http://www.ons.dz/IMG/pdf/NAT31-12-2016.pdf (site officiel de l'ONS) (Mai 2019)

[2]Buch, N., Velastin, S. A., & Orwell, J. (2011). A Review of Computer Vision Techniques for the Analysis of Urban Traffic. IEEE Trans. on Intelligent Transportation Systems, 12(3), 920-939. https://doi.org/10.1109/TITS.2011.2119372(Mai 2019)

[3] H. Maître, " Le traitement des images ", Hermes Lavoisier IC2 2003.

[4] C.F.CREATIS, " Traitement et analyse d'image-ANIMAG ", Institut national des sciences, Université Claude Bernard, 2013-Lyon.

[5] Benchrife Ali, " Traitement d'image ", Université Abou Bakar Belkaid Tlemcen, 2008.

[6] http://www.nymphomath.ch/info/images/images.pdf site internet (Mai 2019)

[7] N. Laouar, M. S. Laraba, Détection d'un mouvement dans une séquence vidéo par filtres morphologiques, Mémoire de Master, ENP, Alger, 2009.

[8] Zeroual Djazia, "Implémentation D'un Environnement Parallèle Pour La Compression D'Image A L'Aide Des Fractales", Université de Batna, 2006.

[9] Y. V. D. R. E. &. D. J. F. Wang, «Tracking moving objects in video sequences,» In Conference on Information Sciences and Systems, vol.2, pp. 24-29, 2000, March.

[10] Cours en ligne disponible sur http://www.imedias.pro/cours-en-ligne.(Mai 2019)

[11] Introduction au traitement des images et à la stéréo-vision,cours http://perso.univ-lemans.fr/~berger/CoursStereoVision/co/Seuillage.html(Juin 2019)

[12] https://www.eyrolles.com/Chapitres/9782212110258/chap5.pdf chapitre 5, Les secrets de l'image vidéo, BELLAICHE Philippe, 2018

[13] Université Paris Descartes. Format vidéo. In http ://wiki.univ-paris5.fr/wiki/Format_vidéo (Juin 2019)

[14] Sarra BENFRIHA et Asma HAMEL, Segmentation d'images par cooperation région-contours, Mémoire Master Professionnel, université Kasdi Merbah Ouergla, 2016

[15] Medjahed Fatiha, Détection et suivi d'objets en mouvement dans une séquence d'images , Mémoire en vue de l'obtention du diplôme de magister, USTO, 2012.

[16] Asma Ouji. Segmentation et classification dans les images de documents numérisés. Autre [cs.OH].INSA de Lyon, 2012. Français. ?NNT : 2012ISAL0044?. ?tel-00749933?

[17] https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5072281-utilisez-ces-features-pour-classifier-des-images, 2019

[18] Mr Mokri Mohammed Zakaria, Classification des images avec les réseaux de neurones convolutionnels, Mémoire de Master, Université de Tlemcen, 2016

[19]https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf(Juin 2019)

[20] F. CHABOT, Analyse fine 2D/3D de véhicules par réseaux de neurones profonds, Thèse de doctorat: Université Clermont Auvergne, 2017.

[21] Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitiones guide to) Convolutional Neural Network of Sentence Classification.

[22] https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e(Juin 2019)

[23] https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab(Juin 2019)

[24] BENTAYEB Islam Chafa, BOUHRAOUA Abdelhalim, Estimation d'état de trafic routier en temps réel, mémoire de fin d'études (master), UMBBoumerdes, 2018.

[25] https://www.youtube.com/watch?v=NM6lrxy0bxs(Juin 2019) vidéo sur youtube de la présentation de YOLO par l'un de ses fondateurs

[26] https://opencv.org/ site officiel de OpenCV (Juin 2019)