



وزارة البحث العلمي والتعليم العالي

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET

DE LA RECHERCHE SCIENTIFIQUE

جامعة عبد الحميد بن باديس مستغانم

Université Abdelhamid Ibn Badis Mostaganem

كلية العلوم و التكنولوجيا

Faculté des Sciences et de la Technologie

DEPARTEMENT DE GENIE DES ELECTRONIQUE



N° d'ordre : M ...../GE/2024

## MEMOIRE

Présenté pour obtenir le diplôme de

### MASTER EN ELECTRONIQUE

Option : électronique des systèmes embarqués

Par

OULD MAMMAR Abderrahmane

BENALIOUA Malika

### Intitulé du sujet

Implémentation d'un système de stationnement intelligent  
dans un parking en utilisant le Machine Learning

Président : M<sup>r</sup>.MERAH Mostefa

Université de Mostaganem

Examinatrice : M<sup>elle</sup> BENDANI Djazia

Université de Mostaganem

Encadreur : M<sup>r</sup>.BENTOUMI Mohamed

Université de Mostaganem

Co-Encadreur : M<sup>r</sup> Benouali abdelhak

Université de Mostaganem

Co-Encadreur : M<sup>r</sup> Benaouali Mohamed

Université de Mostaganem

## Remerciement

*Nous remercions tout d'abord, Allah de nous avoir donné le courage mais surtout le temps d'accomplir ce travail.*

*Nos remerciements vont directement dans un premier temps à notre encadreur Monsieur M.Bentoumi, M.Benaouali et M.Benouali qui étaient à notre disposition et surtout de l'aide qu'ils nous ont apporté afin d'aboutir à ce travail et que Dieu le bénisse d'avantage.*

*Ensuite nous tenons à remercier et à témoigner nos reconnaissances à tout le personnel de l'Université Abdelhamid Ibn Badis de Mostaganem en particulier celui du département de génie électrique de la faculté des sciences et technologies. Sans oublier bien sur nos camarades avec lesquels nous avons su mettre en valeur le travail d'équipe durant nos recherches et réalisation mais surtout de l'attention, l'hospitalité et aussi la considération qu'ils ont démontré à notre égard durant ces trois années.*

*Enfin, la liste ne pouvant pas être exhaustive, permettez-nous d'adresser nos sincères remerciements à nos parents, familles, amis, proches et toute personne qui nous a encouragé et cru en nous dès le début*

## Dédicaces

*Nous tenons à dédier ce travail à nos chers parents qui n'ont jamais cessé de formuler des prières à notre égard, de nous soutenir et de nous épauler pour que nous puissions atteindre nos objectifs.*

*À nos sœurs et nos amis, pour leur soutien moral et leurs conseils précieux tout au long de nos études.*

*À tous ceux qui nous ont encouragés tout au long de notre cursus.*

أصبح دمج الذكاء الاصطناعي في حياتنا اليومية ضرورة في كافة القطاعات، وخاصة قطاع السيارات بسبب الكثافة المتزايدة لعدد المركبات.

تركز أنظمة س ي على جانبيين رئيسيين: اكتشاف موقع ركن السيارة والتعرف على حالة الموقع. التقسيم يجعل من الممكن اكتشاف أماكن وقوف السيارات. ومع ذلك، يتم التعرف على حالة المكان باستخدام أدوات الذكاء الاصطناعي

الهدف من مشروع نهاية الدراسة هذا هو تصميم نظام تلقائي للكشف عن أماكن وقوف السيارات للتحكم في الوصول إلى مواقف السيارات. يتم تنفيذه على راسبيري ضمن لينيكس ويتم برمجته بلغة بايثون. يتكون هذا النظام من وظيفتين رئيسيتين. الأول يعتني بالكشف والتعرف على حالة المساحة والثاني يستخدم لإدارة أجهزة الاستشعار والمركبات لضمان دخول وخروج المركبات من مراب السيارات. يهدف هذا النموذج الأولي إلى تحسين استخدام مواقف السيارات وتسهيل إدارتها.

## Résumé

L'intégration de l'intelligence artificielle dans notre vie quotidienne est devenue une nécessité dans tous les secteurs, en particulier le secteur automobile en raison de la densité croissante de nombre de véhicules.

Les systèmes SI se concentrent sur deux principaux aspects : la détection des emplacements de stationnement et la reconnaissance de l'état des emplacements. La segmentation permet de détecter la place de stationnement. Cependant, la reconnaissance de l'état de la place est réalisée en utilisant les outils de l'intelligence artificielle.

L'objectif de ce projet de fin d'étude est de concevoir un système de détection automatique des places de stationnement pour contrôler l'accès aux parkings. Il est implémenté sur un Raspberry Pi sous Linux et programmé en langage python. Ce système est constitué de deux grandes fonctions. La première s'occupe de la détection et la reconnaissance de l'état de la place et la deuxième sert à gérer les capteurs et les actionneurs pour assurer l'entrée et la sortie des véhicules du parking. . Ce prototype vise à optimiser l'utilisation des parkings et faciliter leur gestion.

Mots-clés : systèmes de stationnement intelligents, intelligence artificielle, Machine-Learning, SVM, HOG, Raspberry, python

## Abstract

The integration of artificial intelligence into our daily lives has become a necessity in all sectors, particularly the automotive sector due to the increasing density of the number of vehicles.

Systems SI focus on two main aspects: parking location detection and location status recognition. Segmentation makes it possible to detect the parking space. However, recognition of the state of the place is carried out using artificial intelligence tools.

The objective of this end-of-study project is to design an automatic parking space detection system to control access to parking lots. It is implemented on a Raspberry Pi under Linux and programmed in Python language. This system is made up of two main functions. The first takes care of the detection and recognition of the state of the space and the second is used to manage the sensors and actuators to ensure the entry and exit of vehicles from the parking lot. . This prototype aims to optimize the use of parking lots and facilitate their management.

## Table des matières

Remerciement .....	ii
Dédicaces .....	iii
ملخص .....	iv
Résumé .....	v
Abstract .....	vi
Table des matières .....	vii
Abréviations .....	x
Liste des figures .....	xi
Liste des tableaux .....	xiii
Introduction Générale .....	1
Problématiques .....	1
L'objectif de notre étude .....	1
Chapitre 1 : Généralités sur l'image .....	3
I.1 Introduction .....	3
I.2 Définition générale .....	3
I.2.1 L'image .....	3
I-2-2 L'image numérique .....	3
1.2.3 L'acquisition de l'image numérique.....	4
I.2.4 Caractéristiques de l'image numérique .....	6
I.3 Traitement d'image.....	6
I.3.1 Prétraitements : .....	6
I.3.2 L'analyse (le traitement) : .....	7
I.4 Conclusion :.....	11
Chapitre 2 : Généralité sur les systèmes de stationnement intelligent.....	12
II.1 Introduction.....	12
II.2 Définition d'un système de stationnement intelligent (SI).....	12
II.3 Historique.....	12
II.4 Le système de stationnement intelligent et l'intelligence artificielle .....	13
II.5 Le développement d'un système de stationnement intelligent .....	13
II.6 Composants du système de stationnement intelligent .....	14
II.7 Différent types des systèmes stationnement intelligents (SI) : .....	15

II.7.1 Systèmes intelligents de guidage de stationnement.....	15
II.7.2 Systèmes automatisés de stationnement .....	16
II.7.3 Reconnaissance de plaques d'immatriculation (LPR).....	16
II.7.4 Applications mobiles de paiement et de réservation .....	17
II.7.5 Stations de recharge pour véhicules électriques (VE) : .....	17
II.8 Avantages et Impacts des Systèmes de Stationnement Intelligent :.....	18
II.9 Le principe de fonctionnement des systèmes de stationnement intelligent .....	19
II.10 Défis et limitations : .....	20
II.11 Conclusion : .....	21
Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent .....	22
III .1 Introduction :.....	22
III.2 Partie matériel .....	22
III.2.1 Raspberry Pi.....	22
A - Historique .....	23
B - Evolution.....	23
C- Composants de base .....	24
III.2.2 Servomoteur SG90 .....	25
III.2.3 Caméra Raspberry Pi 2 .....	26
III.2.4 Capteur de proximité FC-51 .....	27
III.3 Partie logicielle .....	28
III.3.1 Linux :.....	28
III.3.2 Python :.....	28
A - OpenCV :.....	29
B - scikit-learn .....	30
III.3.3 L'intelligence artificielle (IA) .....	31
A - Mise en œuvre de l'IA .....	31
B - L'apprentissage automatique (Machine learning).....	32
C - Principaux Types d'Apprentissage .....	32
D - Étapes de la Mise en Œuvre du Machine Learning .....	32
III.4 Conclusion : .....	33
Chapitre 4 : Implémentation du Système SI.....	34
IV.1 Introduction : .....	34

IV.2 Description de notre système : .....	34
IV.3 Le matériel : Raspberry et ses périphériques : .....	34
IV.4 L'environnement logiciel (Software) : .....	35
IV .4.1 Logique de gestion des scénarios d'utilisation : .....	35
A - Procédure d'entrée : .....	35
B - Procédure de sortie : .....	36
C - Fonctionnement manuel : .....	36
IV.4.2 Phases de Développement .....	37
IV.4.2.1 La Création et entraînement du modèle de classification : .....	37
IV.4.2.2 La Détection en Temps Réel : .....	42
IV.5 Résultats et discussion : .....	43
IV.5.1 Les résultats du modèle de classification .....	43
IV.5.2 Les résultats de détection en temps réel .....	45
IV.5.2.1 La segmentation de la place de stationnement : .....	46
IV.5.2.2 La classification de la place : .....	46
IV .6 Les coûts : .....	47
IV.7 Conclusion : .....	47
Conclusion générale : .....	48
Perspectives : .....	48
Références Bibliographiques : .....	49

## Abréviations

**SI** : Stationnement Intelligent  
**CAN** : Conversion Analogique-Numérique  
**IoT** : Internet Des Objets  
**IA** : Intelligence Artificielle  
**LPR** : Reconnaissance de Plaques d'Immatriculation  
**VE** : Véhicules électriques  
**GPS** : Système Mondial de Positionnement  
**FPGA** : Field-Programmable Gate Array  
**RAM** : Random-Access Memory  
**IHM** : Interfaces Homme-Machine  
**HDMI** : High-Définition MultiMedia Interface  
**CPU** : Central Processing Unit  
**GPIO** : General-Purpose Input/Output  
**USB** : Universel Serial Bus  
**4Kp60** : Résolution (4000 Pixels 60 Hz)  
**LAN** : Local Area Network  
**PoE HAT** : Power Over Ethernet Hardware Attaché on Top  
**PWM** : Modulation de Largeur d'Impulsions  
**CMOS** : Complémentaire Métal Oxyde Semi-conducteur  
**Open CV** : Open Source Computer Vision  
**B.e.m** : Bouton Entrée Manuel  
**HOG** : Histogramme des Gradients Orientés  
**SVM** : Support Vecteur Machine  
**KNN** : k plus proches voisins

## Liste des figures

Figure 1 : Représentation d'une image numérique .....	4
Figure 2 : le processus d'échantillonnage d'un signal .....	5
Figure 3 : Le processus de la quantification d'un signal .....	5
Figure 4 : exemple de la segmentation morphologique .....	8
Figure 5 : exemple de la segmentation par couleur .....	8
Figure 6 : exemple sur la segmentation par seuillage .....	9
Figure 7 : exemple sur la segmentation par contour .....	10
Figure 8 : résultat de l'application du filtre de canny .....	10
Figure 9 : les composants du système stationnement intelligent.....	15
Figure 10 : le système de guidage de stationnement .....	15
Figure 11: exemple de système de stationnement automatisé .....	16
Figure 12 : Exemple de reconnaissance de plaques d'immatriculation .....	16
Figure 13 : exemple de réservation par des applications .....	17
Figure 14 : exemple de solutions de stationnement intelligent incluant des stations de recharge pour véhicule électrique (VE) .....	17
Figure 15 : le pourcentage d'Impacts les Systèmes SI à la vie quotidienne .....	18
Figure 16 : Raspberry PI 4 model B .....	23
Figure 17 : Evolution du Raspberry PI .....	24
Figure 18 : Composants de base du Raspberry Pi 4 .....	25
Figure 19 : un servomoteur SG90 .....	25
Figure 20 : un servomoteur contrôlé par un raspberry .....	26
Figure 21 : Caméra Raspberry Pi 2 .....	27
Figure 22 : Capteur de proximité FC-51 .....	27
Figure 23 : Interface système d'exploitation Linux-Debian .....	28
Figure 24 : Logo Python .....	29
Figure 25 : Localisation et suivi des véhicules avec OpenCV .....	30
Figure 26 : Les différentes caractéristiques de scikit-learn.....	31
Figure 27 : maquette de présentation de notre système de stationnement intelligent ....	34
Figure 28 : Schéma électronique du système SI.....	35
Figure 29 : Organigramme de fonctionnement du système SI.....	36
Figure 30 : Les différents étapes de développement de notre système SI .....	37
Figure 31 : exemple d'une image dans la base de données .....	38

## Liste des figures

---

Figure 32 : Extraction des caractéristiques par le HOG (cellule=4x4) pour une place occupée.....	39
Figure 33 : Extraction des caractéristiques par le HOG (cellule=4x4) pour une place vide .....	39
Figure 34 : segmentation par la détection de contour.....	43
Figure 35 : image segmentée.....	43
Figure 36 : La précision de classification avec SVM et KNN .....	44
Figure 37 : Résultats de classification des places avec HOG associée à SVM.....	45
Figure 38 : Les résultats de détection en temps réel .....	45
Figure 39 : Les résultats de détection en temps réel (Suite) .....	46

Liste des tableaux

Tableau 1 : matrice de Confusion.....	41
Tableau 2 :exemple d'une matrice de Confusion .....	42
Tableau 3 : Résultats de la classification avec SVM et KNN .....	44
Tableau 4 : Les coûts .....	47

## Introduction Générale

Avec l'urbanisation croissante et l'augmentation du nombre de véhicules, trouver une place de stationnement dans les zones urbaines est devenu un défi quotidien pour de nombreux conducteurs. Le manque de solutions de stationnement efficaces entraîne non seulement une perte de temps et de carburant, mais contribue également à la congestion du trafic et à l'augmentation des émissions de gaz à effet de serre. Les solutions traditionnelles de stationnement, telles que les systèmes de parcmètres et les parkings à plusieurs niveaux, montrent leurs limites face à cette demande croissante. En réponse à ces défis, la mise en place de systèmes de stationnement intelligents, intégrant les technologies de l'Internet des objets (IoT) et de l'apprentissage automatique (machine learning), émerge comme une solution innovante et efficace.

### Problématiques

Dans le parking de la faculté des sciences et technologie (FST), le stationnement traditionnel engendre de multiples problèmes qui impactent à la fois les conducteurs (enseignants, étudiants, travailleurs et/ou visiteurs) ainsi que le fonctionnement de la faculté. Voici quelques-unes des principales problématiques liées au stationnement traditionnel :

La gestion inefficace des parkings due à l'utilisation inadéquate des espaces.

Le temps de recherche de stationnement qui conduit à la frustration des conducteurs face à la répétition quotidienne.

La congestion qui conduit avec le temps à la pollution causée par les émissions des gaz d'échappement.

Les problématiques liées au stationnement traditionnel sont nombreuses et variées. Mais le problème principal auquel est confronté un conducteur cherchant une place de stationnement est l'incertitude quant à la disponibilité d'une place dans un parking. Les méthodes traditionnelle, telles que la signalisation manuelle ou les compteurs de places de stationnement, peuvent être peu fiable et surtout ne fournissent pas d'information en temps réel au conducteur. Pour y faire face, il est essentiel de développer des solutions innovantes, telles que le stationnement intelligent qui peuvent informer en temps réel le conducteur de l'état de la place de stationnement.

### L'objectif de notre étude

Dans cette étude, nous envisageons de développer un système de stationnement intelligent (SSI) pour le parking de la FST, qui consiste à collecter des données sur la disponibilité des

---

## Introduction Générale

---

places de stationnement en temps réel, en utilisant des méthodes de traitement des images acquise par une ou plusieurs caméras installées sur le parking. On procède à ce traitement avec un Raspberry Pi, en exploitant des technologies de l'intelligence artificielle telles que l'apprentissage automatique pour détecter l'état de la place et informer le conducteur.

Le mémoire est structuré de la manière suivante :

### Chapitre 1 : *Généralités sur l'image.*

Dans ce chapitre nous abordons une vue générale sur l'image, plus précisément l'image numérique.

### Chapitre2 : **Les systèmes de stationnement intelligent**

Ce chapitre se concentre sur des informations globales sur l'efficacité et les impacts du stationnement intelligent dans la gestion urbaine du trafic.

### Chapitre3 : **Les Composants du Systèmes de Stationnement Intelligent**

Dans ce chapitre, nous détaillons sur le Raspberry PI, ainsi que le langage de programmation, en donnant quelques détails techniques et les modèles d'apprentissage.

### Chapitre4 : **Implémentation de système**

Dans ce chapitre, nous allons présenter la mise en œuvre du système développé SI

## Chapitre 1 : Généralités sur l'image

### I.1 Introduction

Les images constituent l'une des solutions les plus importantes qu'on utilise pour transmettre et livrer le savoir et l'information depuis l'aube de l'humanité, dans la mesure où une image à elle seule peut réunir une quantité énorme d'informations. Un système de traitement d'images se compose principalement des fonctions suivantes : acquisition d'images, un prétraitement pour son amélioration, ensuite pour analyser l'image et pour arriver à une description synthétique de l'information brute incluse dans l'image. Dans ce chapitre nous allons introduire quelques notions générales dans le domaine du traitement d'image. Nous allons aussi présenter les notions nécessaires à la compréhension de l'image et ses caractéristiques.

### I.2 Définition générale

#### I.2.1 L'image

L'image définit une reproduction ou une représentation analogique exacte d'une scène réelle, elle est aussi désignée comme une scène tridimensionnelle sur un support en deux dimensions. Elle contient en chaque point, une intensité lumineuse, qui peut être représentée sous forme d'une fonction  $f$  avec deux variables  $x$  et  $y$  qui présentent les coordonnées linéaires d'un point de l'image, la fonction  $f$  présente l'intensité lumineuse définie sur un domaine bien défini. Dans le cas où c'est une image analogique donc est inexploitable par l'ordinateur, elle nécessite sa numérisation à l'aide de diverses méthodes.

#### I-2-2 L'image numérique

L'image numérique est une représentation visuelle d'un ou plusieurs objets qui peuvent être naturels ou artificiels. Cette dernière est traitée et stockée en binaire. Notons aussi que c'est une grille composée d'un ensemble fini d'éléments, appelés « Picture Élément », ou pixels (Voxels en 3D). Elle est représentée mathématiquement sous forme d'une fonction à deux dimensions de l'intensité lumineuse,  $I(x, y)$ , où  $x$  et  $y$  sont des coordonnées spatiales, et la valeur de  $I$  en  $(x, y)$  est proportionnelle à la luminosité de l'image en ce point (pixel). [1][2][3]

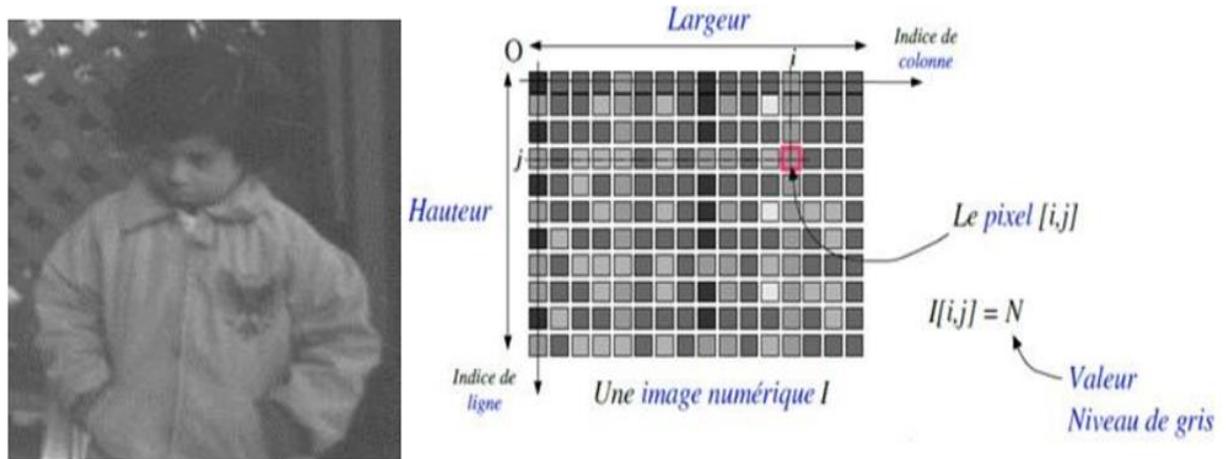


Figure 1 : Représentation d'une image numérique

### 1.2.3 L'acquisition de l'image numérique

L'acquisition d'image est un chaînon important dans la chaîne de production d'image numérique. L'acquisition consiste en un dispositif optique, généralement une caméra ou un scanner, qui capte une onde (lumière) d'une scène puis la numérise. L'objectif de la numérisation est de transformer le signal analogique qui contient une quantité infinie d'amplitudes en un signal numérique contenant lui une quantité finie de valeurs. Le passage de l'analogique au numérique se fait en 3 étapes successives : l'échantillonnage, la quantification ou conversion analogique-numérique (CAN) et le codage. [4]

- **L'échantillonnage** : L'échantillonnage consiste à prélever les valeurs d'un signal à intervalles réguliers définis par le théorème de Shannon. Il produit une suite de valeurs discrètes nommées échantillons. Il a une relation avec les pixels de l'image. Le nombre total de pixels dans une image peut être calculé comme suit :

$$\text{Pixels} = \text{nombre total de lignes} \times \text{nombre total de colonnes}$$

Par exemple, lorsque nous avons un total de 36 pixels, cela signifie que nous avons une image carrée de 6x6. Comme nous le savons en échantillonnage, plus d'échantillons aboutissent finalement à plus de pixels. Cela signifie donc que de notre signal continu, nous avons prélevé 36 échantillons sur l'axe x. Cela fait référence à 36 pixels de cette image. De plus, le nombre d'échantillons est directement égal au nombre de capteurs sur la matrice CCD. (Figure 2). [5]

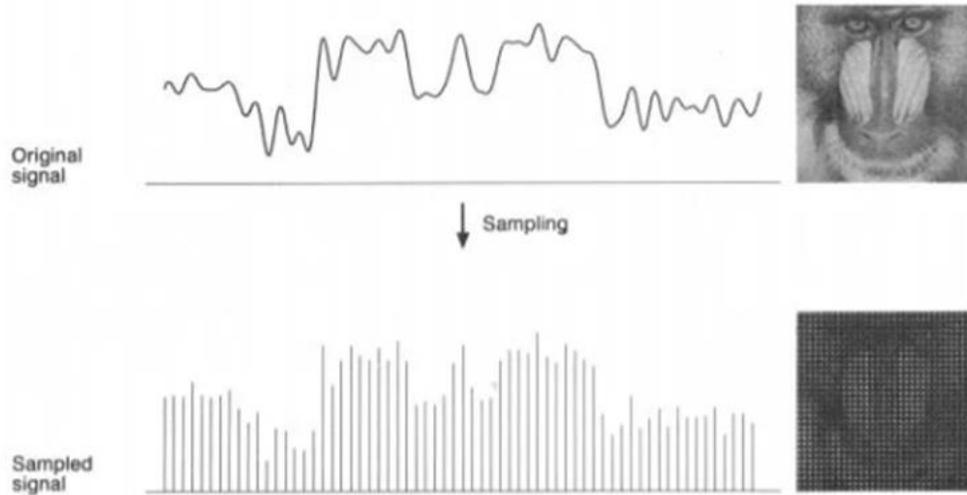


Figure 2 : le processus d'échantillonnage d'un signal

- **La quantification** : La quantification est un processus de transformation d'une image échantillonnée à amplitude réelle en une image dont les amplitudes ne prennent qu'un nombre fini de valeurs distinctes. Dans le cadre du processus de quantification, les valeurs d'amplitude de l'image sont numérisées. En termes simples, lorsque vous quantifiez une image, vous divisez en fait un signal en quanta (partitions). (figure 3). [6]

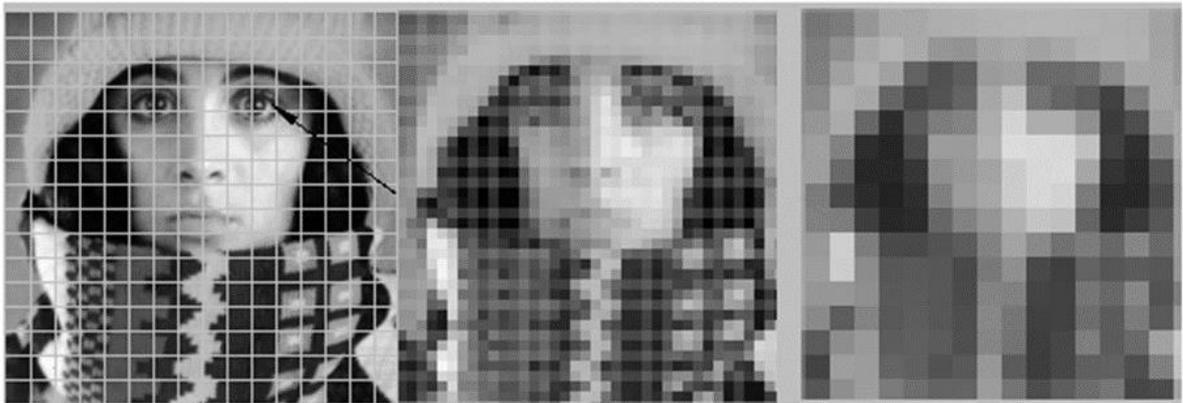


Figure 3 : Le processus de la quantification d'un signal

- **Le codage** :

La représentation numérique d'une image nécessite un très grand nombre de bits. L'objectif du codage d'image est de réduire ce nombre autant que possible et de reconstruire une copie fidèle de l'image originale. Les premiers efforts de codage d'images, uniquement guidés par la théorie de l'information, ont conduit à une multitude de méthodes. Le taux de compression a atteint un niveau de saturation d'environ 10:1 il y a quelques années. Les progrès récents dans l'étude du mécanisme cérébral de la vision et de l'analyse de scènes ont ouvert de nouvelles perspectives en matière de codage d'images. La sensibilité

directionnelle des neurones dans la voie visuelle combinée au traitement séparé des contours et des textures a conduit à une nouvelle classe de méthodes de codage capables d'atteindre des taux de compression aussi élevés que 100:1.

### 1.2.4 Caractéristiques de l'image numérique

Une image numérique est caractérisée par : [7] [8]

- **La Taille** : Appelée aussi la définition, c'est le nombre de pixels constituant l'image. Cette dernière est composée d'une grille de pixels. Ces pixels sont autant de petits carrés porteurs d'une information de couleur élémentaire.
- **La Résolution** : La résolution d'une image est le nombre de pixels par pouce (dpi) qu'elle contient (1 pouce = 2.54 centimètres). Elle peut être exprimée en « ppm » (point par millimètre). Une résolution plus élevée signifie plus de détails sur l'image.
- **Le Poids** : Le poids représente la taille de l'image en octets. Il dépend de la définition (le nombre de pixels) et le codage de la couleur de l'image.
- **Le codage de la couleur** : Le pixel est codé par un ou plusieurs bits. Ce nombre de bits détermine la couleur de pixel (1 bit pour l'image noir et blanc, 8 bits pour l'image niveau de gris, 24 bits pour l'image couleur). C'est ce qu'on appelle le codage de la couleur.

## 1.3 Traitement d'image

Le traitement d'image numérique est l'ensemble d'outils et de techniques permettant de modifier une image numérique dans le but d'améliorer sa qualité d'en extraire des informations. Le traitement d'image est un traitement de signal 2D. L'étape de traitement d'image vient après celles d'acquisition dans la chaîne de production d'image. Parmi ses applications : la télédétection, l'imagerie médicale, les applications militaires, route intelligente, reconnaissance de caractères. [7]

Le traitement d'image se divise en deux étapes :

- Le prétraitement
- L'analyse

### 1.3.1 Prétraitements :

Les prétraitements sont un ensemble de méthodes qui s'appliquent sur l'image acquise afin de faciliter l'extraction des informations. L'image est souvent distordue ou dégradée à cause d'un éclairage non uniforme ou une perturbation lors de l'acquisition. Il existe 3 grandes familles de procédés pour corriger une image :

- **L'amélioration** : elle consiste à corriger l'aspect visuel de l'image c'est-à-dire une image plus contrastée. L'information contenue dans le pixel ne change pas. Parmi les

méthodes de l'amélioration : L'extension dynamique, la correction d'exposition, l'égalisation d'histogramme, le renforcement de contraste...

- **Le dé-bruitage** : il consiste à restaurer l'information (la valeur de luminance originale) contenue dans le pixel. Cette information était infectée par des bruits (variation aléatoires) lors de l'acquisition ou la numérisation. Pour minimiser ce bruit, on applique les filtres cités ci-dessous : Le filtre moyenneur, le filtre gaussien, le filtre médian, le filtre de Nagao ...

### 1.3.2 L'analyse (le traitement) :

L'analyse des images est un processus de décomposition de l'image prétraitée dans le but de trouver, détecter et reconnaître les objets contenus dans la scène observée, afin d'avoir une description du contenu sémantique de la scène.

L'analyse d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années. La segmentation est l'un des processus de base du traitement d'image numérique et de la vision par ordinateur. Elle consiste à définir des régions présentant des similitudes internes, telles que similitudes statistiques, géométriques, de forme... Une image est segmentée en régions pour permettre un traitement et une analyse indépendants de chaque région en fonction d'une politique ou d'un objectif de traitement. Les régions couvrent des zones plus petites que l'image globale mais généralement plus grandes que les entités ponctuelles locales, de sorte qu'une application peut utiliser des métriques régionales et locales de petits points d'intérêt (key points) en tant que signature d'objet.

La segmentation se base sur plusieurs méthodes. Parmi ces méthodes :

- Segmentation morphologique :

Dans certaines applications d'analyse d'images et de vision industrielle, telles que l'inspection des défauts industriels ou l'imagerie biomédicale, la segmentation basée sur le seuillage ou la détection des contours n'est pas suffisante car la qualité de l'image est insuffisante ou les objets inspectés se touchent ou se chevauchent. Dans de telles applications, la segmentation morphologique est une méthode efficace de segmentation d'images. La segmentation morphologique divise une image en fonction de la surface topographique de l'image. L'image est séparée en régions qui ne se chevauchent pas, chaque région contenant une particule unique.



Figure 4 : exemple de la segmentation morphologique

- Segmentation par couleur :

La segmentation des couleurs compare la caractéristique de couleur de chaque pixel avec les caractéristiques de couleur des pixels environnants ou avec un classificateur de couleurs entraîné pour segmenter une image en régions de couleur. Utilisez la segmentation des couleurs pour séparer les objets de couleur qui vous intéressent du fouillis d'arrière-plan.

Vous pouvez utiliser la segmentation des couleurs dans une grande variété d'applications de vision industrielle, telles que les suivantes :

Inspection : partitionnez une image en différentes régions en fonction de la couleur de la pièce dans chaque région.

Comptage : segmentez une image pour compter rapidement le nombre d'objets avec une composition de couleurs particulière. Les figures suivantes illustrent une segmentation des couleurs qui peut être utilisée pour compter le nombre de bouteilles contenant chaque couleur de liquide. (Voir figure 5 ci-dessous)



Figure 5 : exemple de la segmentation par couleur

- Segmentation par seuillage :

La segmentation par seuil est un processus utilisé pour séparer différentes régions d'une image en fonction de leurs valeurs d'intensité de pixels. En définissant un seuil d'intensité spécifique, les pixels supérieurs ou inférieurs à cette valeur sont classés en groupes distincts, ce qui donne une image binaire ou multi-niveaux. Cette technique est particulièrement utile

lorsqu'une distinction claire entre les objets souhaités et leurs arrière-plans peut être établie, comme l'identification du texte d'un document numérisé ou la segmentation de cellules dans une image microscopique.

Dans les applications pratiques, le choix de la valeur seuil appropriée peut être considéré comme un art et une science, car cela peut nécessiter un équilibre entre expérimentation et connaissance préalable des images traitées. Diverses méthodes, telles que le seuillage global, adaptatif ou multi-niveaux, sont parfois utilisées pour optimiser les résultats de segmentation. Une segmentation de seuil efficace peut améliorer les tâches d'analyse ou de classification en aval, permettant aux développeurs et aux professionnels techniques d'extraire des informations significatives de leurs données d'imagerie.

La figure 6 nous montre un exemple de la segmentation par seuillage.

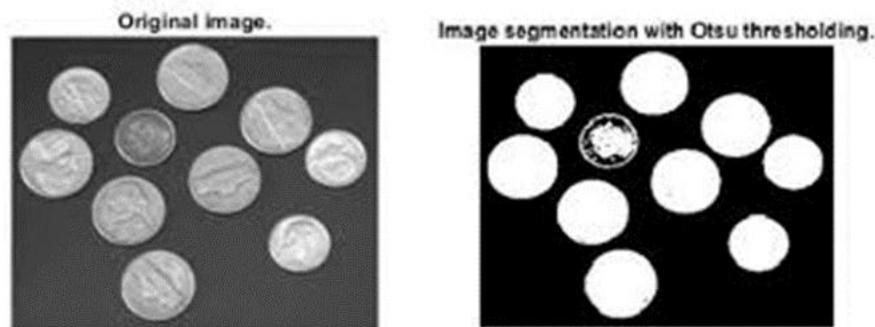


Figure 6 : exemple sur la segmentation par seuillage

- Segmentation par détection de contour.

La segmentation basée sur les bords est utilisée pour délimiter les objets dans une image en identifiant et en analysant les bords présents. Cette méthode part du principe que les limites des objets dans les images correspondent généralement à des changements brusques d'intensité, de couleur ou de texture. À la base, la segmentation basée sur les bords repose sur la détection de ces discontinuités et les utilise comme indicateurs pour segmenter différents objets ou caractéristiques de l'arrière-plan ou les uns des autres. Le succès de la segmentation basée sur les contours dépend de la qualité de la détection des contours et du contraste au sein de l'image. Cela met en évidence l'idée que, même si la technique est puissante, elle peut être remise en question par des images à faible contraste ou dans des conditions bruyantes où les bords ne sont pas clairement définies. (figure 7).

Il y a plusieurs méthodes pour détecter les contours dans l'image. Ces méthodes sont des filtres qu'on applique sur l'image. Nous citons quelques-uns : Filtre de *Prewitt*, filtre de *Sobel*, filtre de *Canny*...



Figure 7 : exemple sur la segmentation par contour

On peut citer en dessous **Le filtre de Canny** qui est le plus utilisé pour détecter les contours d'une l'image.

**Le filtre de Canny** est l'un des algorithmes les plus connus pour la détection de contour. Il a été développé par *John Canny* en 1986. Il se décompose en étapes suivantes :

- Minimiser le bruit en appliquant un filtre gaussien.
- Calculer le gradient d'intensité de l'image.
- Suppression du non-maxima pour éliminer les parasites lors de la détection de contour.
- Appliquer un double seuillage.
- Différencier les contours par seuillage hystérésis.

L'algorithme de détection de contour de *Canny* est contrôlé par deux paramètres : • La taille du filtre appliqué (ex : filtre gaussien de 3x3, 5x5 ou 7x7). • Les deux seuils d'hystérésis.

La figure ci-dessous nous montre le résultat de l'application d'un filtre de canny.

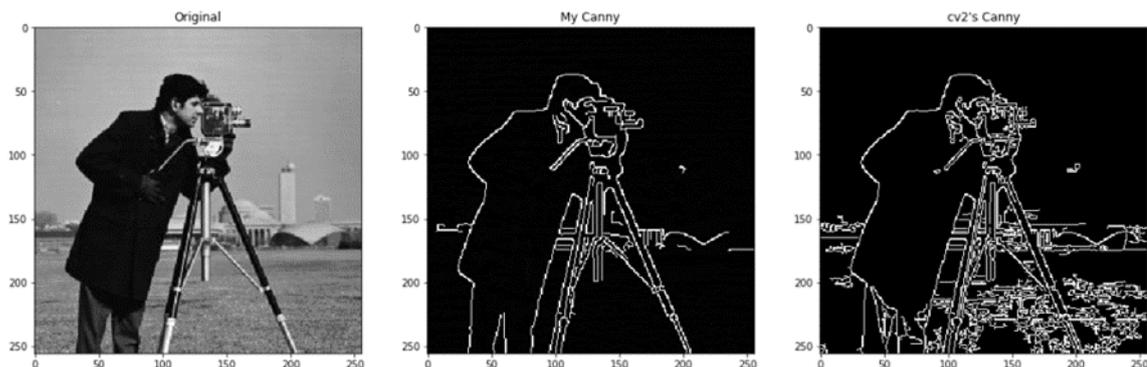


Figure 8 : résultat de l'application du filtre de canny

### I.4 Conclusion :

Dans ce chapitre nous avons présentés une vue générale sur l'image, plus précisément l'image numérique, où on a abordé les caractéristiques de cette dernière suivie par ces techniques de traitement.

Dans notre cas ; le traitement d'image modernise et améliore l'expérience de stationnement, offrant des solutions innovantes pour les défis quotidiens rencontrés dans la gestion des parkings. En alliant technologie et infrastructure, les systèmes de stationnement deviennent plus intelligents, plus sûrs et plus efficaces.

## Chapitre 2 : Généralité sur les systèmes de stationnement intelligent

### II.1 Introduction

La gestion efficace des parkings privée est un défi majeur dans les établissements qui contiennent un grand nombre de voitures. Les conducteurs qui sont notamment des employés de l'établissement passent souvent un temps précieux à la recherche d'une place de stationnement disponible, ce qui entraîne de la frustration et de la congestion. Pour résoudre ces problèmes, il est crucial de développer des systèmes intelligents capables de détecter la disponibilité des places de stationnement en temps réel.

Les systèmes de stationnement intelligent (SI) font partie de notre quotidien et sont l'avenir des modes de stationnement. Ils offrent des solutions à certaines problématiques comme l'optimisation de l'utilisation de l'espace, et de la réduction du temps de recherches.

Dans ce chapitre on va présenter les différents types du système de stationnement intelligent ainsi que leur implémentation.

### II.2 Définition d'un système de stationnement intelligent (SI)

Un système de stationnement intelligent est une solution technologique avancée conçue pour détecter en temps réel la disponibilité des places de stationnement. Ces systèmes utilisent diverses technologies telles que les capteurs, l'Internet des objets (IoT), les applications mobiles et les logiciels de gestion pour fournir des informations en temps réel sur la disponibilité des places de stationnement et faciliter la gestion des parkings.

### II.3 Historique

L'évolution du stationnement intelligent, de ses débuts avec les parcmètres mécaniques dans les années 1930 à son état actuel intégrant des technologies avancées, a transformé la gestion urbaine et l'expérience des conducteurs. Les premiers parcmètres, tels que celui de Carl Magee en 1935, étaient à monnaie et étaient suivis par les parcmètres électromécaniques des années 1950-1960, qui étaient plus fiables grâce à des composants électriques. Les parcmètres électroniques avec affichages numériques ont rendu les paiements plus flexibles dans les années 1970-1980. Dans les années 1990, des systèmes de stationnement automatisés ont été introduits, ce qui a simplifié l'accès et le paiement par cartes de crédit. L'arrivée des capteurs au sol et des réseaux IoT dans les années 2000 a rendu possible la détection en temps réel des véhicules et la transmission des données à des systèmes centralisés. Au début des années 2010, l'utilisation d'applications mobiles et de paiement en ligne a bouleversé la façon dont les automobilistes

trouvaient et réglaient leur stationnement. Les années 2020 ont été marquées par l'utilisation de l'intelligence artificielle et de l'apprentissage automatique afin d'analyser les données, anticiper la demande et ajuster les tarifs de manière dynamique. Des exemples concrets tels que SFpark à San Francisco et le système de Smart Parking à Barcelone démontrent des résultats encourageants, tels que la diminution du temps nécessaire pour trouver des places et l'amélioration de la fluidité du trafic [10].

### II.4 Le système de stationnement intelligent et l'intelligence artificielle

Le passage du stationnement traditionnel au stationnement intelligent, enrichi par l'intelligence artificielle (IA), transforme radicalement l'expérience des conducteurs et la gestion urbaine. Les systèmes de stationnement intelligents, en incorporant des technologies comme des capteurs au sol, des caméras intelligentes et des applications mobiles, permettent de diminuer considérablement le temps nécessaire pour trouver des places, d'optimiser l'utilisation des espaces et d'améliorer la sécurité. Par exemple, des villes importantes telles que San Francisco, grâce à son projet SFpark, ont remarqué une diminution de 30% du temps de recherche d'une place. Encore une fois, l'emploi de la vision par ordinateur pour repérer les places libres à Barcelone a permis d'améliorer la fluidité du trafic et de diminuer les émissions de CO<sub>2</sub>.

L'intégration de l'IA permet une analyse prédictive de la demande, une tarification dynamique, et une surveillance renforcée, tout en proposant un accompagnement sur mesure et des options de paiement numérique aux conducteurs. Bien que les coûts d'implémentation, la sécurité des données et l'acceptation par les utilisateurs soient des défis, les avantages sont importants. La transition vers un stationnement intelligent fait partie d'une vision plus globale des smart cities, où l'amélioration de la qualité de vie est mise en avant en tant que priorité. Grâce aux progrès technologiques et l'intégration future des véhicules autonomes, le stationnement intelligent est prévu pour devenir un élément clé des infrastructures urbaines modernes.

### II.5 Le développement d'un système de stationnement intelligent

L'implémentation d'un système de stationnement intelligent implique plusieurs étapes clés pour assurer son succès. Voici une vue d'ensemble du processus d'implémentation [11] :

- **Évaluation des besoins** : Dans un premier temps, il est nécessaire d'évaluer les exigences particulières de parking. Cela englobe la détection des défis majeurs cité précédemment tels

que la non-viabilité des places disponibles, l'utilisation inefficace de l'espace, les inégalités d'accès, les problèmes de sécurité, et ainsi de suite.

- **Définition des objectifs** : Dans ce cas il est nécessaire d'Établir des objectifs clairs pour le système de stationnement intelligent, en tenant compte des besoins identifiés. Ces objectifs pourraient inclure l'optimisation de l'utilisation de l'espace, l'utilisation efficace de l'espace, l'amélioration de la sécurité, etc.
- **Sélection des technologies** : Identification des technologies appropriées pour répondre aux besoins et aux objectifs définis. Cela pourrait inclure des capteurs de stationnement, des caméras de surveillance, des systèmes de guidage pour les conducteurs, des applications mobiles de réservation et de paiement, etc.
- **Planification du déploiement** : Élaboration d'un plan détaillé pour le déploiement du système, y compris la disposition des équipements, les exigences en matière d'infrastructure, les coûts associés et le calendrier de mise en œuvre.
- **Installation des équipements** : l'installation des équipements nécessaires selon le plan défini. Cela peut inclure l'installation de capteurs de stationnement sur chaque place, la mise en place de caméras de surveillance, la configuration des panneaux d'affichage pour les conducteurs, etc.
- **Intégration des systèmes** : Intégration des composants du système de stationnement intelligent avec une manière transparente pour assurer un fonctionnement harmonieux. Cela peut nécessiter l'intégration de logiciels de gestion des données, de systèmes de surveillance, d'applications mobiles, etc.
- **Test et ajustement** : Effectuez des tests complets du système pour s'assurer qu'il fonctionne comme prévu. Identifiez les problèmes potentiels et apportez les ajustements nécessaires avant le déploiement complet.
- **Évaluation continue** : Surveillez en permanence les performances du système de stationnement intelligent et recueillez les commentaires des utilisateurs pour identifier les possibilités d'amélioration continue.

### II.6 Composants du système de stationnement intelligent

Les systèmes SI se composent de trois blocs : capteurs, système de commande (microcontrôleurs) et des actionneurs.

Les systèmes SI se basent essentiellement sur une caméra qui est liée au système d'analyse et de commande qui peut être un ordinateur ou un microcontrôleur. Les actionneurs peuvent être des barrières, des significations lumineuses.

## Chapitre 2 : Généralité sur les systèmes de stationnement intelligent

La figure ci-dessous illustre en détails les composants du système stationnement intelligent.

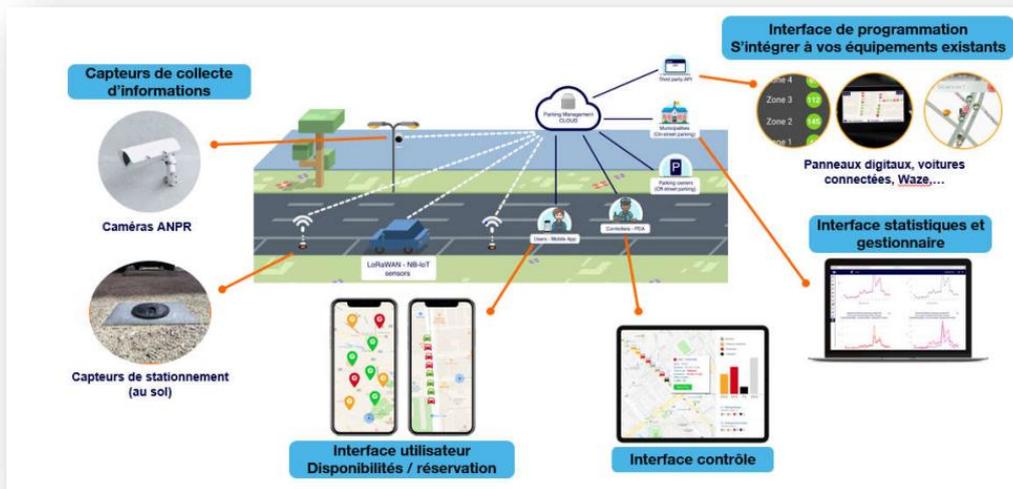


Figure 9 : les composants du système stationnement intelligent.

### II.7 Différent types des systèmes stationnement intelligents (SI) :

#### II.7.1 Systèmes intelligents de guidage de stationnement

Ces systèmes utilisent des capteurs ou des caméras pour surveiller la disponibilité des places de stationnement en temps réel et fournir aux conducteurs des informations sur les places disponibles via des applications mobiles, des panneaux électroniques ou des systèmes de navigation embarqués [12] (Figure 10).

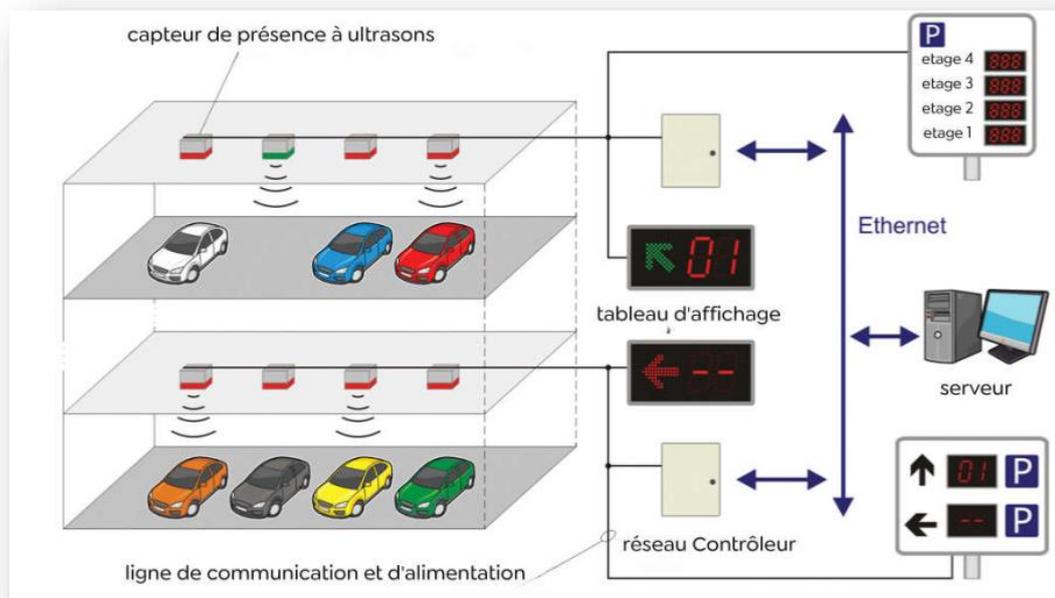


Figure 10 : le système de guidage de stationnement

### II.7.2 Systèmes automatisés de stationnement

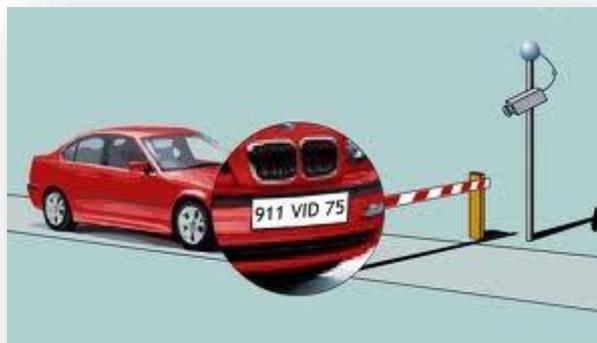
Les systèmes de stationnement automatisés utilisent la robotique et l'automatisation pour garer et récupérer les véhicules sans besoin de conducteurs humains. Ces systèmes peuvent optimiser l'utilisation de l'espace et réduire le temps et l'effort nécessaires pour le stationnement comme la montre la figure ci-dessous [13].



*Figure 11: exemple de système de stationnement automatisé*

### II.7.3 Reconnaissance de plaques d'immatriculation (LPR)

La technologie de LPR permet la reconnaissance automatique des numéros de plaques d'immatriculation, permettant une gestion efficace des entrées et sorties dans les installations de stationnement. Elle peut également être utilisée à des fins de facturation et pour renforcer la sécurité [14]. La figure 12 nous montre un exemple de reconnaissance des plaques d'immatriculation.



*Figure 12 : Exemple de reconnaissance de plaques d'immatriculation*

### II.7.4 Applications mobiles de paiement et de réservation

Les applications mobiles permettent aux utilisateurs de trouver des places de stationnement, de les réserver à l'avance et de faire des paiements électroniques (Figure 13). Cela élimine le besoin de billets physiques ou de transactions en espèces, rationalisant le processus de stationnement. [15]



*Figure 13 : exemple de réservation par des applications*

### II.7.5 Stations de recharge pour véhicules électriques (VE) :

Avec la montée en puissance des véhicules électriques, les solutions de stationnement intelligent peuvent inclure des stations de recharge pour VE équipées de surveillance et de facturation pour soutenir la demande croissante de transport électrique (voir figure 14).



*Figure 14 : exemple de solutions de stationnement intelligent incluant des stations de recharge pour véhicule électrique (VE)*

En résumé, les solutions de stationnement intelligent visent à améliorer l'expérience globale de stationnement pour les conducteurs, elles permettent de réduire le temps consacré à la recherche d'une place, et l'utilisation efficace de l'espace.

### II.8 Avantages et Impacts des Systèmes de Stationnement Intelligent :

Le stationnement intelligent, avec sa capacité à détecter la disponibilité des places de stationnement dans un parking, offre une gamme d'avantages. Voici quelques-uns des avantages et impacts les plus importants [16] [17] (figure 15) :

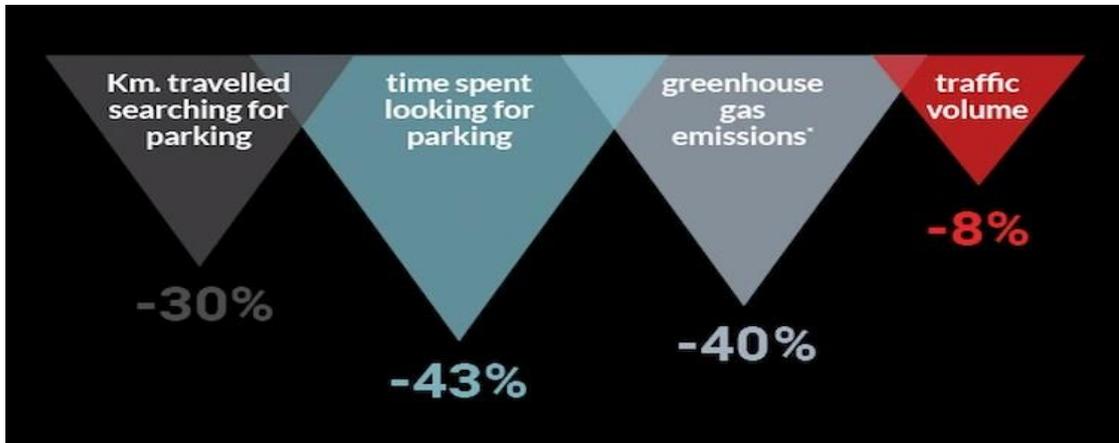


Figure 15 : le pourcentage d'Impacts les Systèmes SI à la vie quotidienne

- Réduction du temps de recherche : Les conducteurs consacrent souvent une grande partie de leur temps à la recherche d'une place de stationnement, ce qui peut causer de la frustration et une perte de temps. Le stationnement intelligent offre aux conducteurs des informations en temps réel sur la disponibilité des places, ce qui permet de trouver rapidement un emplacement adéquat, ce qui permet de réduire le temps de recherche.
- Optimisation de l'utilisation de l'espace : Le stationnement intelligent permet d'identifier en temps réel les places de stationnement disponibles, ce qui permet une utilisation plus efficace de l'espace sur un parking. Cela peut aider à réduire le gaspillage d'espace, optimiser le nombre de véhicules disponibles.
- Amélioration de l'expérience utilisateur : Les systèmes de stationnement intelligent offrent une expérience utilisateur améliorée en fournissant des informations en temps réel sur la disponibilité des places de stationnement, des options de réservation et de paiement pratiques, et des itinéraires optimisés vers les places de stationnement disponibles.
- Gestion plus efficace des parkings : Les informations sur l'utilisation des parkings recueillies grâce aux systèmes de stationnement intelligents facilitent une gestion plus efficace des parkings. Ces données peuvent être exploitées par les gestionnaires afin d'améliorer la répartition des places, de planifier la maintenance et de prendre des décisions éclairées concernant les investissements à venir.

- Réduction de la congestion : Lorsqu'il s'agit d'un parking public, les systèmes de stationnement intelligents facilitent la localisation des places de stationnement disponibles, ce qui permet aux conducteurs de réduire le temps nécessaire pour trouver une place et contribue à réduire la congestion routière.

### II.9 Le principe de fonctionnement des systèmes de stationnement intelligent

Le principe de fonctionnement des systèmes de stationnement intelligent repose sur l'intégration de diverses technologies pour surveiller, gérer et optimiser l'utilisation des places de stationnement. Voici les principaux éléments de fonctionnement de ces systèmes :

- Détection de la disponibilité des places :

Utilisation de capteurs au sol, de caméras intelligentes ou de technologies de détection avancées pour surveiller en temps réel l'occupation des places de stationnement.

- Collecte et transmission des données :

Les capteurs et les caméras collectent des données sur l'occupation des places, qui sont ensuite transmises à un système centralisé via des réseaux IoT ou sans fil.

- Analyse des données et prévision de la demande :

Les données collectées sont analysées à l'aide d'algorithmes d'intelligence artificielle pour prévoir la demande de stationnement à différents moments de la journée et dans différentes zones.

- Affichage des places disponibles

Les informations sur les places disponibles sont affichées aux conducteurs via des panneaux d'affichage électroniques, des applications mobiles ou des sites web, leur permettant de trouver rapidement une place de stationnement.

- Guidage des conducteurs :

Les conducteurs sont guidés vers les places disponibles à l'aide d'indications visuelles ou sonores, souvent intégrées dans les applications mobiles ou les systèmes de navigation GPS.

- Optimisation de l'utilisation des espaces :

Les données sur l'utilisation des places de stationnement sont utilisées pour optimiser l'allocation des espaces, en ajustant les tarifs de stationnement en fonction de la demande et en dirigeant les conducteurs vers les zones moins occupées.

- Surveillance et gestion en temps réel :

Les opérateurs de stationnement peuvent surveiller et gérer les parkings en temps réel à partir d'un tableau de bord centralisé, en répondant aux problèmes tels que les infractions de stationnement ou les pannes de capteurs.

Grâce à la combinaison de ces divers éléments, les systèmes de stationnement intelligent proposent une solution globale visant à améliorer l'efficacité du stationnement, diminuer la congestion urbaine et améliorer l'expérience client.

### II.10 Défis et limitations :

Malgré les nombreux avantages des systèmes de stationnement intelligent, Il y'a un certain nombre de difficultés possibles que les systèmes SI ne peuvent prendre en considération [14].

Voici quelques-uns des principaux défis et limitations associés à ces systèmes :

- Coûts d'installation et de maintenance : L'installation initiale des infrastructures nécessaires, telles que les capteurs de stationnement, les panneaux de signalisation et les systèmes de gestion des données, peut être coûteuse. De plus, les coûts continus de maintenance et de mise à niveau des systèmes doivent également être pris en compte.
- Pauvre résolution de l'image, généralement parce que la place est trop loin mais parfois résultant de l'utilisation d'une caméra de faible qualité.
- Mauvais éclairage et de faible contraste en raison de surexposition, réflexion ou ombres
- Vie privée et sécurité des données : Les systèmes de stationnement intelligent collectent et traitent souvent des données personnelles, telles que les mouvements des véhicules et les habitudes de déplacement des conducteurs. Il est crucial de mettre en place des mesures de protection de la vie privée et de sécurité des données pour garantir la confidentialité et la sécurité des informations collectées.

En résumé, même si les systèmes de stationnement intelligent présentent de nombreux bénéfices, il est crucial de prendre conscience et de surmonter les défis et les contraintes liés à leur mise en place afin d'assurer leur réussite à long terme. Il est essentiel d'adopter une approche globale, en étroite collaboration avec les autorités locales, les fournisseurs de technologie, les utilisateurs finaux et d'autres parties prenantes impliquées.

### II.11 Conclusion :

Dans ce chapitre, nous avons exposé une perspective globale sur les systèmes de stationnement intelligent qui présentent de nombreux bénéfices qui contribuent à améliorer la gestion du stationnement en milieu urbain, à diminuer la congestion et la pollution, et à améliorer l'expérience globale des conducteurs dans les zones urbaines. On a présenté aussi les techniques de développement d'un SI ainsi que les différents types de ces systèmes qui existent aujourd'hui. Finalement on a précisé quelque défis et limitations de ses systèmes.

## Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent

### III .1 Introduction :

Nous avons pour objectif de créer un système de stationnement intelligent tel qu'un système embarqué dans notre projet de fin d'études. Étant donné que notre système nécessite un système capable d'accueillir des capteurs de présence et une caméra en tant qu'entrées, ainsi que des signaux de commandes en sorties pour la manipulation des moteurs, nous avons opté pour le Raspberry Pi comme carte électronique. Par conséquent, nous avons opté pour Python comme logiciel de programmation en raison de son intégration dans Linux qui est le système d'exploitation utilisé sur Raspberry Pi.

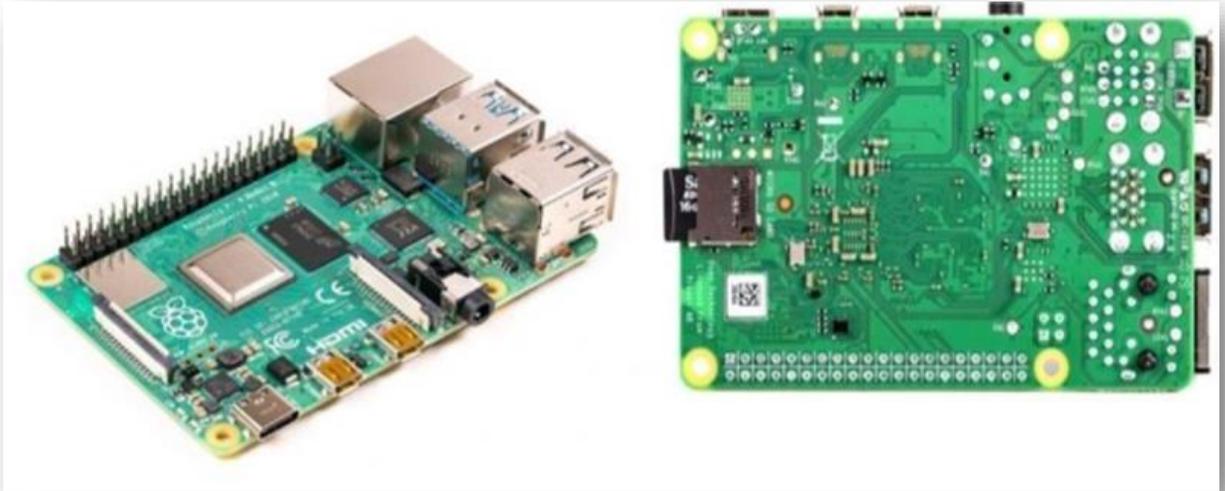
### III.2 Partie matériel

La partie matérielle qui constitue un système de stationnement intelligent est généralement composée d'une commande (microcontrôleur, Arduino, Raspberry...etc), des capteurs (camera, capteur de proximité,...) et des actionneurs (moteurs, servomoteur, Led...).

#### III.2.1 Raspberry Pi

Le Raspberry Pi est un mini-ordinateur mono-carte, souvent utilisé avec le système d'exploitation linux mais il peut supporter d'autres systèmes d'exploitation comme Windows par exemple. Il est destiné à des applications d'informatique embarquée. Le cœur de l'ordinateur est un FPGA (Broadcom2836) intégrant un processeur quad-core ARM11 CortexA7 cadencé à 700 MHz, de 256 Mo jusqu'à 4 Go de RAM selon la version et de nombreux périphériques.

Le Raspberry Pi peut être directement connecté à une IHM classique, souris/clavier/écran HDMI ou vidéo composite ou connecteur d'extension qui supporte les entrées/sorties parallèles ainsi que la plupart des bus de communication comme le bus CAN, Ethernet, I2C.... C'est un support particulièrement économique et puissant qui peut être facilement mis en œuvre dans des systèmes nécessitant un accès au monde physique par des capteurs/actionneurs disposant d'interfaces numériques [19]. La figure 16 présente un modèle de Raspberry PI 4 model B.



*Figure 16 : Raspberry PI 4 model B*

### A - Historique

Le Raspberry PI, ce petit ordinateur, était créé par des étudiants qui voulaient faire l'ordinateur le moins cher du monde afin d'encourager l'apprentissage de la programmation.

En 2006, Eben Upton étudiant de la faculté des sciences informatiques de l'université de Cambridge en Angleterre et ses collègues ont réalisés le premier prototype sur l'Atmel AtMega 664. Ce premier prototype était de taille d'une clé USB, et contient qu'un port USB et HDMI comme périphériques d'entrées/sorties. Les premières cartes sont produites en Mars 2012 à Taiwan et en Chine. Ils étaient équipés d'une RAM de 256Mo et un port RJ45. Puis la production est passée au Royaume-Uni en 2013-2014.

Aujourd'hui la fondation Raspberry PI annonce plus de 6 millions de Raspberry PI qui ont été vendus dans le monde entier [19] [20].

### B - Evolution

Le développement de l'ordinateur Raspberry Pi est un processus continu. Jusqu'à nos jours, la fondation Raspberry Pi a publié 12 modèles sur 4 générations d'ordinateur. Les modèles sont des versions mises à niveau de leurs versions précédentes pour rendre l'ordinateur plus efficace et pratique pour les utilisateurs, notamment en raison de sa faible consommation énergétique et de ses périphériques d'entrées/sorties. Les différences entre ces modèles sont le CPU utilisé, la taille de la RAM, la taille du GPIO, le nombre de ports USB, les périphériques réseau (Ethernet, WIFI, Bluetooth) et les périphériques de multimédia [19].

La figure suivante présente les différentes générations du Raspberry Pi :

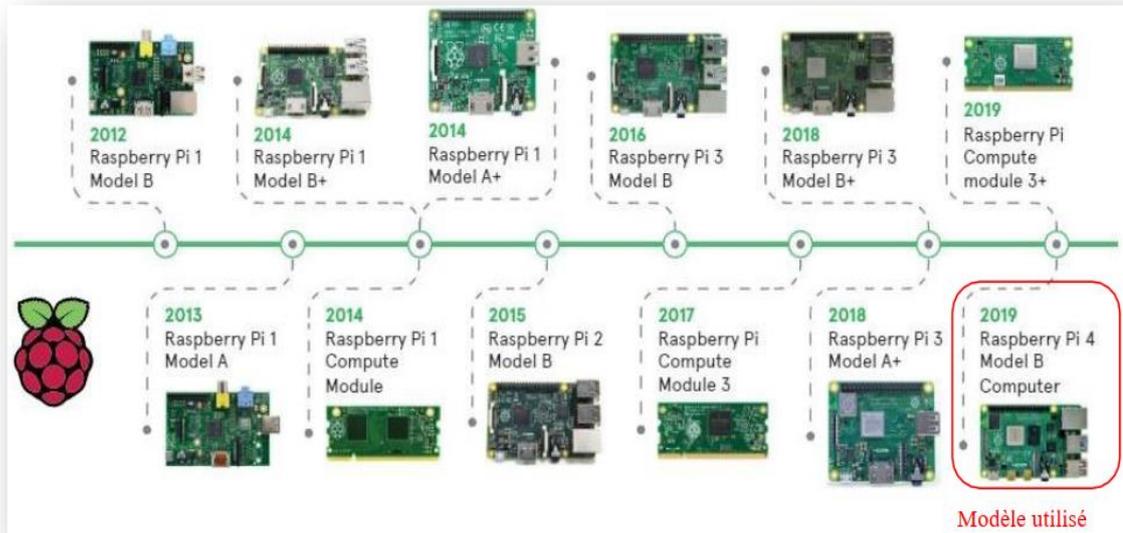


Figure 17 : Evolution du Raspberry PI

### C- Composants de base

Le Raspberry Pi 4 model B est l'avant dernier produit de la gamme d'ordinateurs Raspberry Pi. Il offre des augmentations révolutionnaires de la vitesse du processeur, des performances multimédias, de la mémoire et de la connectivité par rapport au Raspberry Pi 3 model B+ de la génération précédente, tout en conservant une compatibilité descendante et une consommation d'énergie similaire. Pour l'utilisateur final, le Raspberry Pi 4 modèle B offre des performances de bureau comparables aux systèmes PC x86 d'entrée de gamme. C'est pour cela que nous l'avion choisis [19].

Les principales caractéristiques de ce produit sont : [19] [21]

- Un processeur 64 bits quad-core ARM Cortex A72 cadencé à 1.5GHz intégré dans FPGA Broadcom 2711.
- La prise en charge du double affichage à des résolutions allant jusqu'à 4K via une paire de ports micro-HDMI.
- Le décodage vidéo matériel jusqu'à 4Kp60
- Une Capacité de RAM jusqu'à 8 Go
- Double bande LAN sans fil 2.4/5.0 GHz + Bluetooth 5.0
- Gigabit Ethernet.
- Capacité PoE (via un module complémentaire PoE HAT distinct).

La figure 18 ci-dessous résume tous les composants de base du Raspberry Pi 4

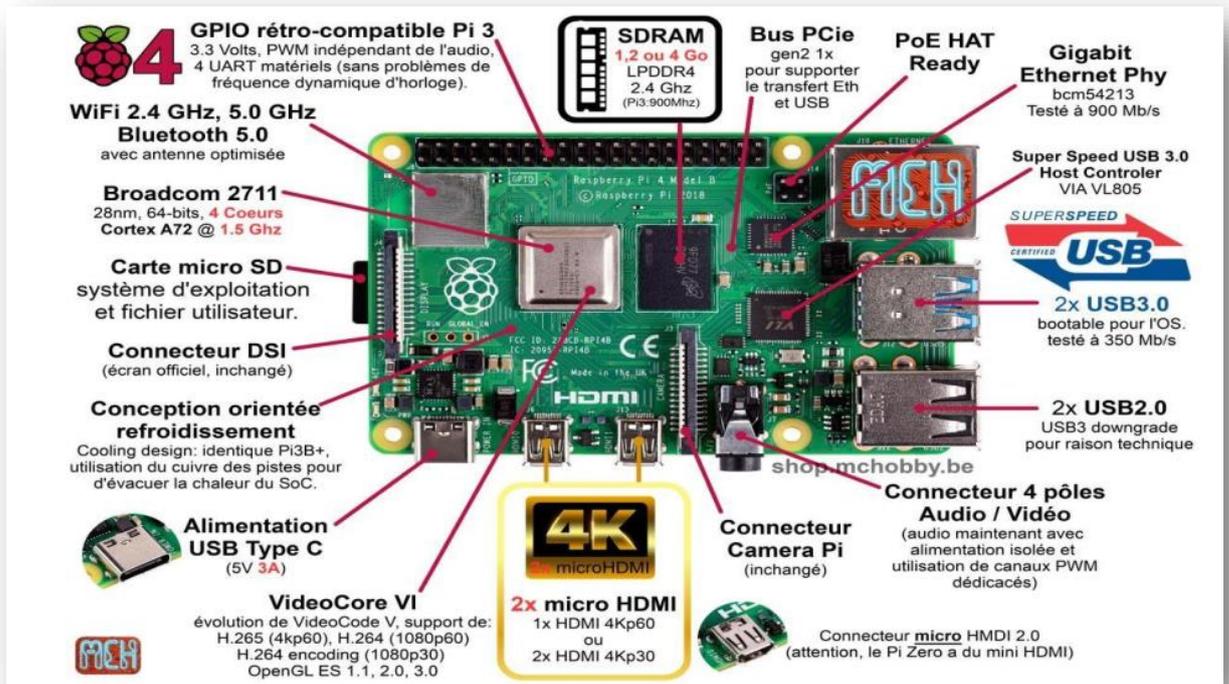


Figure 18 : Composants de base du Raspberry Pi 4

### III.2.2 Servomoteur SG90

Un servomoteur est un type de moteur alimenté par une source CC, soit à partir d'une alimentation externe, soit par un contrôleur. Un servomoteur petit et léger avec une puissance de sortie élevée est appelé micro servomoteur SG90. Cela signifie que le micro servomoteur SG90 ne fonctionnera que dans la mesure nécessaire pour accomplir la tâche à accomplir. Il existe une large gamme d'applications pour les servomoteurs SG90, notamment les caméras, les télescopes, les antennes, l'automatisation industrielle et les robots [22].

La Figure 19 montre un exemple d'un servomoteur SG90 qu'on a utilisé dans ce projet.



Figure 19 : un servomoteur SG90

## Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent

Les parties stationnaires et rotatives d'un micro-servomoteur sont enroulées avec des enroulements de stator et de rotor. Outre sa consommation réduite d'électricité, ce moteur est extrêmement performant et il est relativement simple à entretenir. Les servomoteurs sont pilotés par des amplificateurs pilotes, qui reçoivent les commandes de vitesse des contrôleurs (voir figure 20).

Les servomoteurs fonctionnent avec des convertisseurs PWM et requièrent un contrôleur pour contrôler l'impulsion, qui peut être géré par des microcontrôleurs comme l'Arduino, Raspberry ou d'autres contrôleurs programmables.

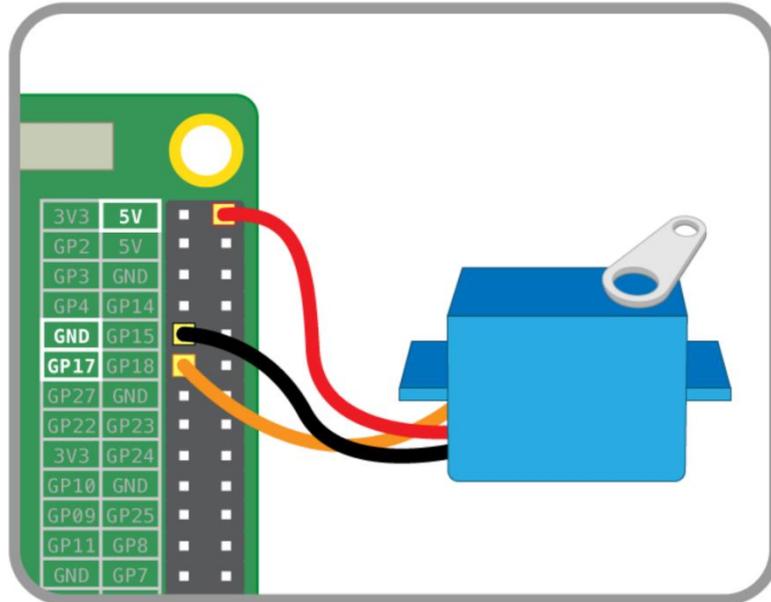


Figure 20 : un servomoteur contrôlé par un raspberry

### III.2.3 Caméra Raspberry Pi 2

Le module caméra v2 de type CMOS est doté d'un capteur Sony IMX219 de 8 mégapixels (par rapport au capteur OmniVision OV5647 de 5 mégapixels de la caméra d'origine). La taille de pixel est 1.12 um x 1.12 um, celle du capteur est  $\frac{1}{4}$ , la résolution maximale de l'image est 3280 x 2464 pixels et la Capacité vidéo supporte jusqu'à 1080p30, 720p60, et VGA90. Le module est équipé d'une lentille fixe. Contrairement aux caméras avec mise au point ajustable, la lentille fixe est calibrée pour fournir une mise au point nette à une certaine distance, ce qui simplifie l'utilisation pour la plupart des applications générales. Le capteur et les composants associés sont montés sur une petite carte de circuit imprimé. Cette carte contient un connecteur FFC pour la nappe plate (Flat Flexible Cable) qui relie la caméra au Raspberry Pi, des composants électroniques pour gérer l'alimentation et l'interface entre le capteur et le Raspberry Pi. Le Microcontrôleur gère la communication entre le capteur d'image et le Raspberry Pi, contrôlant des aspects comme l'exposition et la balance des blancs. Le

## Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent

module caméra v2 se connecte au Raspberry Pi via le port CSI (Camera Serial Interface) à l'aide d'une nappe plate [23] (voir figure 21).



Figure 21 : Caméra Raspberry Pi 2

### III.2.4 Capteur de proximité FC-51

Le capteur de proximité FC-51 est composé d'une diode infrarouge (émetteur) et d'une photodiode (récepteur). Suivant la distance à laquelle se trouvent l'obstacle, le récepteur recevra plus ou moins de lumière infra-rouge réfléchi (voir figure 22).

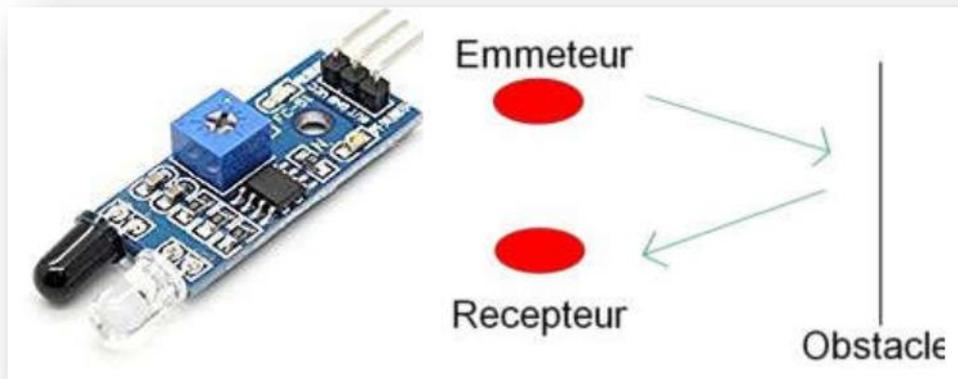


Figure 22 : Capteur de proximité FC-51

Les caractéristiques principales de ce capteur : [24]

- Distance de détection : une plage de distance allant de 2 à 30 cm
- Alimentation : une alimentation électrique comprise entre 3,3V et 5 V
- Interface de sortie(OUT) : Lorsqu'un obstacle est détecté, la sortie sera à l'état bas, c'est-à-dire à 0.
- LED de détection : Lorsque le capteur détecte un obstacle, une LED verte intégrée s'allume.
- Réglage de la distance de détection : doté d'un potentiomètre qui permet d'ajuster la distance de détection

### III.3 Partie logicielle

Dans notre système de stationnement intelligent on a utilisé Linux comme système d'exploitation pour le Raspberry et python pour la programmation et l'utilisation des outils de Machine Learning.

#### III.3.1 Linux :

Avec la montée du concept 'open-source' et des logiciels libres, le monde du calcul informatique a grandement évolué en puissance et applications, et plus particulièrement dans le domaine des systèmes embarqués.

Linux est un système d'exploitation Open source basés sur le noyau Unix (voir figure 23). C'est un système fiable, flexible et d'un niveau de sécurité très haut. Linux est intégré sous plusieurs distributions comme Ubuntu, Debian... Les distributions incluent le noyau Linux, les logiciels et les bibliothèques du système de support. Linux est le principal système d'exploitation sur les serveurs et les ordinateurs centraux, mais également sur les systèmes embarqués tel que les routeurs, les Smart TVs, les automobiles, les consoles de jeux et même les Smart phones qui utilisent Android, un dérivé de Linux [25].

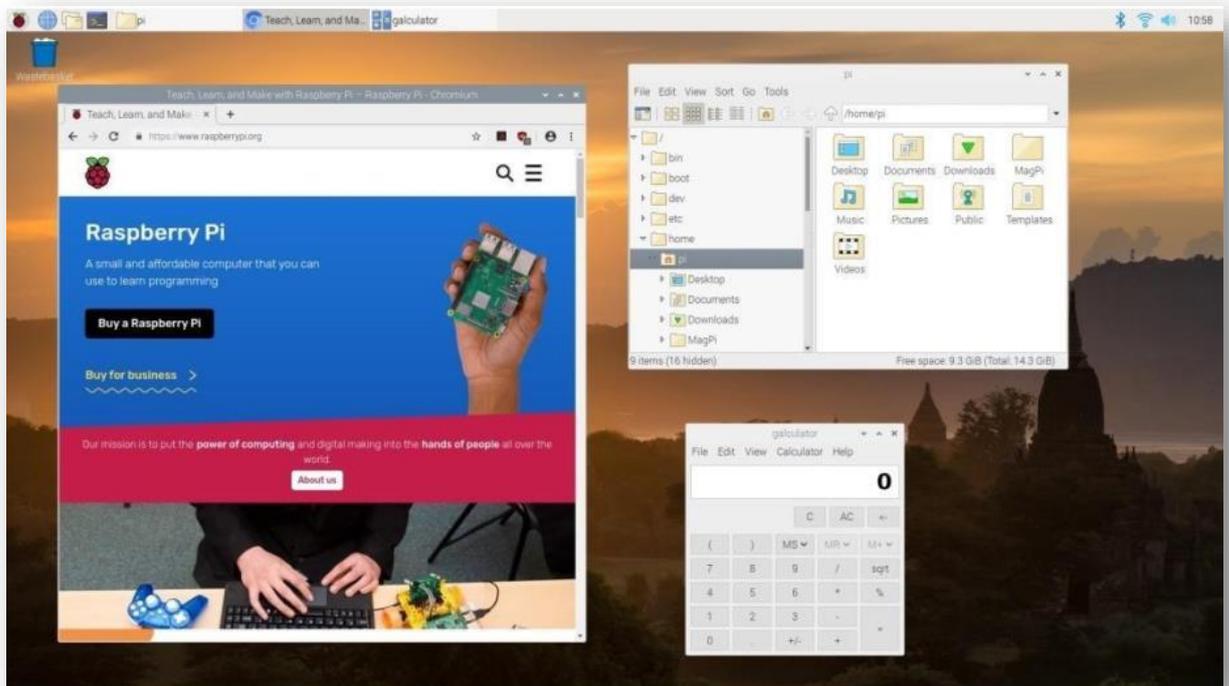


Figure 23 : Interface système d'exploitation Linux-Debian

#### III.3.2 Python :

Python est un langage de programmation merveilleux, puissant et facile à utiliser (facile à lire est à écrire). Il permet l'interaction de Raspberry Pi avec le monde réel. La syntaxe Python

---

### Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent

---

est très propre, avec un accent sur la lisibilité. Il utilise des mots clés en anglais standard comme : print, input, if...

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes (recyclage de mémoire) et d'un système de gestion d'exceptions.

Le langage python est placé sous une licence libre. Il fonctionne sur la plupart des plateformes informatiques, des Smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par MacOS, ou encore Android, IOS. Il peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser [26].



*Figure 24 : Logo Python*

A - OpenCV :

Open Source Computer Vision Library en anglais, est une librairie ou bibliothèque qui contient un ensemble de fonctions de programmation dédiées à la vision par ordinateur (traitement d'image ou vidéo) en temps réel et d'apprentissage automatique. Cette bibliothèque est développée par Intel en 1999 dans le but d'améliorer les applications intensives du CPU. OpenCV est constituée de plus de 2500 algorithmes. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, trouver des images similaires dans une base de données d'images ... [27].

La Figure 25 montre un exemple d'utilisation d'OpenCV pour la localisation et suivi des véhicules.

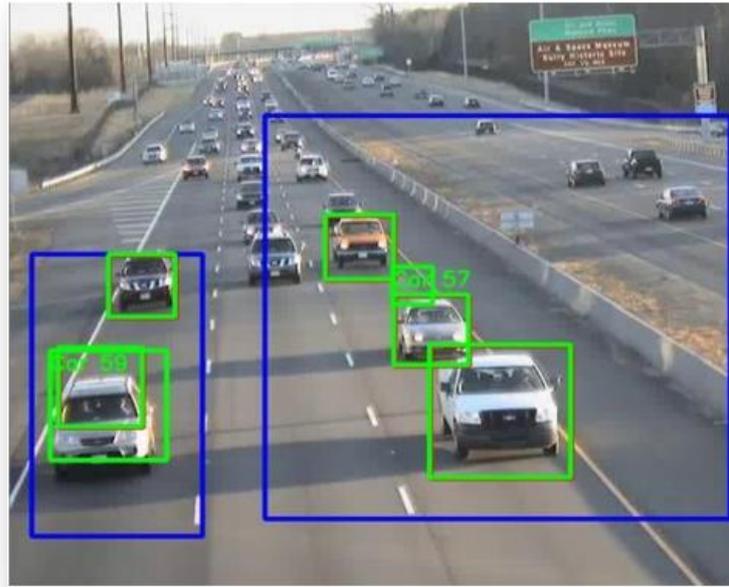


Figure 25 : Localisation et suivi des véhicules avec OpenCV

La bibliothèque OpenCV offre une implémentation efficace de l'algorithme de Canny. Dans notre système, nous avons employé la détection des contours de Canny afin de segmenter les images de parking en plusieurs places de stationnement.

### B - scikit-learn

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria. Elle propose dans son Framework de nombreuses bibliothèques d'algorithmes à implémenter, clé en main. Ces bibliothèques sont à disposition notamment des data scientists.

Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy, SciPy et matplotlib.

Scikit-learn propose une mise en œuvre intuitive et intégrée des SVM, ce qui permet à notre système de former et d'évaluer rapidement des modèles SVM sur des jeux de données pour classer les emplacements de stationnement [28].

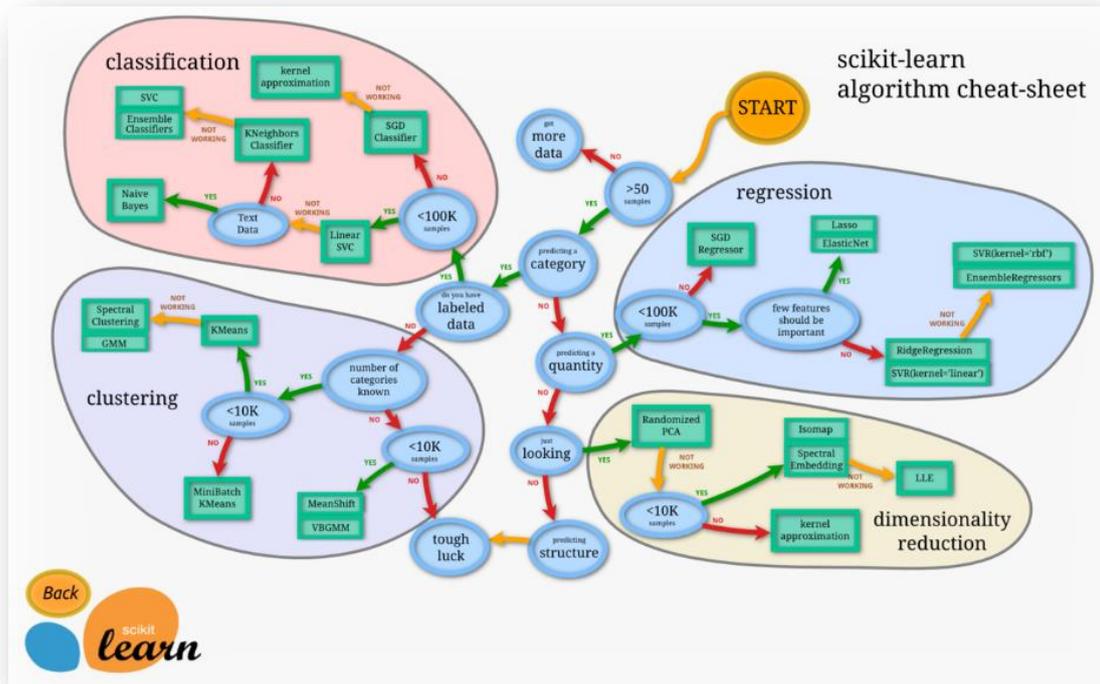


Figure 26 : Les différentes caractéristiques de scikit-learn

### III.3.3 L'intelligence artificielle (IA)

Le domaine de l'intelligence artificielle (IA) se focalise sur la conception de machines capables de réaliser des tâches qui requièrent habituellement l'intelligence humaine. Parmi ces missions figurent la détection de raisons, la résolution de problèmes, la prise de décisions, l'acquisition et la compréhension du langage ordinaire. L'IA peut être appliquée de diverses façons, y compris en utilisant des méthodes comme l'apprentissage automatique, les réseaux de neurones artificiels et de machine learning. Elle est de plus en plus utilisée dans de multiples secteurs, de la médecine à la finance en passant par l'industrie, où elle présente des avantages considérables en termes d'efficacité et de précision. Dans notre cas l'intelligence artificielle nous sert à automatiser notre système de stationnement et le rendre intelligent [29].

#### A - Mise en œuvre de l'IA

L'IA peut être utilisée de diverses façons en fonction des besoins et des applications particulières. Voici certaines des principales approches pour mettre en place l'IA : [29]

- Apprentissage automatique (Machine Learning) : Une approche où les algorithmes sont entraînés à partir de données pour reconnaître des modèles et prendre des décisions. Cela englobe différentes méthodes telles que les réseaux de neurones artificiels, les arbres de décision, les machines à vecteurs de support, etc.

---

## Chapitre 3 : Les Composants du Systèmes de Stationnement Intelligent

---

- Réseaux de neurones artificiels (ANN) : Inspirés par le fonctionnement du cerveau humain, ces réseaux de neurones utilisent des couches de nœuds interconnectés pour traiter des données et apprendre des modèles.
- Vision par ordinateur : Une discipline de l'IA qui permet aux machines de comprendre et d'interpréter visuellement le monde. Cela inclut la reconnaissance faciale, la détection d'objets, la segmentation d'images, etc.

Dans la section suivante, nous allons présenter l'apprentissage automatique (Machine Learning), un outil essentiel utilisé dans ce projet.

### B - L'apprentissage automatique (Machine learning)

L'apprentissage automatique est une branche de l'intelligence artificielle qui se focalise sur le développement d'algorithmes et de méthodes qui permettent aux ordinateurs de "apprendre" à partir de données et de faire des prédictions et/ou des décisions sans être explicitement programmés pour chaque tâche [30].

### C - Principaux Types d'Apprentissage

- **Apprentissage supervisé** : Chaque exemple d'entraînement est lié à une étiquette ou à une valeur cible, ce qui permet de former l'algorithme sur un ensemble de données étiquetées.
- **Apprentissage non supervisé** : L'algorithme essaie de trouver des structures ou des motifs dans des données non étiquetées.
- **Apprentissage par renforcement** : Un agent apprend à prendre des décisions en interagissant avec un environnement et en recevant des récompenses ou des punitions pour ses actions [30].

### D - Étapes de la Mise en Œuvre du Machine Learning

- **Collecte de données** : Collecter un corpus de données qui reflète le problème que l'on souhaite résoudre. Cela englobe des informations structurées (bases de données, tableurs) ainsi que des informations non structurées (images, texte).
- **Préparation des données** : En premier lieu, il est nécessaire de nettoyer les données, puis de les transformer (normaliser, standardiser, encoder...) et de les diviser en ensembles d'entraînement et de test.
- **Sélection et entraînement du modèle** : Le Choix d'un modèle ou d'une combinaison de modèles en fonction du problème à résoudre, puis de l'entraîner sur l'ensemble des données d'entraînement.

- **Évaluation du modèle** : Évaluation des performances du modèle sur l'ensemble de test à l'aide de métriques appropriées (précision, rappel, F1-score, RMSE, etc.).
- **Optimisation et réglage des hyper paramètres** : Ajustement des paramètres du modèle afin d'optimiser ses performances (Recherche en grille, Recherche à l'échelle, Optimisation bayésienne).
- **Déploiement** : Intégration du modèle dans un système de production où il peut être utilisé pour faire des prédictions sur de nouvelles données, tout en assurant la surveillance et la maintenance du modèle afin de garantir sa performance optimale [30].

### III.4 Conclusion :

Dans ce chapitre nous avons présenté le Raspberry PI, et les bibliothèques openCv (traitement d'image) et Scikit Learn (Machine Learning) qu'utilise le langage de programmation Python, en donnant quelques détails techniques, qui vont nous permettre de les intégrer notre système.

## Chapitre 4 : Implémentation du Système SI

### IV.1 Introduction :

Dans ce chapitre, nous allons présenter la mise en œuvre du système développé SI : l'emplacement physique du système, l'environnement matériel (Hardware) et l'environnement logiciel (Software). Puis nous allons faire une analyse des résultats obtenus lors de sa mise en marche.

### IV.2 Description de notre système :

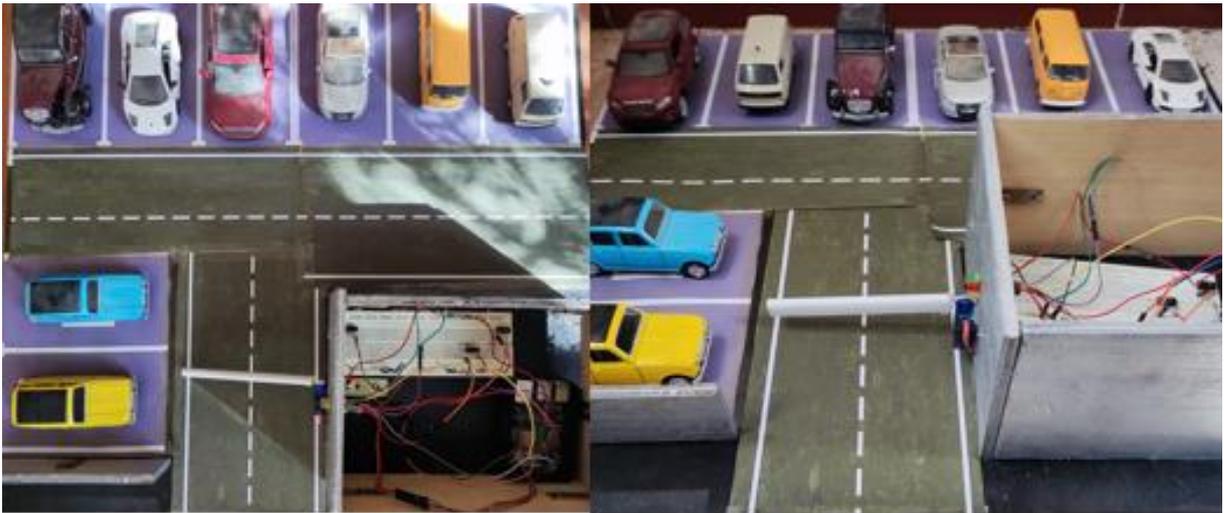


Figure 27 : maquette de présentation de notre système de stationnement intelligent

### IV.3 Le matériel : Raspberry et ses périphériques :

Le système SI développé est constitué de :

- Un micro-ordinateur Raspberry Pi 4 Model B (voir chapitre2).
- Webcam USB d'une résolution de (640x480) avec un capteur de taille. (4.86x3.64mm<sup>2</sup>), est utilisée pour acquérir des images de parking.
- 2 boutons poussoir utilisés comme capteurs de présence.
- 1 servomoteur avec une vitesse de rotation de 60 degrés/0.1sec et de plage de tension de fonctionnement de 4.8V-6V. son rôle est d'entraîner une barrière.
- Un système de feux colorés (trois LED) :

#### Signification des feux colorés :

<b>Orange</b>	Acquisition d'image par caméra(s)
<b>Vert</b>	Accès autorisé
<b>Rouge</b>	Accès non autorisé

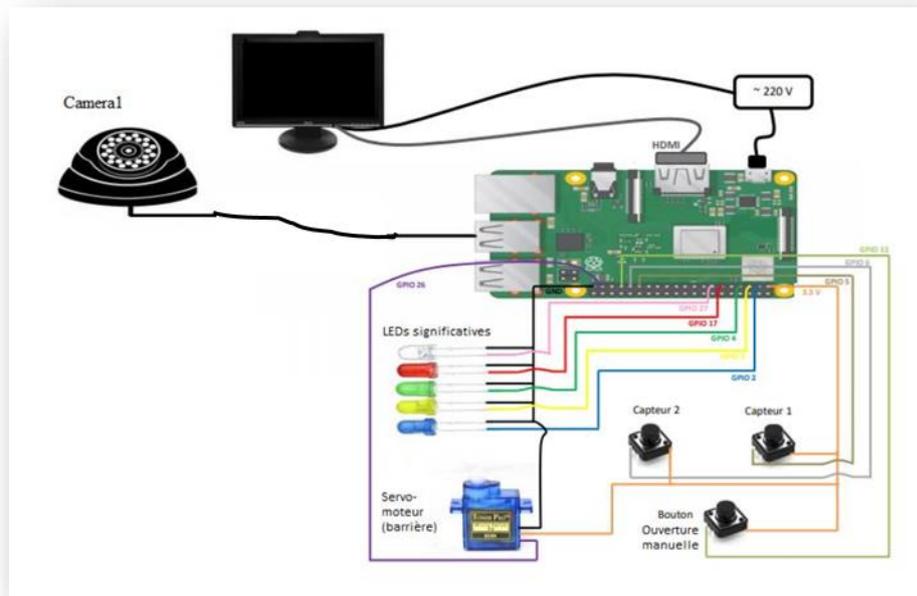


Figure 28 : Schéma électronique du système SI

### IV.4 L'environnement logiciel (Software) :

La gestion des périphériques, leurs états et les actions sur ces derniers, et des différentes situations (entrées/sorties de véhicule) sont réalisées sous forme de programmes implémentés sur Raspberry, écrits en langage Python sous Linux et font appel à la bibliothèque OpenCV et la bibliothèque scikit-learn .

#### IV .4.1 Logique de gestion des scénarios d'utilisation :

Lorsque le système est en marche trois cas d'évènements sont considérés :

1. Présence d'un véhicule à l'entrée (procédure d'entrée) ;
2. Présence d'un véhicule à la sortie (procédure de sortie) ;
3. Fonctionnement manuel ;

#### A - Procédure d'entrée :

Le système entame cette procédure si détection d'un véhicule à l'entrées ( $C1=1$ ,  $C2=0$ ) ; alors → la LED orange s'allumera indiquant que la caméra prend une image du parking. L'image est ensuite transmise à l'unité de traitement (Raspberry). Puis, l'image acquise est traitée à l'aide d'un système global. Deux situations peuvent se présenter :

## Chapitre 4 : Implémentation du Système SI

- Place disponible : Le véhicule est autorisé pour l'accès au parking (allumage de la LED verte dans ce cas). Après l'accès du véhicule (fermeture du C2) la barrière se referme.
- Place occupée : Le véhicule n'est pas autorisé (une LED rouge s'allume).

B - Procédure de sortie :

Dans le cas d'une voiture sortante (C1=0, C2=1) la barrière s'ouvre pour laisser passer le véhicule. Une fois le capteur C1 est actionné la barrière se referme.

C - Fonctionnement manuel :

Cette procédure est abordée lorsqu'un véhicule d'urgence ou prioritaire veut rentrer ; dans ce cas un bouton poussoir est actionné pour la gestion manuelle du parking.

Avec C1 : capteur de présence à l'entrée.

C2 : capteur de présence à la sortie.

La figure ci-dessous présente le diagramme de fonctionnement logique de notre système :

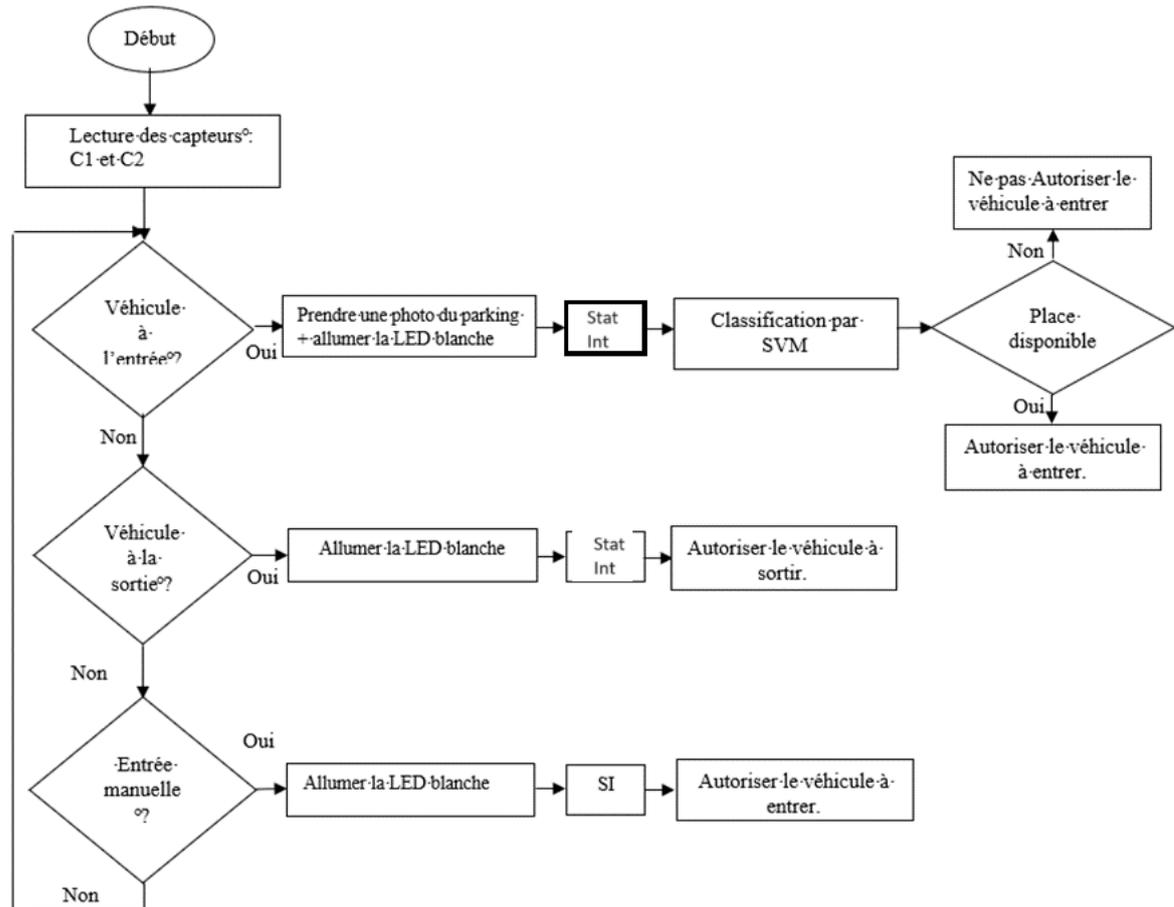


Figure 29 : Organigramme de fonctionnement du système SI

### IV.4.2 Phases de Développement

L'objectif de ce travail est la détection de la disponibilité d'une place de stationnement dans un parking. Cela est fait en deux parties l'une dépendant de l'autre. La première partie, connue sous la Création et entraînement du modèle de classification, et La deuxième partie, appelée la Détection en Temps Réel

Notre proposition de travail englobe les deux étapes qui sont résumées dans la figure suivante

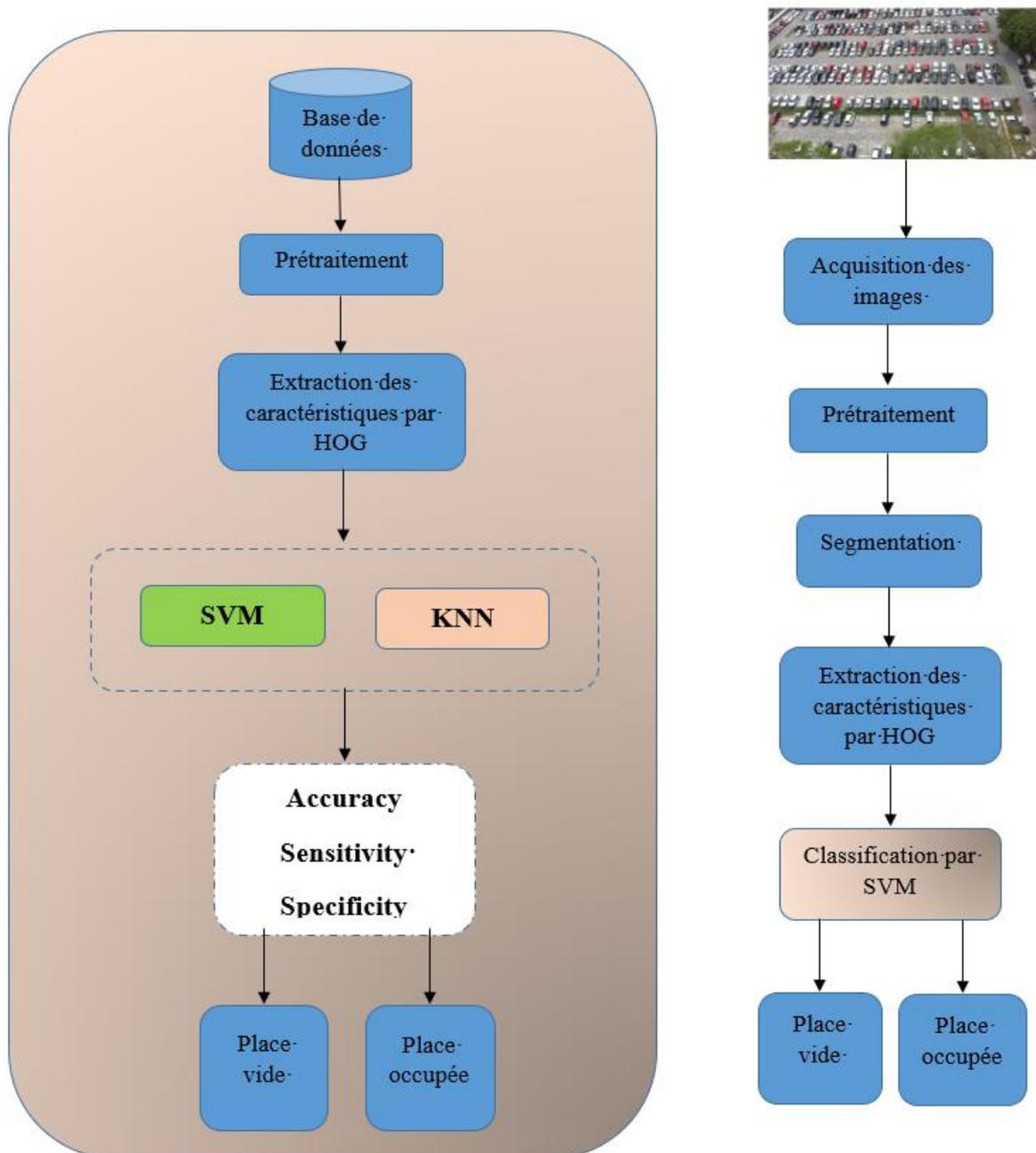


Figure 30 : Les différents étapes de développement de notre système SI

#### IV.4.2.1 La Création et entraînement du modèle de classification :

Elle consiste à créer une base de données contenant des images des places de stationnement avec des annotations binaires (1 pour les places libres, 0 pour les places occupées). Ensuite,

---

## Chapitre 4 : Implémentation du Système SI

---

nous extrairons les caractéristiques des images annotées pour obtenir des vecteurs de caractéristiques, ce qui nécessite plusieurs méthodes et techniques qu'on que nous étudierons par la suite. Par la suite, en utilisant les vecteurs de caractéristiques extraites pour entraîner un modèle de Machine Learning que nous examinerons plus loin afin de classer les images de stationnement en deux catégories : libre ou occupée.

### *A - La base des données :*

La collection de données comprend 1500 images de places de stationnement libres et 1500 images de places de stationnement occupées. Ces images présentent diverses conditions d'éclairage, telles que le jour, la nuit et les périodes de transition. Les illustrations utilisées dans cette étude présentent une variété de résolutions, allant de (28 \* 48) pixels à (93 \* 154) pixels, ce qui permet d'obtenir différents niveaux de détails visuels pour l'analyse.



*Figure 31 : exemple d'une image dans la base de données*

### **• Base d'apprentissage :**

Cette base renferme 80% de la totalité de la base de données. Cela représente 1200 places de stationnement disponibles et 1200 places de stationnement occupées.

### **• Base de test :**

Cette base dispose de 300 emplacements disponibles et 300 emplacements occupés. En représentant 20% de l'ensemble de la base de données

### *B- Prétraitement et redimensionnement :*

Notre prétraitement Consiste à filtrer le bruit pour améliorer la qualité des images traitées, pour cela on a utilisé le filtre médian. Pour le redimensionnement nous avons simplement effectué un redimensionnement des images en 16x24 pixels, pour faciliter l'extraction des caractéristiques.

### *C - Extraction des caractéristiques :*

Nous avons opté pour l'utilisation de l'algorithme HOG pour extraire les attributs et créer des descripteurs. Après avoir lu les images de la base une par une, et le traiter. On extrait un vecteur descripteur contenant tous les attributs extraits à toutes les résolutions. Ce vecteur descripteur est stocké dans la base de données.

### *D - Histogramme des gradients orientés HOG :*

Histogramme des gradients orientés (HOG) est utilisé pour extraire les caractéristiques globales, à cette méthode, le but est de reconnaître l'image comme le groupe d'histogrammes locaux.



*Figure 32 : Extraction des caractéristiques par le HOG (cellule=4x4) pour une place occupée*



*Figure 33 : Extraction des caractéristiques par le HOG (cellule=4x4) pour une place vide*

La première étape consiste à calculer l'histogramme d'orientation du dégradé autour de la région de 16 x 24 pixels de chaque point d'intérêt. Premièrement, la région est divisée en sous-région de 4 X 4. Pour chaque sous-région, on calcule l'orientation du gradient de 64 cases  $C(N)$ ,  $N = 1$  à 8 qui forme un vecteur caractéristique de dimension 128 ( $4 \times 4 \times 8$ ). Chaque pixel d'une cellule vote pour une orientation entre 0 et 180 dans le cas non signé ou entre 0 et 360 dans le cas signé [31].

Enfin, combiner ensemble l'histogramme d'orientation Gradient de la zone d'intérêt pour former un vecteur de caractéristiques de taille 128 dimension. Le vecteur de HOG pour chaque cellule est normalisé en utilisant des vecteurs dans leurs blocs respectifs.

### *E - Classification :*

Nous avons opté pour l'utilisation de deux algorithmes séparés (SVM et KNN) pour la classification. Cette étape prend comme entrée le vecteur de caractéristique de chaque image obtenue de l'étape précédent et l'étiquette de la classe (vide ou occupée) qui appartient ce vecteur et parcourt la base d'image pour obtenir la base d'apprentissage.

### *F - Séparateurs à vaste marge (SVM) :*

Séparateurs à vaste marge est un algorithme consiste à séparer par un hyperplan des individus représentés dans un espace de dimension égale au nombre de caractéristiques, les individus étant alors séparés en deux classes. Cela est possible quand les données à classer sont linéairement séparables. Dans le cas contraire, les données seront projetées sur un espace de plus grande dimension afin qu'elles deviennent linéairement séparables [32].

Dans notre système mis au point, les images sont séparées de manière linéaire et classées de manière binaire (place de stationnement vide/occupée). On considère un ensemble de  $m$  vecteurs de données auquel on associe des étiquettes  $t_i \in \{-1, +1\}$  représentatives de leurs classes

La classification en utilisant un SVM implique une première étape de Créez des ensembles d'entraînement et de test pour les données, puis entraînez le modèle SVM sur les vecteurs de caractéristiques extraits. Ensuite, nous utilisons le modèle SVM pour prédire l'occupation des places de stationnement en se basant sur les données de test.

### *G - Nearest Neighbors (KNN)*

Notre programme effectue une tâche de classification en utilisant l'algorithme des  $k$  plus proches voisins (KNN). Il commence par extraire les caractéristiques HOG. Ensuite, les données sont préparées pour l'apprentissage en combinant les caractéristiques extraites dans une matrice globale. Les caractéristiques sont séparées des étiquettes correspondantes. Ensuite, les données sont divisées en ensembles d'entraînement et de test.

### *H- Métrique*

Dans le cadre de cette étude, nous avons évalué la performance de classification en utilisant trois métriques pour une mesure quantitative : la précision de classification (Accuracy), la spécificité et la sensibilité. Ces métriques sont détaillées ci-dessous : [33]

### *I - Précision de classification (Accuracy) :*

Elle est effectivement une mesure couramment utilisée pour évaluer la performance d'un modèle de classification. Elle exprime le taux global de bonnes classifications et est définie comme la proportion d'images correctement classifiées par rapport au nombre total d'images

$$\text{(Accuracy)} = \frac{\text{nombre d'image correctement classifiées}}{\text{nombre total d'image}} \times 100$$

J -Spécificité :

Elle est une métrique utilisée pour évaluer la performance d'un classifieur dans le contexte de la classification binaire. Elle mesure le taux d'exemples négatifs TN correctement prédits par le classifieur.

$$\text{Spécificité} = \frac{TN}{TN+FP} \times 100$$

K - Sensitivité :

C'est une métrique qui mesure le taux d'exemples positifs correctement prédits par un classifieur.

$$\text{Sensitivité} = \frac{TP}{TP+FN} \times 100$$

Ces métriques sont complémentaires qui résument la performance d'un classifieur en prenant en compte l'ensemble des composantes de la matrice de confusion. Cette dernière est composée de quatre valeurs (Tableau 3.1) dans le cas d'un problème de classification à deux classes (classe positive vs classe négative).

*Tableau 1 : matrice de Confusion*

		Valeur prédit	
		Place vide (- 1)	Place occupée (1)
Valeur réel	Place vide (-1)	TP – True Positive	FP – False Positive
	Place occupée (1)	FN – False Negative	TN – TrueNegative

- TruePositive (TP): place vide et trouver par le classifieur vide
- False Positive (FP):place vide et trouver par le classifieur occupée
- TrueNegative (TN):place occupée et trouver par le classifieur occupée
- False Negative (FN):place occupée et trouver par le classifieur vide

Exemple :

## Chapitre 4 : Implémentation du Système SI

Tableau 2 : exemple d'une matrice de Confusion

		Valeur prédit	
		Place vide (- 1)	Place occupée (1)
Valeur réel	Place vide (-1)	165	12
	Place occupée (1)	23	200

$$(\text{Accuracy}) = \frac{165+200}{200+12} \times 100 = 91,25\%$$

$$\text{Spécificité} = \frac{200}{200+12} \times 100 = 94,34\%$$

$$\text{Sensitivité} = \frac{165}{165+23} \times 100 = 87,78\%$$

### IV.4.2.2 La Détection en Temps Réel :

Implique d'obtenir une image en temps réel lorsqu' un véhicule traverse un capteur à l'entrée du parking. Ensuite, cette image sera prétraitée pour améliorer leur qualité. Ensuite, elle sera segmentée en zones d'intérêt en fonction du nombre maximal de places dans le parking (chaque place sera représentée par une image segmentée). Ensuite, les caractéristiques de chaque image seront extraites pour être classées par le modèle créé dans la première étape afin d'obtenir soit des places disponibles, soit des places occupées.

Dans cette partie, nous avons suivi le même processus de prétraitement et d'extraction des caractéristiques que dans la première partie.

#### A - Acquisition de la donnée :

Les images utilisées pour la détection de la disponibilité d'une place de stationnement sont acquises à l'aide d'une caméra installée au niveau du parking. Elle commence à acquérir lorsque la détection d'un véhicule à l'entrée grâce à un capteur positionne juste avant la barrière.

#### B- Segmentation :

La segmentation d'image implique la division d'une image en différentes parties ou zones d'intérêt distinctes. La segmentation est employée dans le cadre d'un système de stationnement intelligent afin de séparer les différentes places de stationnement dans l'image capturée par la caméra. Chaque espace segmenté est associé à une place de stationnement spécifique, ce qui facilite l'analyse et la classification ultérieure de l'occupation des places.

## Chapitre 4 : Implémentation du Système SI

Au sein de notre projet de développement, nous avons utilisé la segmentation en détectant les contours en utilisant le filtre de Canny. Après avoir réduit le bruit en employant le flou gaussien et les caractéristiques morphologiques.

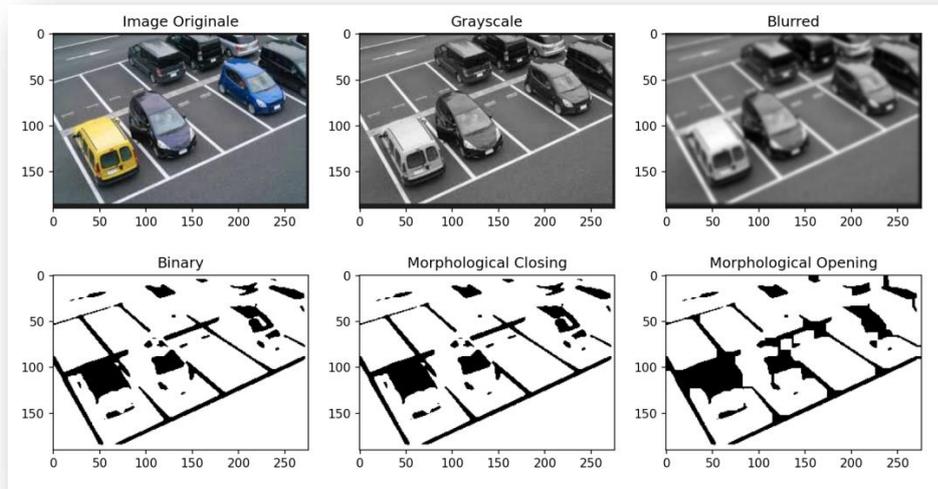


Figure 34 : segmentation par la détection de contour

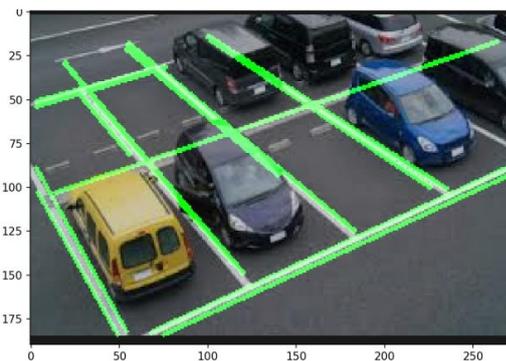


Figure 35 : image segmentée

### IV.5 Résultats et discussion :

Les résultats obtenus sont scindés en deux parties distinctes. La première partie comprend les résultats du modèle de classification, tandis que la deuxième partie comprend les résultats de détection en temps réel.

#### IV.5.1 Les résultats du modèle de classification

Les résultats du modèle de classification sont présentés dans le tableau 3. Le meilleur taux de précision de 98,66 % a été obtenu en utilisant SVM en combinaison avec l'extracteur HOG. En outre, d'autres paramètres sont calculés, notamment la sensibilité (pour évaluer la précision de la classification des places libres) et la spécificité (pour évaluer la précision de la classification des places occupées), qui s'élèvent respectivement à 95,89 % et 100 %. Il convient

## Chapitre 4 : Implémentation du Système SI

également de mentionner que le classifieur KNN a également obtenu des résultats comparables à ceux du SVM

Tableau 3 : Résultats de la classification avec SVM et KNN

Base de données	extracteur	Taille de cellule	Nombre de caractéristique	Classification	Evaluation		
					Accuracy (%)	Sensitivité (%)	Spécificité (%)
Images segmenté	HOG	8*8	32	SVM	97.66	98.63	97.27
				KNN	95.83	100	92.74
		4*4	240	SVM	98.66	95.98	100
				KNN	97.75	100	94.98

La figure 36 ci-dessous montre le résumé de la précision de classification avec SVM et KNN en utilisant l'extracteur HOG :

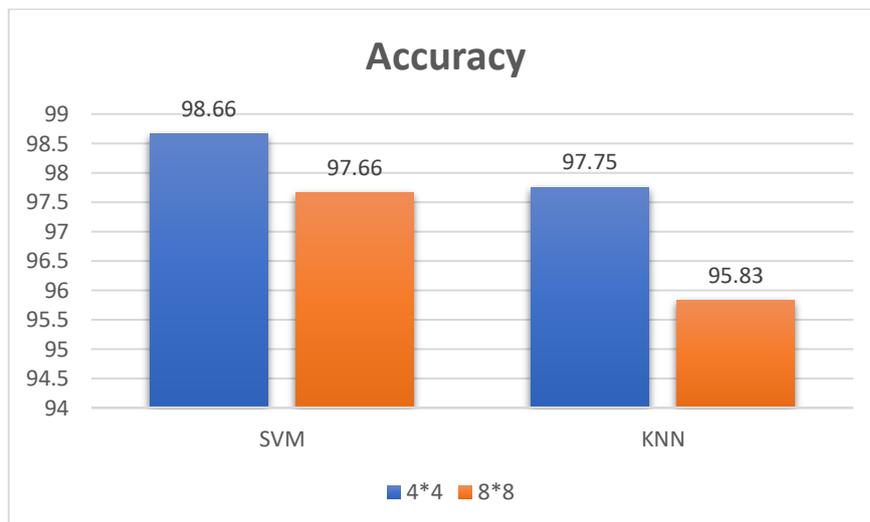


Figure 36 : La précision de classification avec SVM et KNN

En se basant sur ces résultats, il est possible de formuler les observations suivantes : Des résultats satisfaisants ont été obtenus lors de l'extraction des caractéristiques HOG (4,4), en raison de ses scores de précision de classification les plus élevés, elle a obtenu respectivement 98,66% et 97,75% pour SVM et KNN.

Les performances de la méthode d'extraction des caractéristiques HOG (8,8) sont également satisfaisantes, même si elle est légèrement inférieure à HOG (4,4), avec des précisions de classification de 97,66% et 95,83% respectivement pour SVM et KNN.

Globalement, l'extraction de caractéristiques HOG (4,4) associée à SVM semble fournir les meilleurs résultats en termes de précision.

La figure 37 ci-dessous récapitule les résultats de classification des places vides (sensibilité) et des places occupées (spécificité) à l'aide du classifieur SVM.

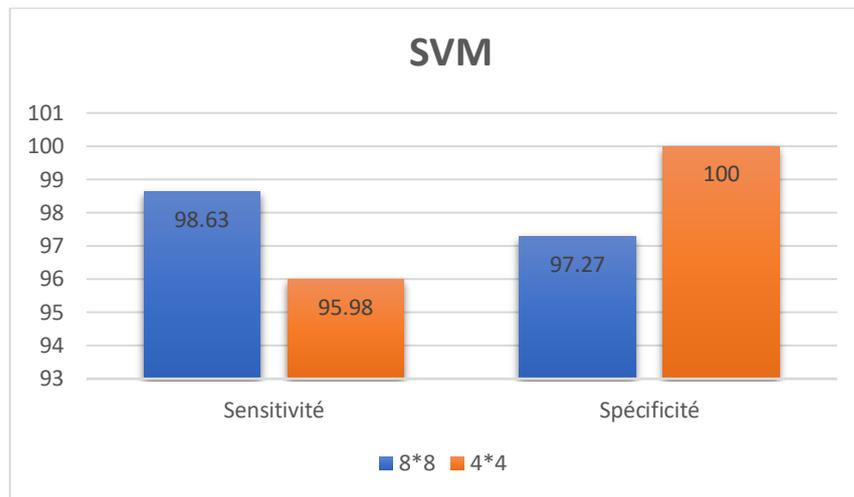


Figure 37 : Résultats de classification des places avec HOG associée à SVM

IV.5.2 Les résultats de détection en temps réel

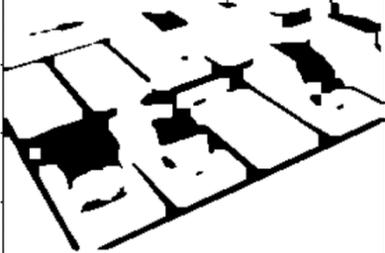
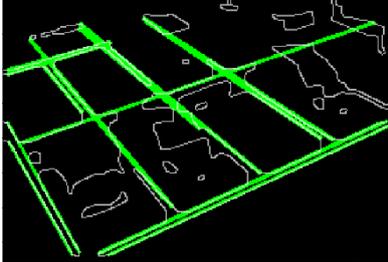
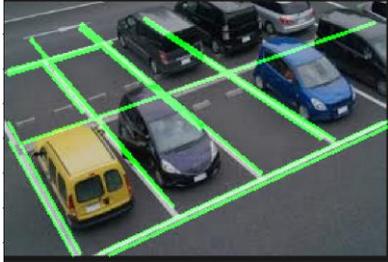
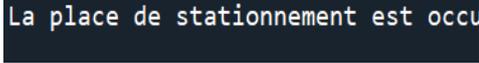
	
1-Image originale	2-Détection de contours-Canny
	
3- Détection de contours-Canny	4- Tous les contours sur l'image
	
5-Place extraite	6-Résultats

Figure 38 : Les résultats de détection en temps réel

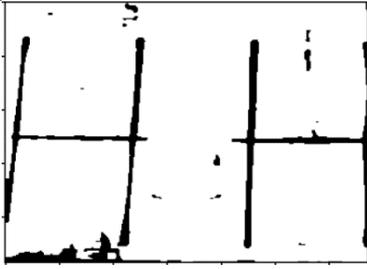
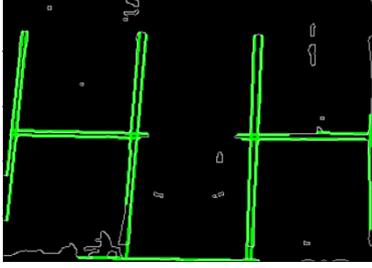
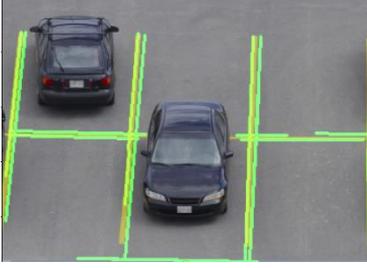
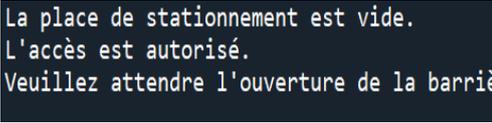
	
<p>1-Image originale</p>	<p>2-Détection de contours-Canny</p>
	
<p>3- Détection de contours-Canny</p>	<p>4- Tous les contours sur l'image</p>
	
<p>5- Détection de la place</p>	<p>6-Résultats</p>

Figure 39 : Les résultats de détection en temps réel (Suite)

L'analyse des résultats obtenus est scindée en deux parties à savoir la segmentation de la place de stationnement, et la classification de la place :

#### IV.5.2.1 La segmentation de la place de stationnement :

La segmentation de la place dans l'image capturée a souvent été bien réalisée, ceci peut être observé à travers les figures 39. Cependant il y a des cas où la détection ne se fait pas correctement, ceci se voit à la figure 38. On peut citer quelques causes qui font que la segmentation peut faillir :

- Les contours de la place ne sont pas suffisamment francs (l'intensité lumineuse des contours de la place ne change pas fortement par rapport à celles des voisins).
- La surexposition ou la sous-exposition de lumière sur scène, c'est à dire un contraste insuffisant, ce qui produit des contours non fermés ou même absence de contours.

#### IV.5.2.2 La classification de la place :

L'algorithme Séparateurs à vaste marge (SVM) nous a donné de bons résultats de classification (vide/occupée) sous diverses conditions.

## Chapitre 4 : Implémentation du Système SI

---

Cependant il y a des cas où la classification ne se fait pas correctement. Les deux raisons probables des résultats erronés sont les suivantes :

- L'espace entre les places de stationnement est très réduit.
- La position de l'image capturée diffère des images utilisées dans notre base de données d'apprentissage.

### IV.6 Les coûts :

Pour évaluer le travail sur ce projet, nous avons réalisé ce tableau qui contient les différents coûts :

*Tableau 4 : Les coûts*

Volume horaire de travail /Durée / Etude	Environ 600 heures/4 mois
Compétences exploitées	Electronique, le système RaspberryOS. Linux, Traitement d'Image, langage Python, l'intelligence artificielle
Programmation	Environ 300 lignes
Coût financier	Environ 25000 DZD

### IV.7 Conclusion :

Dans ce chapitre nous avons présenté la conception et l'implémentation de notre système RAPI, les résultats obtenus par le système, ainsi des futures améliorations sur le système.

## Conclusion générale :

Ce projet de fin d'étude avait pour ambition de concevoir un prototype de maquette d'une barrière de parking et de mettre en place un Système Embarqué de stationnement intelligent (SI) afin de détecter la disponibilité d'une place dans un parking en utilisant des techniques de machine learning comme le SVM (Support Vector Machine) et le KNN (KNearest Neighbors).

Il a fallu dans un premier temps se renseigner sur les méthodes et les techniques de traitement d'image pour la détection des places de stationnement, ainsi que la segmentation de la place de stationnement. Nous avons choisi la méthode de détection de contour de Canny car elle est très utilisée et donne de bons résultats

Ensuite nous avons choisi un dispositif sur lequel le système doit être implémenté. Ce dispositif doit répondre à certains critères comme : la robustesse, la connectivité avec le monde réel (capteurs, caméras, actionneurs...). Donc nous avons utilisé le Raspberry PI 4 model B. Puis nous avons étudié la conception et l'implémentation du système.

Le système réalisé dans ce PFE a montré de bonnes performances, cependant il y a quelques points qui nécessitent des améliorations que nous allons présenter dans les perspectives.

## Perspectives :

Dans le but d'améliorer notre système RAPI et le rendre plus fiable, nous pensons à :

- L'utilisation des techniques de prétraitements plus complexes pour avoir des résultats de traitement (détection de contours) plus propres.
- La détection de présence des véhicules par caméra (traitement vidéo).
- L'adaptation du système pour une utilisation durant la nuit (éclairage automatique et ajustable).
- Les données collectées par notre système seront utilisées pour fournir des informations en temps réel sur la disponibilité des places de stationnement aux conducteurs via une application mobile dédiée.

## Références Bibliographiques :

[1]. A.Tauvy-N.Carayon-S.Soissons, « traitement des images numériques », école nationale supérieure des télécommunications, Paris.

[http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egalhisto/rapport.htm#\\_Toc36969443](http://www.tsi.enst.fr/pages/enseignement/ressources/mti/egalhisto/rapport.htm#_Toc36969443)

[2]. <https://fr.wikipedia.org/wiki/Image>. Dernière modification le 16 juin 2024

[3] « Image numérique », Cour en ligne, Université rennes2, France. URL :[https://www.sites.univrennes2.fr/arts-spectacle/cian/image\\_numFlash/pdf/chap3\\_tout.pdf](https://www.sites.univrennes2.fr/arts-spectacle/cian/image_numFlash/pdf/chap3_tout.pdf)

[4]. <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>

[5]. [https://fr.wikipedia.org/wiki/%C3%89chantillonnage\\_\(signal\)](https://fr.wikipedia.org/wiki/%C3%89chantillonnage_(signal)). Dernière modification le 18 mai 2023.

[6]. [https://fr.wikipedia.org/wiki/Quantification\\_\(signal\)](https://fr.wikipedia.org/wiki/Quantification_(signal)). Dernière modification le 18 septembre 2023

[7]. M.Lossendière, « caractéristiques d'une image numérique », 31 Aout 2016.

URL : <http://www.lossendiere.com/2016/08/31/caracteristiques-dune-image-numerique/>

[8]. <http://fpt113-vg.espaceweb.usherbrooke.ca/dochtml/caracteristique-image.htm>

[9]. [https://fr.wikipedia.org/wiki/Traitement\\_d%27images](https://fr.wikipedia.org/wiki/Traitement_d%27images). Dernière modification le 19 octobre 2023.

[10]. <https://en.wikipedia.org/wiki/SFpark>

[11]. NordaStelo. 15 /04/2021 .Le futur du stationnement intelligent. Lien :

<https://geniecollectif.norda.com/le-futur-du-stationnement-intelligent/>

[12]. Indoor PGS-UltrasonicSeries .lien:

<https://www.akeparking.com/parkingsolutions/indoorparkingguidance/UltrasonicSeries/list/37.html>

[13].<https://www.ornikar.com/permis/conseilsconduite/stationnement/amenagements/parkings-automatisees>

[14]. Système de reconnaissance des plaques d'immatriculation.Lien :

<https://equinsaparking.com/fr/solutions-de-gestion/systeme-de-reconnaissance-des-plaquesdimmatriculation/>

[15]. Pay&Walk .lien : <https://www.smartparking.com/smartpark-system/pay-walk>

[16]. DAVID PERUCHA. 29/11/2017 .lien : <https://www.planet-sansfil.com/lora/smartparking-se-developpe-libelium-technologie-lorawan/>

[17]. <https://www.mokosmart.com/fr/smart-parking-system-using-iot/>

---

## Références Bibliographiques

---

[18]. L'intégration des technologies numériques dans la gestion dynamique du stationnement 03/05/2017 .lien :

<https://aqtr.com/association/actualites/lintegrationtechnologiesnumeriques-gestion-dynamique-stationnement>

[19]. [https://fr.wikipedia.org/wiki/Raspberry\\_Pi](https://fr.wikipedia.org/wiki/Raspberry_Pi). Dernière modification le 29 mai 2024.

[20]. BINAIRE, « Raspberry pi :la petite histoire d'une grande idée », Le monde.fr, 28 Décembre 2015.

URL : <https://www.lemonde.fr/blog/binaire/2015/12/28/raspberry-pi-la-petite-histoire-dunegrande-idee/>

[21]. <https://www.dzduino.com/uk-raspberry-pi-4-modele-b-soc-bcm2711-ram-2go-ddr4-usb-3-0-poe?search=Raspberry%20pi&description=true>

[22]. <https://robocraze.com/blogs/post/all-about-servo-motor-sg90>

[23].

[24]. [https://mataucarre.fr/index.php/2017/05/24/capteur-de-proximite-infra-rouge-fc-51-arduino/#google\\_vignette](https://mataucarre.fr/index.php/2017/05/24/capteur-de-proximite-infra-rouge-fc-51-arduino/#google_vignette)

[25]. <https://fr.wikipedia.org/wiki/Linux>.Dernière modification le 18 juin 2024.

[26].[https://fr.wikipedia.org/wiki/Python\\_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)).Dernière modification le 5 juin 2024.

[27]. <https://en.wikipedia.org/wiki/OpenCV>. Dernière modification le 19 février 2024.

[28]. <https://fr.wikipedia.org/wiki/Scikit-learn>. Dernière modification le 23 avril 2024

[29]. <https://datascientest.com/intelligence-artificielle-definition#:~:text=Une%20intelligence%20artificielle%20ou%20IA,penser%20comme%20un%20%C3%AAtre%20humain>.

[30]. <https://datascientest.com/machine-learning-tout-savoir>

[31]. [https://fr.wikipedia.org/wiki/Histogramme\\_de\\_gradient\\_orient%C3%A9](https://fr.wikipedia.org/wiki/Histogramme_de_gradient_orient%C3%A9)

[32]. [https://fr.wikipedia.org/wiki/Machine\\_%C3%A0\\_vecteurs\\_de\\_support#](https://fr.wikipedia.org/wiki/Machine_%C3%A0_vecteurs_de_support#)

[33]. <https://kobia.fr/classification-metrics-precision-recall/>