

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ MOULOUD MAMMERI DE TIZI OUZOU



FACULTÉ DE GÉNIE ÉLECTRIQUE ET INFORMATIQUE  
DÉPARTEMENT D'INFORMATIQUE

## Mémoire de fin d'études de master académique

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes informatiques

Rédigé par : Yacine HADJAR et Silia MEDRAR

### THÈME

## VÉHICULE INTELLIGENT POUR LA DÉTECTION DES PLAQUES D'IMMATRICULATION SUSPECTES

Mémoire soutenu publiquement le  
08/09/2019 devant le jury composé de :

Président : **M<sup>r</sup> Sami Sadi**

Examineur : **M<sup>r</sup> Mohamed Ramdani**

Encadré par : **M<sup>r</sup> Idir Filali**

< 2018 / 2019 >

## Remerciements

Avant tous on remercie dieu le tout puissant de nous avoir donné la santé, la force et le courage mais surtout la volonté d'entamer et de terminer ce travail.

On exprime notre profond et sincère remerciement à mon promoteur Mr FILALI, pour avoir cru en nos capacités pour mener à bien notre travail, pour le temps qu'il nous a consacré ainsi que pour ses remarques et conseils.

On tient également à remercier Mr RAMDANE pour ses précieux conseils et encouragement. On accorde un grand merci aussi à nos chères familles et Les personnes qui nous ont aidées et soutenues durant tout le long de notre cursus.

Nos vifs remerciements iront aussi aux membres de jury qui me fera l'honneur d'évaluer ce mémoire de fin d'études.

## Dédicaces

On dédie ce valeureux mémoire :

A nos chers parents, aucune dédicace ne saurait exprimer notre respect, notre amour éternel et notre reconnaissance pour les sacrifices qu'ils ont consenti pour notre instruction et notre bien-être. On espère que leurs bénédictions nous accompagneront toujours.

À nos frères et tous nos proches qui nous ont soutenu.

À nos amis qui nous ont toujours encouragé.

À tous ceux qui nous aiment.

## Sommaire

Introduction Générale .....	1
CHAPITRE I : LA VISION PAR ORDINATEUR.....	1
I. Introduction .....	1
II. Système visuel humain .....	1
III. Vision par ordinateur .....	3
IV. Traitement d'image.....	4
IV.1. Image .....	4
IV.1.1. Définition d'une image .....	4
IV.1.2. Types D'images.....	4
IV.1.2.1. Image en pixel (Matricielle) .....	4
IV.1.2.2. Image Vectorielle .....	5
IV.1.3. Caractéristiques d'une image.....	5
IV.1.3.1. Dimension.....	6
IV.1.3.2. Résolution .....	6
IV.1.3.3. Profondeur.....	7
IV.1.3.4. Bruit.....	7
IV.1.3.5. Luminance.....	8
IV.1.3.6. Contraste .....	9
IV.1.3.7. Photométrie.....	9
IV.1.4. Image numérique.....	10
IV.1.4.1. Définition .....	10
IV.1.4.2. Diversité des images .....	10
a) Nombre de dimensions .....	11
b) Hétérogénéité des dimensions.....	11
c) Dimension de leurs éléments .....	11
d) Espace de valeur.....	11
IV.1.4.3. Codage de couleur .....	12
a) Noir et Blanc.....	12
b) Niveau de gris .....	12
c) Couleur.....	12
IV.1.4.4. Histogramme d'une image .....	13

a) Définition .....	13
b) Propriétés .....	13
IV.1.4.5. Image en couleur .....	14
IV.2. Traitement d'images numériques.....	16
IV.2.1. Définition du traitement d'image.....	16
IV.2.2. Segmentation d'image .....	16
IV.2.3. Filtrage d'image .....	17
V. Application de la vision par ordinateur .....	17
VI. Conclusion.....	18
CHAPITRE II : TRAITEMENT D'IMAGE ET OCR.....	22
I. Introduction .....	22
II. Détection d'objets.....	22
II.1. Correction des bruits.....	23
II.2. Filtrage.....	23
II.2.1. Principe du filtrage .....	24
II.2.1.1. Filtrage global.....	24
II.2.1.2. Filtrage Local.....	24
II.2.2. Types de filtrage.....	25
II.2.2.1. Filtre passe-bas .....	25
II.2.2.2. Filtre passe-haut.....	25
II.2.3. Techniques de filtrage.....	26
II.2.3.1. Filtres linéaires.....	26
a) Filtre moyenneur.....	27
II.2.3.2. Filtres non linéaires.....	28
a) Filtre médiane .....	28
b) Filtres morphologiques.....	28
II.2.4. Avantage du filtrage.....	30
II.3. Segmentation.....	31
II.3.1. Les types de segmentation d'images.....	31
II.3.1.1. Basée sur les pixels.....	31
II.3.1.2. Basée sur les régions .....	31
II.3.1.3. Basée sur les contours.....	31
II.4. Seuillage .....	32

II.5.	Détection de contours.....	32
II.5.1.	Type de contours.....	33
II.5.2.	Méthode de détection des contours.....	34
II.5.2.1.	Méthode de gradient.....	34
II.5.2.2.	Méthode de laplacien.....	36
III.	Introduction au monde artificiel.....	37
III.1.	Intelligence Artificielle (IA).....	37
III.2.	Machine Learning (ML).....	37
III.3.	Deep Learning (DL).....	38
IV.	Reconnaissance d'objet (plaques D'immatriculations) .....	39
IV.1.	Anciennes méthodes de détection et de reconnaissance de plaque d'immatriculation.....	39
IV.2.	Reconnaissance optique de caractères .....	40
IV.2.1.	Domaine d'application.....	40
IV.2.2.	Principe de fonctionnement .....	41
IV.2.2.1.	Étape de prétraitement.....	42
a)	Redressement .....	42
b)	Nettoyage de bruits.....	42
c)	Segmentation des caractères.....	42
IV.2.2.2.	Reconnaissance des caractères.....	42
IV.2.3.	Quelques méthodes utilisées pour la détection des caractères.....	43
IV.2.3.1.	L'algorithme K Nearest Neighbors (KNN) .....	43
IV.2.3.2.	Support Vector Machine (SVM).....	44
IV.2.3.3.	Les réseaux neuronaux.....	45
IV.2.3.4.	Arbres de decision .....	46
IV.3.	Systèmes existants de la reconnaissance des plaques d'immatriculations .....	47
IV.3.1.	Asia Vision Technology Limited (AVT).....	47
IV.3.2.	LAPI Engine.....	47
IV.3.3.	SeeTec (AGL).....	48
V.	Conclusion.....	48
CHAPITRE III : OUTILS LOGICIELS ET MATÉRIELS .....		49
I.	Introduction .....	49
II.	Tesseract.....	49

II.1.	Définition .....	49
II.2.	Mode de fonctionnement .....	50
II.2.1.	Seuillage adaptatif.....	50
II.2.2.	Analyse des composants connectés.....	50
II.2.3.	Recherche des lignes et des mots .....	50
II.2.3.1.	Recherche des lignes .....	50
II.2.3.2.	Ajustement de ligne de base .....	51
II.2.3.4.	Recherche de mots proportionnels.....	51
II.2.4.	Reconnaissance des mots.....	52
II.2.4.1.	Séparer les caractères collés .....	52
II.2.4.2.	Associer les caractères brisés.....	52
II.2.5.	Classification des caractères .....	53
II.2.5.1.	Caractéristiques.....	53
II.2.5.2.	Classification.....	53
II.2.5.3.	Données d'entraînement.....	54
II.2.5.4.	Analyse linguistique.....	54
III.	Systèmes Embarqués .....	55
III.1.	Introduction.....	55
III.2.	Définition .....	55
III.3.	Caractéristiques spécifiques d'un système embarqué .....	56
III.4.	Domaine d'application des systèmes embarqués .....	57
IV.	Logiciel.....	58
IV.1.	Python.....	58
IV.2.	Python IDLE.....	59
IV.3.	OpenCV .....	59
IV.4.	Base de données .....	62
IV.5.	Json .....	63
IV.6.	Multiprocessing.....	63
IV.7.	Multithreading.....	63
IV.8.	Socket .....	64
IV.9.	Module python utilisés .....	65
IV.9.1.	Cv2 .....	65
IV.9.2.	Pickle .....	65

IV.9.3.	Numpy.....	65
IV.9.4.	Pytesseract.....	65
IV.9.5.	Struct.....	66
IV.9.6.	Math.....	66
IV.9.7.	Concurrent.futures .....	66
IV.9.8.	ThreadPoolExecutor .....	66
IV.9.9.	OS.fork .....	67
V.	Matériel.....	67
V.1.	Raspberry Pi.....	67
V.1.1.	Introduction .....	67
V.1.2.	Les microcontrôleurs.....	67
V.1.3.	Arduino.....	68
V.1.4.	Raspberry Pi .....	68
V.1.5.	Un tableau comparatif .....	69
V.1.6.	Les Pins (brochage) de la Raspberry Pi.....	70
V.1.7.	Quelque exemple de projets nécessitant plutôt un Arduino ou un Raspberry	71
V.1.8.	Systèmes d'exploitation sur les Raspberry Pi.....	72
V.1.8.1.	Raspbian.....	72
V.1.8.2.	Windows 10 IoT Core .....	73
V.1.8.3.	Kali Linux.....	73
V.2.	Camera .....	74
V.3.	Breadboard.....	75
V.4.	Moteur.....	76
V.5.	L298N .....	76
V.6.	Ultrason (HC-SR04).....	78
VI.	Conclusion.....	79
CHAPITRE IV : Implémentation et Réalisation .....		79
I.	Introduction .....	79
II.	Fonctionnement de l'application .....	79
II.1.	Software .....	79
II.1.1.	Reconnaissance de plaques.....	80
II.1.2.	Détection des marquages au sol des routes .....	83
II.1.3.	Détection des feux de signalisation.....	84

II.2. Hardware.....	85
II.2.1. Détection de distance et d'obstacles.....	86
III. Test.....	89
IV. Conclusion.....	90
Conclusion Générale.....	91

## Liste des figures

Figure 1: Le champ de vision d'un être humain .....	3
Figure 2: Image réelle et représentation machine [8].....	4
Figure 3: Image matricielle et image vectorielle [2].....	5
Figure 4: Image de dimension 20x20.....	6
Figure 5: Exemple de deux différentes résolutions d'une image.....	7
Figure 6: Exemple d'une image affectée par un bruit [9] .....	8
Figure 7: Contraste d'une image.....	9
Figure 8: La photométrie [10] .....	9
Figure 9: Système de coordonnées [6] .....	10
Figure 10: Image à niveau de gris et histogramme correspondant [6].....	13
Figure 11: Deux différentes images avec le même histogramme [6].....	14
Figure 12: Exemple d'une image en couleur [7].....	14
Figure 13 : Respectivement, canal Rouge, Vert et Bleu.....	15
Figure 14: Représentation par filtre matriciel des canaux RVB [7].....	15
Figure 15: Segmentation d'une image.....	17
Figure 16: Exemple de filtre sur une image [11].....	17
Figure 17: un exemple de filtrage linéaire.....	26
Figure 18 : application d'un filtre moyenneur (k=3, k=9) sur une image [16].....	27
Figure 19 : exemple d'un filtre médian.....	28
Figure 20: Différents exemple sur la dilatation [17].....	29
Figure 21: Différents exemple sur l'érosion [17] .....	30
Figure 22 : exemple de discontinuité [18].....	33
Figure 23 : Types de contours : Marche d'escalier, Rampe, Toit, Ligne [37] .....	33
Figure 24 : principe des filtres médian.....	35
Figure 25: Résultats d'un filtre moyenneur horizontal, vertical, et module [33].....	35
Figure 26: Relation entre IA, ML, DL [38] .....	38
Figure 27 : moteur de reconnaissance optique de caractères Tesseract [75] .....	49
Figure 28 : Un mot coupé à hauteur fixe. [41].....	51
Figure 29 : Un espacement difficile entre les mots. ....	51
Figure 30 : Points candidats pour le découpage. [41].....	52
Figure 31 : Un mot facilement reconnu en utilisant Tesseract. [41].....	52
Figure 32 : a) "h" sain, b) "h" cassé, c) des caractéristiques adaptées aux prototypes. [41]...53	53
Figure 33 : schéma résumant le Mode fonctionnement de Tesseract. ....	54
Figure 34 : logo Python [73] .....	59
Figure 35 : logo OpenCV [74] .....	60
Figure 36: brochage de Raspberry Pi 3 B+ [68].....	71
Figure 37: Logo du système d'exploitation Raspbian [69] .....	72
Figure 38: Logo du système d'exploitation Windows 10 IoT Core [71].....	73
Figure 39 : Logo du système d'exploitation Kali Linux [63] .....	74
Figure 40 : Les ports DSI et CSI sur une raspberry Pi 3 .....	74
Figure 41 : Platine Labdec (Breadboard).....	75
Figure 42 : une figure illustrant les composant d'un moteur à courant continu [66] .....	76
Figure 43: Double pont-H (L298n) [70] .....	77
Figure 44 : capteur ultrason [72].....	78

Figure 45 : fonctionnement du capteur ultrason [72].....	78
Figure 46 : schema resumant la decoposition des taches afin de reconnaitre les plaques d'immatriculations.....	80
Figure 47 : Diagramme résumant les étapes de reconnaissance des plaques d'immatriculations.....	80
Figure 48 : resultat de l'image en noir et blanc et en binaire.....	81
Figure 49 : detection de contours et elimination de bruits.....	81
Figure 50 : detection des regions susceptible d'etre des plaques d'immatriculation.....	82
Figure 51 : Isolation des plaques d'immatriculations.....	82
Figure 52 : représentation des caractères détectés sur l'image en une chaine numérique.....	82
Figure 53 : représentation des caractères détectés sur l'image en une chaine numérique.....	83
Figure 54 : Detection des marquages au sol des routes.....	84
Figure 55 : détection des feux signalisation.....	85
Figure 56 : le prototype du véhicule intelligent.....	85
Figure 57 : obstacle detecte et affichage de la distance.....	87
Figure 58 : route droite.....	87
Figure 59 : route barrée par un obstacle.....	88
Figure 60 : Exemple de virage gauche et droite.....	88
Figure 61 : résultat afficher lors de la détection d'un véhicule suspect.....	89
Figure 62 : détection d'une plaque d'immatriculation suspecte.....	89

## Liste des tableaux

Tableau 1 : avantage et inconvénients de ces trois types de segmentations .....	32
Tableau 2 : les matrices de convolutions de Robert, Sobel et Prewitt.....	36
Tableau 3 : les divers domaines d'utilisation des systèmes embarqués.....	57
Tableau 4 : comparaison entre les propriétés de l'Arduino UNO et quelques Modèles de Raspberry Pi.....	69
Tableau 5 : résumé du fonctionnement d'un L298n .....	77

## Résumé

Avec la nette croissance du marché automobile et le manque d'efficacité des moyens classiques existants, il est devenu très difficile de contrôler et sécuriser le trafic routier, chose qui fait ressentir le besoin de recourir à d'autres alternatives et de nouvelles technologies encore plus infaillibles. Un véhicule intelligent doté d'un système de reconnaissance de caractères, permettant la lecture et reconnaissance automatique des plaques d'immatriculation, semble être un excellent procédé pour pallier à certains problèmes de gestion et de sécurité dans les routes.

Cependant, notre projet vise à réaliser un prototype de voiture autonome utilisant un Raspberry Pi comme puce de traitement. Une caméra HD ainsi qu'un capteur à ultrasons sont utilisés pour fournir les données nécessaires du monde réel à la voiture.

Le véhicule est capable d'atteindre la destination donnée en toute sécurité et intelligemment, évitant ainsi le risque d'erreurs humaines. Ce véhicule sera doté d'un système de reconnaissance de plaques d'immatriculations, qui est une technique d'identification qui utilise des méthodes de traitement d'images et de vision par ordinateur pour lire les plaques minéralogiques des voitures.

Afin de réaliser ce travail, on a dû passer en revue plusieurs notions et concepts notamment en traitement d'images, vision par ordinateur, La reconnaissance optique de caractères (OCR), électronique, systèmes embarqués...etc

## Introduction Générale

Assurer la sécurité des routes et des conducteurs, a toujours été l'une des principales missions des différents organismes s'occupant de la gestion et de la sécurité du réseau routier. Ce besoin passe par la mise en place de systèmes tels que : les radars, les barrages routiers, les points de contrôles ...etc.

Cependant, avec la nette croissance du marché automobile ces dernières années, il est devenu très difficile de gérer ce gigantesque réseau, de plus les moyens classiques cités, semblent inefficaces face à certains problèmes qui demandent le discernement et plus de précision, dont on peut citer : détection de voiture volées, l'usurpation de plaques d'immatriculation... les systèmes tels que les radars, sont immobiles et leur mode de fonctionnement ne permet pas la reconnaissance des caractères des plaques d'immatriculations, de ce fait, ils ont pas la capacité d'authentifier les véhicules suspects afin de les signaler.

Afin de combler le manque d'efficacité des moyens déjà à disposition, on a besoin de se tourner vers des idées novatrices, simples, rapides et plus faciles à mettre en place.

Nous avons pensé à réaliser un mécanisme à la fois efficace et pratique. Notre solution se présente sous forme d'une voiture capable de rouler intelligemment sans être télécommandée, et qui sera dotée d'un système qui permet la lecture automatique des plaques d'immatriculation. Ce procédé permet d'isoler et de capturer l'image d'une plaque minéralogique, d'en extraire les caractères et de les convertir en données numériques afin de les analyser et de les comparer avec une base de données et cela se fait de manière continue et en temps réel.

Ce projet, va nous permettre de subvenir aux besoins d'identification et de classification des véhicules empruntant les routes, ce qui va simplifier par la suite la gestion du trafic et l'identification des automobilistes et/ou des véhicules suspects.

# CHAPITRE I

## LA VISION PAR ORDINATEUR

# CHAPITRE I : LA VISION PAR ORDINATEUR

## I. Introduction

La vision est une faculté dont l'être humain est doté dès sa naissance, cette faculté innée est, entre autres, le pouvoir de discerner et reconnaître les formes, les couleurs et les textures du monde qui nous entoure. Toutes ses informations seront après traitées et interprétées par le cerveau pour qu'elles soient ensuite utilisées.

De nos jours, l'intelligence artificielle parvient de plus en plus à reproduire divers comportements ou facultés humaines, la vision est une des facultés qui captivent les chercheurs, et grâce aux différentes avancées technologiques, les chercheurs ont pu offrir la possibilité d'avoir des aperçus du monde par le biais d'appareils d'acquisition d'image (appareils photo, caméras). Cependant, l'émergence de la vidéo comme média d'information s'accompagne d'un besoin croissant d'interprétations des informations contenues dans ces dernières à savoir de permettre à la machine (l'ordinateur) de comprendre ce qu'elle voit et cela en analysant, classifiant et quantifiant le contenu de chaque image. C'est ce qui nous mène à la vision par ordinateur qui est une discipline dont les premières bases théoriques ont été énoncées vers les années 60.

Depuis, étant donné le spectre très large d'application qui peut être envisagé, la vision par ordinateur a vite fait de dépasser le cadre restreint des laboratoires de recherche et a gagné du terrain dans le domaine applicatif. En effet cette technologie est maintenant répandue dans de nombreux domaines de la vie tel que : l'industrie, la médecine, chirurgie, le militaire, l'aérospatiale, et surtout dans le domaine de la sécurité et de la surveillance (du trafic routier par exemple) ce qui fait l'objet d'étude dans notre travail.

## II. Système visuel humain

Les applications de vision par ordinateur sont intéressantes et utiles, mais les algorithmes sont intensifs en calcul. Avec l'avènement du cloud computing, nous avons beaucoup plus de puissance de traitement pour travailler avec. Mais avant de parler sur la vision par ordinateur, nous devons comprendre pourquoi ces fonctions ont été

construites en premier lieu. Il est important de comprendre le fonctionnement du système visuel humain afin de pouvoir développer les bons algorithmes.

Le but des algorithmes de Vision par ordinateur consiste à comprendre le contenu des images et des vidéos que les humains réussissent à faire sans effort ! Alors, comment pouvons-nous faire en sorte que les machines le fassent avec la même précision ?

L'œil humain saisit toutes les informations, telles que la couleur, les formes, la luminosité, etc...

Voici quelques points à connaître :

- Notre système visuel est plus sensible au contenu basse fréquence qu'au contenu haute fréquence. Une faible teneur en fréquence se réfère à des régions planes où les valeurs de pixels ne changent pas rapidement et le contenu à haute fréquence se réfère à des régions avec des coins et des bords, où les valeurs de pixel fluctuent énormément. On remarque souvent qu'il est facile de voir s'il y a des taches sur une surface plane, mais il est difficile de repérer quelque chose comme cela sur une surface très texturée.
- L'œil humain est plus sensible aux changements de luminosité que le changement de couleur.
- Notre système visuel est sensible au mouvement. Nous pouvons rapidement reconnaître si quelque chose bouge dans notre champ de vision même si nous ne sommes pas directement en le regardant.
- Nous avons tendance à noter mentalement les points saillants de notre champ de vision. Considérons une table blanche avec quatre pieds noirs et un point rouge à l'un des coins de la surface de la table. Lorsqu'on la regarde, on notera immédiatement que la surface et les jambes ont des couleurs opposées et qu'un point rouge apparaît sur l'un des coins. Notre cerveau est vraiment intelligent de cette façon. [1]

Pour avoir une idée du champ de vision, sur la figure 1 nous pouvons comprendre les angles sous lesquels nous voyons diverses choses :

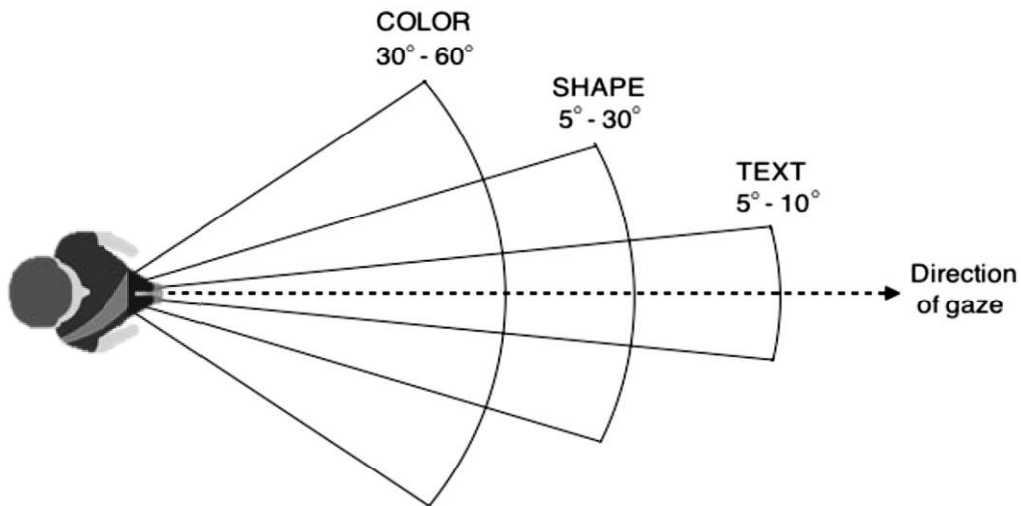


Figure 1: Le champ de vision d'un être humain

### III. Vision par ordinateur

La vision par ordinateur est un domaine inclus dans le rayon de l'intelligence artificielle qui propose plusieurs biais et méthodes d'analyse, de classifications et de traitements dans le but de comprendre l'histoire qui se déroule dans une image.

En tant qu'humains, c'est assez simple. Mais pour les ordinateurs, la tâche est extrêmement difficile... Pour atteindre ce but et permettre cette compréhension à l'ordinateur, la vision par ordinateur se réfère à des algorithmes variés et fait appel à plusieurs technologies.

On distingue deux types de traitement d'images auxquels la vision par ordinateurs fait recours :

- La vision haut niveau dont le but d'extraire les attributs symboliques, par exemple la reconnaissance des lettres écrites à la main.
- La vision bas niveau (ou vision pré-attentives) qui traite une grande quantité de pixels et les transforme en attributs nécessaire pour la vision haut niveau, par exemple l'extraction des contours, détection des régions, texture, etc.

En effet, la vision par ordinateur fait recours aux techniques de traitement d'images vu que le principal paramètre d'entrée de cette dernière est l'image.

## IV. Traitement d'image

### IV.1. Image

#### IV.1.1. Définition d'une image

Une image est la projection d'un ensemble de rayons lumineux provenant du monde extérieur sur un plan. Les blocs de construction d'une image sont les pixels.

D'un point de vue humain, une image contient plusieurs informations sémantiques, mais d'un point de vue mathématique ce n'est qu'une matrice de nombres représentant un signal.

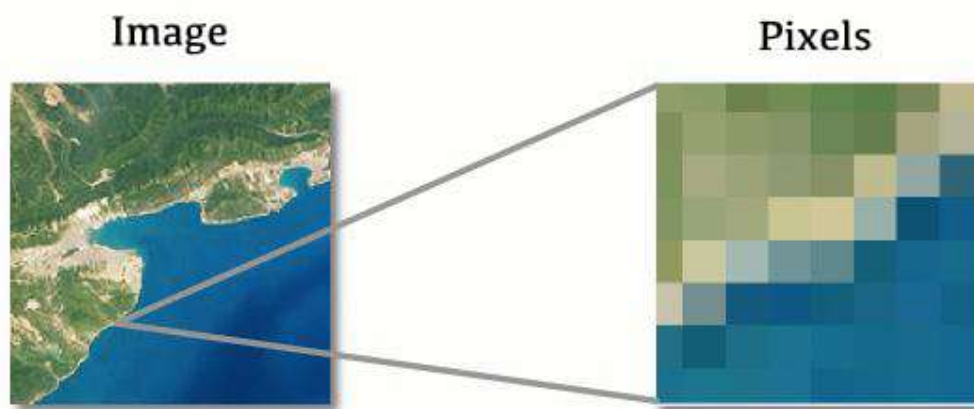


Figure 2: Image réelle et représentation machine [8]

Il existe trois principaux types d'images :

- Les images binaires (uniquement en noir et blanc) et dont la valeur de chaque pixel est soit 0 soit 1.
- Les images en niveaux de gris.
- Les images couleurs.

#### IV.1.2. Types D'images

##### IV.1.2.1. Image en pixel (Matricielle)

Chaque image est constituée d'un ensemble de pixels, qui sont les blocs de construction bruts d'une image. On associe à un pixel une valeur de couleur, d'entropie, de variance, etc.

Si nous considérons une image comme une grille, chaque carré de la grille contient un seul pixel. Par exemple, supposons que nous avons une image avec une résolution de 500 x 300. Cela signifie que notre image est représentée comme une grille de pixels, avec 500 lignes et 300 colonnes. Au total, il y a  $500 \times 300 = 150.000$  pixels dans notre image.

#### IV.1.2.2. Image Vectorielle

Une image vectorielle est dépourvue de matrices, elle est composée de ligne, de segments qui sont liés entre eux à l'aide de formule mathématique. En d'autres termes elle est composée d'objets géométrique individuels (segments de droites, polygones, arcs, cercles, etc.).

Grâce à la vectorisation, chaque élément a une place bien définie ce qui fait que l'image vectorielle est redimensionnable sans déformation ou perte de qualité, contrairement à une image matricielle. [2]

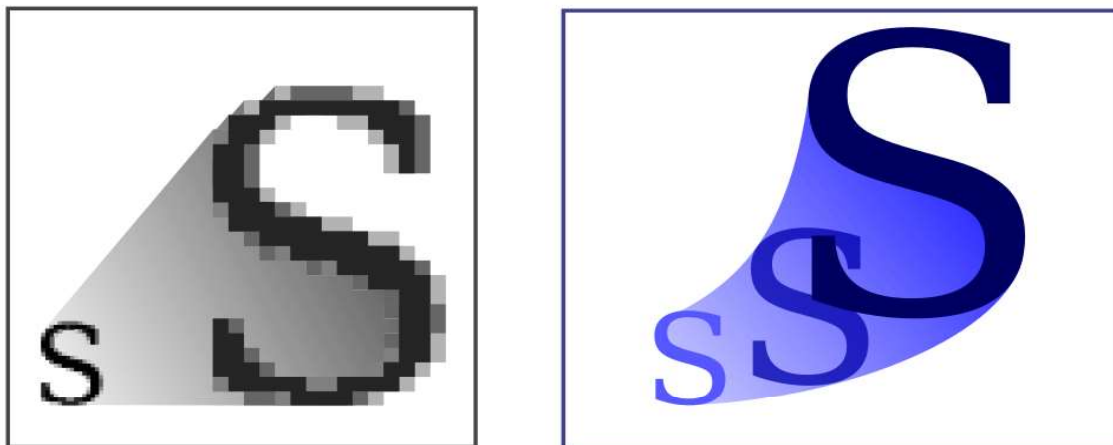


Figure 3: Image matricielle et image vectorielle [2]

#### IV.1.3. Caractéristiques d'une image

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

#### IV.1.3.1. Dimension

La dimension d'une image correspond à la taille de l'image, cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels).

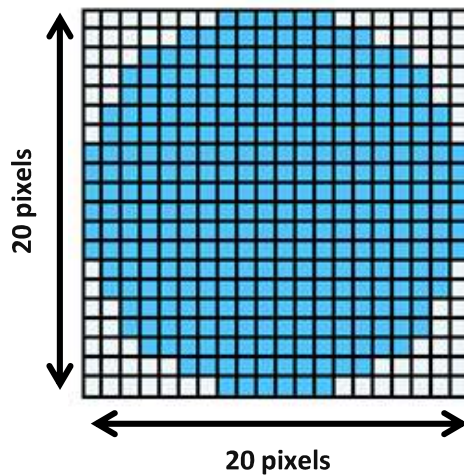


Figure 4: Image de dimension 20x20

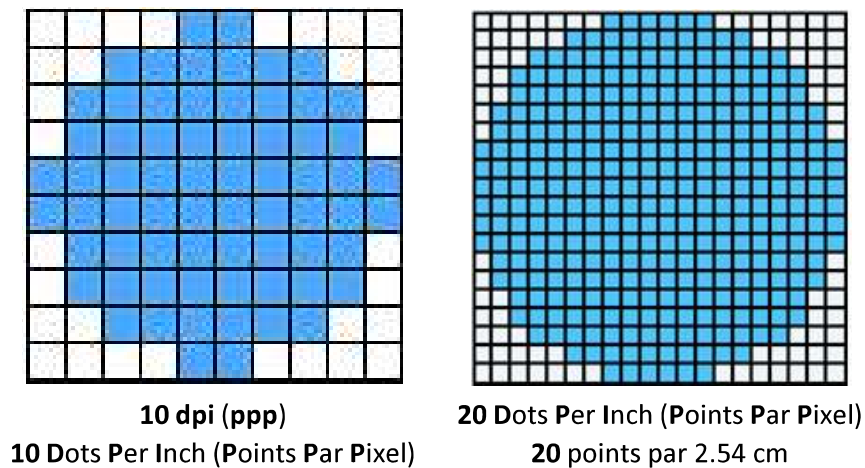
Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image. Une image possédante 10 colonnes et 10 lignes aura une dimension  $100=10 \times 10$ .

#### IV.1.3.2. Résolution

La résolution d'une image est le nombre de pixels par pouce qu'elle contient (1 pouce = 2.54 centimètres). Elle est exprimée en « PPP » (point par pouce) Correspond au nombre de pixels par pouce.

Un point est considéré comme un pixel et la résolution dépend de la taille de l'image. La résolution d'une image permet de définir la finesse de l'image. Plus il y a de pixels (ou points) par pouce et plus il y aura d'information dans l'image (plus précise).

Par exemple, une résolution de 300dpi signifie que chaque pouce de l'image comporte 300 pixels dans sa largeur et 300 pixels dans sa hauteur, il est donc composé de 90 000 pixels (300x300 ppp).



*Figure 5: Exemple de deux différentes résolutions d'une image*

#### IV.1.3.3. Profondeur

La profondeur désigne la quantité d'informations chromatiques disponibles pour chaque pixel dans une image (bits par pixel). Le nombre de couleurs disponibles et la précision de la représentation des couleurs dans une image sont proportionnels au nombre de bits d'informations par pixel. Ainsi, dans une image d'une profondeur de 1 bit par pixel, les pixels peuvent prendre deux valeurs possibles (noir et blanc) Une image avec une profondeur de 8 bits (1octet) par pixel compte 256 valeurs possibles (niveau de gris). [3]

Pour les images RVB, chaque canal de couleur peut avoir un octet pour chaque pixel et donc on obtient plus de 16 millions de valeurs chromatiques possibles :  $256^3 = 16\ 777\ 216$ .

#### IV.1.3.4. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

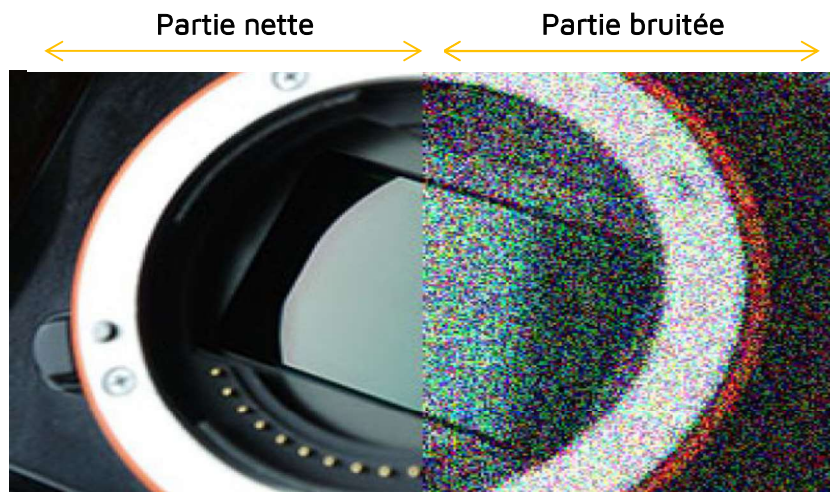


Figure 6: Exemple d'une image affectée par un bruit [9]

#### IV.1.3.5. Luminance

La luminance correspond au degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. [4]

$$L = \frac{I}{S} \quad (1)$$

Avec :

- L : luminance
- I : intensité lumineuse
- S : la surface

Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir. Ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

#### IV.1.3.6. Contraste

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



Figure 7: Contraste d'une image

#### IV.1.3.7. Photométrie

La photométrie correspond à la mesure des grandeurs relatives aux rayonnements, évaluer selon l'impression visuelle produite par la perception d'un rayonnement lumineux par l'œil humaine de la luminosité, ou clarté, d'une surface.

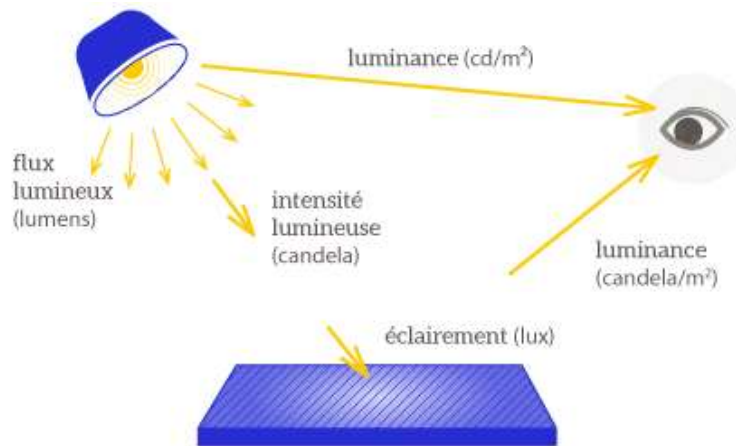


Figure 8: La photométrie [10]

Cette science étudie plusieurs grandeurs telles que l'intensité lumineuse, le flux lumineux, la luminance et l'éclairement. Les unités correspondantes étant la Candela, le Lumens et la Candela par mètre carré, Lux. [5]

## IV.1.4. Image numérique

### IV.1.4.1. Définition

Une image numérique est définie comme une fonction  $f(m, n, \dots)$  à valeurs discrètes et finies (mais pas forcément scalaires), et dont les coordonnées  $m, n, \dots$  sont également à valeurs discrètes et finies.

$$f : N^d \rightarrow R^b \quad (2)$$

Avec : **N** : nombres naturels,  
**R** : nombres réels,  
**b** : nombre de valeur par pixel,  
**d** : dimension.

$$m, n, \dots \rightarrow f(m, n, \dots) \quad (3)$$

Ainsi par exemple, une image à niveaux de gris correspond à  $d = 2$  (l'image a deux dimensions) et  $b = 1$  (il n'y a qu'une valeur par pixel : le niveau de gris).

Pour une image couleur classique,  $d = 2$  et  $B = 3$  (il y a trois bandes : rouge, vert, bleu). Enfin, une image IRM correspond à  $d = 3$  (l'image est 3D) et  $b = 1$ .

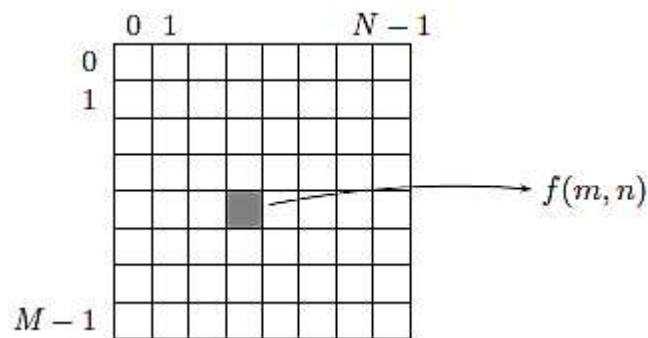


Figure 9: Système de coordonnées [6]

Dans le cas d'une image à deux dimensions  $f(m, n)$  de taille  $M \times N$  on adopte le système de coordonnées représenté dans figure précédente : le pixel (0,0) est situé en haut à gauche.

### IV.1.4.2. Diversité des images

Les images numériques peuvent se distinguer sur plusieurs aspects [6] :

### a) Nombre de dimensions

Une image peut posséder deux dimensions (photographie) et voir même 3 dimensions (IRM). Les éléments d'une image 2D sont appelés pixels (picture element) et ceux d'une image 3D sont appelés voxels (volume element).

### b) Hétérogénéité des dimensions

Dans une image classique 2D, les deux dimensions de l'image sont des dimensions spatiales. Mais là encore, les différentes dimensions de l'image ne s'expriment pas forcément dans la même unité.

Par exemple, une image hyper spectrale est une image  $2D+\lambda$  (deux dimensions spatiales, une dimension dépendant de la longueur d'onde), une vidéo peut être considérée comme une image  $2D+t$  (deux dimensions spatiales, une dimension temporelle), et une séquence IRM peut être considérée comme une image  $3D+t$  (trois dimensions spatiales, une dimension temporelle).

### c) Dimension de leurs éléments

Dans une image, chaque élément (pixel) peut être scalaire ou vectoriel. Par exemple, les pixels d'une image 2D à niveaux de gris (grayscale image) ne contiennent qu'une seule valeur (l'intensité du gris).

Les pixels d'une photographie classique contiennent trois valeurs (correspondant aux proportions du rouge, du vert et du bleu). Les images issues des satellites Pléiades sont des images RVB-IR, donc contiennent quatre valeurs (rouge, vert, bleu et infrarouge).

### d) Espace de valeur

Les images grand public sont généralement à valeurs dans  $\{0, 1, \dots, 255\}$ , mais une « image binaire » (binary image) est à valeurs dans  $\{0,1\}$ . De manière général, on considère qu'une image est à valeurs dans  $\{i_1, \dots, i_L\}$  où  $i_L \in \mathbb{R}$  et  $L \in \mathbb{N}^*$

#### IV.1.4.3. Codage de couleur

##### a) Noir et Blanc

Dans ce cas un pixel d'une image n'a la possibilité d'avoir que deux valeurs distinctes le 0 pour le noir et le 1 pour le blanc.

##### b) Niveau de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images au niveau de gris, nous pouvons attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la « couleur » de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

##### c) Couleur

La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation.

**Exemple** : dans le model RGB, les images en couleur sont représentées par trois matrices de taille identique, donnant l'intensité du Rouge, Vert, et Bleu pour chaque pixel. Les éléments de ces trois matrices sont encore des entiers de 0 à 255. Il y a donc  $256^3 = 2^{24} = 16\ 777\ 216$  teintes possibles pour chaque pixel. L'image finale est obtenue en combinant les trois images. [6]

#### IV.1.4.4. Histogramme d'une image

##### a) Définition

L'histogramme d'une image numérique représente la distribution des valeurs des pixels. C'est la fonction discrète  $h$  telle que :

$$h(i) = n_i \quad (4)$$

Où :  $n_i$  est le nombre de pixels de l'image ayant l'intensité  $i$ , que l'on appelle « effectif ».

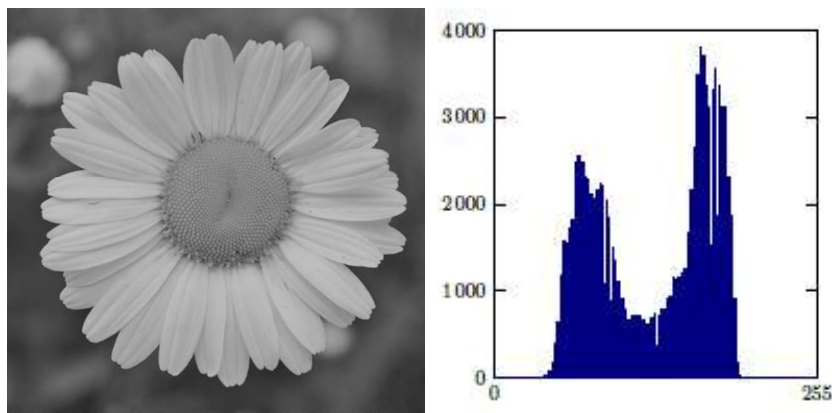


Figure 10: Image à niveau de gris et histogramme correspondant [6]

La figure 10 représente une image et son histogramme. On distingue sur celui-ci deux modes : celui de gauche correspond aux tons sombres de l'image (principalement le fond) et celui de droite correspond aux tons clairs (les pétales et le centre).

##### b) Propriétés

L'histogramme peut être interprété comme la densité de probabilité discrète des intensités si les effectifs sont normalisés par le nombre de pixels  $M \times N$  :

$$p(i) = \frac{n_i}{M \times N} \quad (5)$$

L'histogramme donne une information globale sur les intensités de l'image, mais perd l'information spatiale de l'image. Ainsi, deux images très différentes peuvent avoir le même histogramme. Les deux images sur la figure 11 ont le même histogramme. L'image de droite correspond en fait aux pixels de l'image de gauche ordonnés selon leur niveau de gris. [6]



Figure 11: Deux différentes images avec le même histogramme [6]

#### IV.1.4.5. Image en couleur

Une image en couleur est composée de plusieurs canaux : un rouge, un vert, et une composante bleue. Nous pouvons diviser l'image en ses composants respectifs.

La représentation de la répartition des intensités nécessite 3 histogrammes et sur chaque histogramme est représenté la distribution de la lumière d'une couleur.

**Exemple** : la figure 12 représente une image très « bleu » à cause de l'océan et du ciel. [7]



Figure 12: Exemple d'une image en couleur [7]



Figure 13 : Respectivement, canal Rouge, Vert et Bleu

**[Figure 13 - Droite]** Le canal rouge est très sombre. C'est logique, car la scène océanique contient très peu de couleurs rouges. Les couleurs rouges présentes sont soit très sombres, et donc non représentées, soit très claires, et font probablement partie de la mousse blanche de la vague et du soleil.

**[Figure 13 - Centre]** Le canal vert est davantage représenté dans l'image car l'eau de la mer contient des teintes verdâtres.

**[Figure 13 - Gauche]** Enfin, le canal bleu est extrêmement léger et très proche du blanc pur à certains endroits. Ceci est dû au fait des nuances de bleu qui sont fortement représentées sur l'image.

Une méthode alternative pour visualiser les canaux d'une image peut être vu dans la figure ci-dessous. Afin de montrer la couleur réelle de chaque canal, nous devons d'abord séparer l'image pour ne laisser que la couleur du canal à traiter et les autres canaux sont mis à zéro.



Figure 14: Représentation par filtre matriciel des canaux RVB [7]

Par exemple, afin de construire la représentation du canal rouge de l'image, on définit des matrices de zéros pour les canaux vert et bleu qui seront fusionnées avec

la matrice du canal rouge inchangé. Nous prenons des approches similaires pour la représentation des autres canaux. [7]

## **IV.2. Traitement d'images numériques**

### **IV.2.1. Définition du traitement d'image**

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées, qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information.

Quelques exemples de types d'informations qu'il est possible d'obtenir d'une image numérique :

- La luminance moyenne.
- Le contraste moyen.
- La couleur prédominante.
- Le taux d'acuité moyen (précis ou flou).
- Le taux d'uniformité des couleurs.
- La présence ou l'absence de certains objets.

### **IV.2.2. Segmentation d'image**

La segmentation consiste en la répartition de l'ensemble des pixels de l'image en groupe, chacun d'eux formera une région de l'image selon un critère d'homogénéité.

Cette technique de traitement d'image vise à séparer le plus précisément possible les différents objets se trouvant dans l'image traitée, et ainsi extraire de l'information.

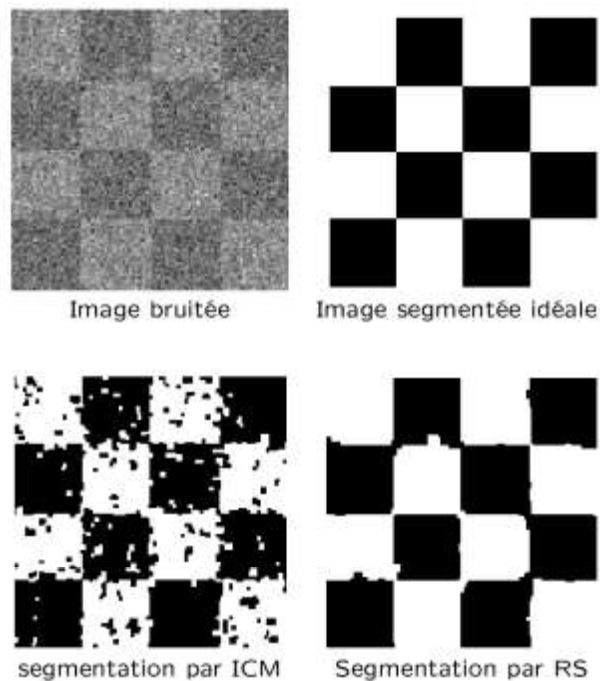


Figure 15: Segmentation d'une image

#### IV.2.3. Filtrage d'image

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

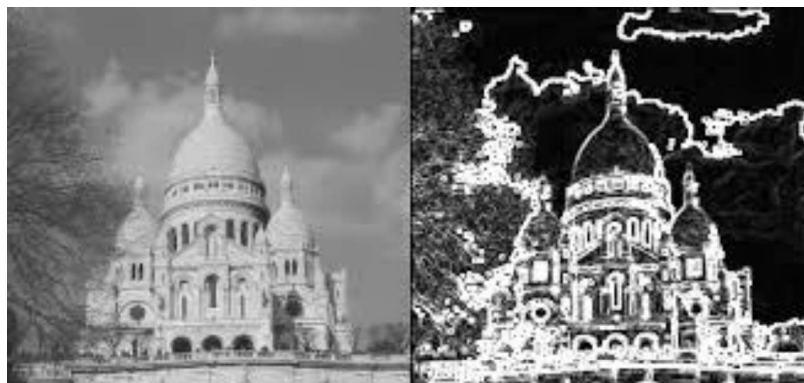


Figure 16: Exemple de filtre sur une image [11]

### V. Application de la vision par ordinateur

La vision par ordinateur est un domaine très jeune et en pleine croissance, dont le principal but est de permettre à une machine d'imiter la vision humaine grâce à

l'analyse, le traitement et la compréhension d'une ou plusieurs images prises par un système d'acquisition. Cette approche est utilisée dans divers domaines.

En effet l'application de la vision par ordinateur va de la vision industrielle (application de la vision par ordinateur aux procédés industriels, effectuer un contrôle de qualité sur des produit industriels automatiquement) à la recherche dans le domaine de la l'intelligence artificielle et des ordinateur ou robots capables de comprendre le monde qui les entoure.

La vision par ordinateur a su s'immiscer et elle est pleine expansion dans plusieurs domaines. Parmi ces domaines, nous citons :

- La robotique, la vision par ordinateur est utilisée pour la fabrication des robots autonomes capable d'identifier et de manipuler des objets et d'interagir avec le monde extérieur.
- La médecine, l'imagerie médicale est très importante comme le démontre le rôle de IRM (imagerie résonance magnétique) pour identifier des maladies et pour faire des recherches médicales.
- La surveillance, la reconnaissance faciale et la reconnaissance de personnes et leurs mouvements sont utile pour l'authentification et la sécurisation de di vers organismes.

Gestion de trafic routier, la reconnaissance automatique de véhicule de leurs types et de leurs plaques d'immatriculation permet de gérer et de sécuriser le trafic.

## VI. Conclusion

Nous avons introduit dans ce chapitre une branche du vaste domaine l'intelligence artificielle qui est la vision par ordinateur et quelques notion et concepts reliés à cette dernière que nous allons utiliser vers la suite dans la réalisation de notre projet. Nous avons aussi parler du traitement d'image et de ses quelques concepts de bases. Dans le prochain chapitre on va introduire les notions de détection et reconnaissance d'objet en général, puis on va se focaliser sur le but de sujet, les plaques d'immatriculation et la reconnaissance de caractère.

# CHAPITRE II

## TRAITEMENT D'IMAGE ET OCR

## CHAPITRE II : TRAITEMENT D'IMAGE ET OCR

### I. Introduction

Par le passé, quasiment toutes les tâches ou les fonctionnalités déductives et intelligentes étaient gérées manuellement par le biais de l'humain. La capacité du son cerveau à reconnaître d'une manière quasi parfaite plusieurs aspects du monde extérieur, que ça soit des visages, des objets, des caractères, même si ces derniers peuvent se trouver à une grande distance ou dans des conditions de luminosité extrême.

Cette capacité humaine surprenante a beaucoup captivé l'attention des chercheurs et des développeurs, ce qui les a poussés à œuvrer afin de reproduire ces fonctionnalités pour diverses applications qui réduisent considérablement la quantité de travail pour l'être humain avec un gain de temps immense.

Dans la suite de cette partie, nous allons nous intéresser à la détection et à la reconnaissance d'objets en générale et aux plaques d'immatriculations en particulier. Nous allons également passer en revue les méthodes les plus utilisées dans ce contexte.

Les plaques d'immatriculation sont l'un des principaux traits utilisés pour l'identification de véhicules ainsi que l'acquisition de diverses informations concernant ces derniers (Propriétaire, antécédents, wilaya d'appartenance...) ce qui permet de faciliter et de gérer plusieurs tâches à savoir la gestion du trafic routier, détection de fraude et la vérification automatiques et rapide d'antécédent d'un véhicule, etc.

### II. Détection d'objets

La détection d'objets est un domaine de recherche très actif dont le but est de localiser la position spatiale de régions d'intérêt correspondant à des parties d'objets sur une image numérique. Une attention particulière est portée à la détection de visage, la détection de personne et la détection de caractère. Ces méthodes proposent des applications dans de multiples domaines, tels la recherche d'image par le contenu ou la vidéo surveillance. La détection d'objet peut passer par plusieurs phases qu'on citera dans ce qui suit.

## II.1. Correction des bruits

En traitement d'image, toute brusque fluctuation d'un pixel par rapport à ses voisins est considérée comme un bruit provoquant ainsi la dégradation de la qualité d'image. Le bruit est gênant pour les traitements quand il faut extraire de l'information depuis l'image et cause des ennuis considérables à la reconnaissance. Prenons exemple de la reconnaissance des caractères, en effet à cause de ce dernier des caractères peuvent être reconnus à la place d'autres ou encore un pixel noir peut être assimilé comme une virgule et bien d'autres contraintes. Dans un souci d'optimisation des résultats de la reconnaissance, un traitement de réduction voire suppression des parasites "bruit" est nécessaire afin d'améliorer la netteté de l'image à traiter. [23]

Ce problème peut être provoqué par une ou plusieurs sources, on peut citer les plus fréquentes :

- **Bruit lié aux dispositifs d'acquisition** : La caméra, les câbles et tous autres blocs de connexion engendrent un bruit.
- **Bruit lié à la scène** : Il se trouve parfois que la scène soit polluée de poussière, ou couverte de nuages, brouillard ou buée qui dégrade la qualité des images. Cependant, là encore, la connaissance a priori du phénomène perturbateur permet d'envisager une modélisation et donc une correction systématique.
- **Bruit lié à la transmodulation** : Les images qui sont capturées à partir d'une séquence vidéo qui regroupe le son et l'image dans un même signal sont atteintes par un bruit dû à la transmodulation qui se produit du fait qu'on utilise une double modulation : modulation audio et vidéo. Il arrive, ainsi, des cas où des composantes du signal audio brulent notre signal vidéo.

Pour éliminer le bruit et rendre l'image plus nette, on utilise les techniques de filtrage, qui seront abordées dans la section suivante.

## II.2. Filtrage

Un filtre est souvent une transformation mathématique permettant pour chaque pixel de la zone à laquelle il s'applique de modifier sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients. Le filtre est représenté par un tableau

(matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre. Un exemple de filtre 3x3 :

$$\begin{bmatrix} P(x-1, y-1) & P(x, y-1) & P(x+1, y-1) \\ P(x-1, y) & P(x, y) & P(x+1, y) \\ P(x-1, y+1) & P(x, y+1) & P(x+1, y+1) \end{bmatrix}$$

### II.2.1. Principe du filtrage

Le filtrage consiste à modifier la valeur des pixels d'une image numérique, généralement dans le but d'améliorer son aspect, lorsque l'image ne peut pas être exploitée directement. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels tirés de l'image d'origine :

#### II.2.1.1. Filtrage global

Où chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Cette catégorie utilise, par exemple les opérations sur les histogrammes. [13]

#### II.2.1.2. Filtrage Local

Où chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire. [13]

**Gestion des bords** : Si le pixel considéré est proche du bord de l'image, certains points du voisinage sont en dehors de l'image d'origine. Il convient alors de choisir une stratégie pour gérer ces pixels extérieurs. Les stratégies couramment employées [14] :

- **Mise à zéro** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est considérée comme nulle. C'est-à-dire :

$$P[-1, y] = 0$$

▪ **Continuité** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel le plus proche qui est dans l'image d'origine. C'est à dire :

$$P [-1, y] = P [0, y]$$

▪ **Miroir** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel symétrique par rapport au bord de l'image. C'est à dire :

$$P [-1, y] = P [1, y]$$

▪ **Sphérique** : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel correspondant si l'image était projetée sur une sphère. C'est à dire :

$$P [-1, y] = P [Largeur-1, y]$$

### II.2.2. Types de filtrage

On distingue généralement les types de filtres suivants :

#### II.2.2.1. Filtre passe-bas

Consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de "lissage", "dé-bruitage" ou encore "filtre antibruit".

Le filtrage passe-bas supprime les éléments de l'image qui sont décrits par des hautes fréquences, c'est-à-dire les détails, les textures et les contours, Il réalise donc un lissage (ou floutage) de l'image.

➤ Diminue le bruit mais atténue les détails de l'image (flou plus prononcé).

#### II.2.2.2. Filtre passe-haut

À l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, et donc les différences entre les pixels voisins. C'est la raison pour laquelle le terme de "filtre d'accentuation" est parfois utilisé.

➤ Accentue les contours et les détails de l'image mais amplifie le bruit.

### II.2.3. Techniques de filtrage

Il en existe plusieurs, chaque filtre cherche à atténuer un type de défaut bien précis. Il n'y a pas de filtre universel capable de corriger tous les défauts. Il faut choisir les bons filtres suivant les défauts que nous désirons corriger.

#### II.2.3.1. Filtres linéaires

Le filtre local est dit linéaire (par convolution) si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage y compris le pixel en question.

La convolution est courante en traitement d'images. Elle consiste en une opération de multiplication de deux matrices de tailles différentes, mais de même dimensionnalité (par exemple 1D, 2D), produisant une nouvelle matrice (également de même dimensionnalité).

La convolution est donc un traitement qui s'effectue grâce à une matrice appelée matrice de convolution ou noyau (kernel). Le filtre parcourt toute la matrice principale de manière incrémentale et génère une nouvelle matrice constituée des résultats de la multiplication.

On notera que le filtrage linéaire consiste simplement à remplacer la valeur de chaque pixel

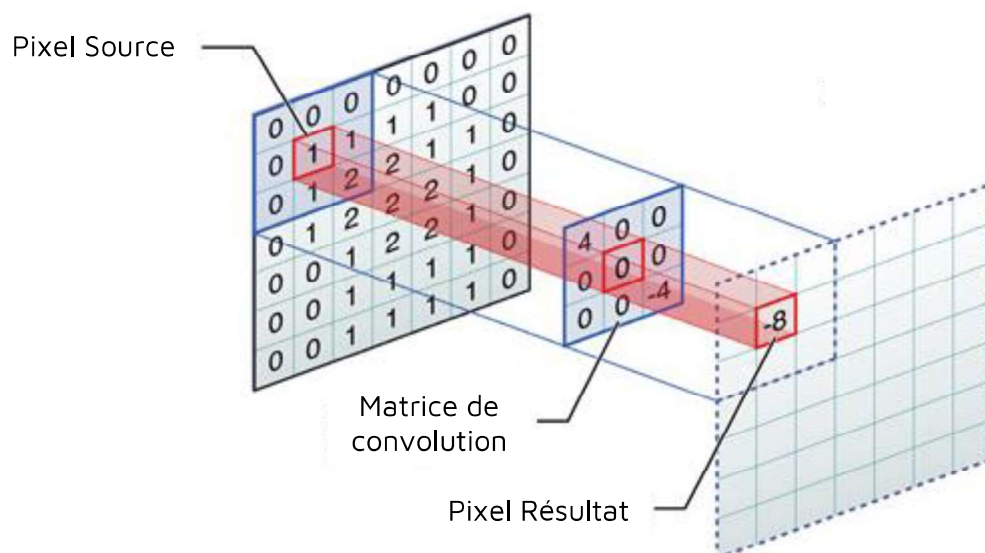


Figure 17: un exemple de filtrage linéaire

### a) Filtre moyenneur

Cette technique est la plus directe pour adoucir une image trop heurtée dans ses contours. Le niveau de gris de chaque pixel est remplacé par la moyenne des niveaux de gris de ses huit voisins. Ce filtrage pourra se faire à l'aide du filtre [15] :

$$H = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Le facteur 1/9 sert à normaliser le filtre de manière que celui-ci n'influe pas sur l'intensité globale de l'image. Cette technique a pour défaut de créer un flou sur les bords des objets. Pour une image d'entrée I on a :

$$I'(x,y) = I(x,y) * H(x,y) \quad (1)$$

$$I'(x,y) = \sum_{i=-w}^w \sum_{j=-w}^w I(x-i, y-j) \cdot H(i,j) \quad (2)$$

Les effets du filtre moyenneur varient avec la taille du noyau : plus les dimensions du noyau seront importantes, plus le bruit sera éliminé ; mais en contrepartie, les détails fins seront eux-aussi effacés et les contours étalés. Nous présentons ici 2 filtres moyenneurs de kernels 3\*3 et 9\*9 :



Image Normal

Filtre 3\*3

Filtre 9\*9

Figure 18 : application d'un filtre moyenneur (k=3, k=9) sur une image [16]

### II.2.3.2. Filtres non linéaires

Si le filtre ne peut pas être exprimé par une combinaison linéaire, il est appelé "non linéaire". Les filtres non-linéaires sont plus complexes à mettre en œuvre que les filtres linéaires (en termes de complexité algorithmique et de temps de traitement). Cependant les résultats obtenus avec les filtres non-linéaires sont très souvent de meilleure qualité que ceux obtenus par les filtres linéaires.

#### a) Filtre médiane

Consiste à remplacer l'intensité de chaque pixel par la valeur médiane de ses voisins choisis à l'avance. C'est un filtre passe bas qui élimine le bruit de type neige, mais il entraîne cependant une petite perte de résolution. Son principe est : Pour chaque pixel on classe les pixels voisins du pixel courant, par valeurs croissantes, on prend la valeur médiane des pixels classés et on l'affecte au pixel courant (voir figure 19). [16]

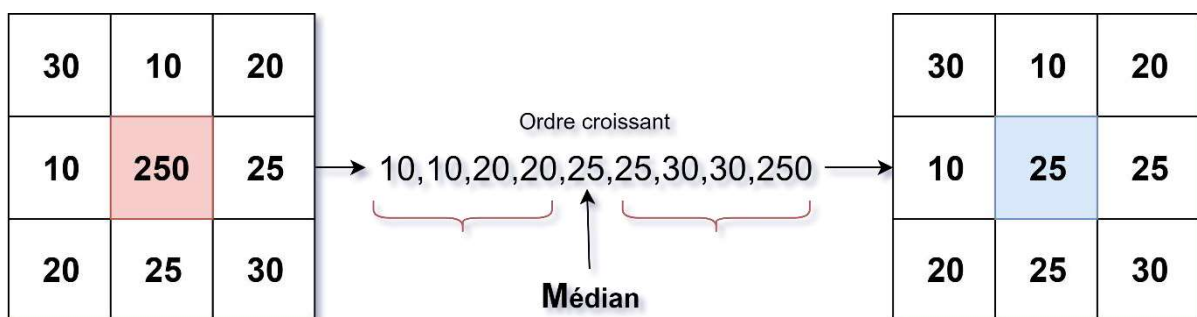


Figure 19 : exemple d'un filtre médian

#### b) Filtres morphologiques

C'est une classe d'opérateur de traitements d'images qui a été développée à l'origine pour des images binaires, mais qui a depuis été étendue aux images en niveaux de gris.

Les opérateurs morphologiques travaillent sur le voisinage local de chaque pixel. La forme de ce voisinage est appelée élément structurant. Les éléments structurants couramment utilisés sont le disque, isotrope, et le carré, qui permet d'accélérer les calculs. Parmi les opérateurs morphologiques, on distingue :

## ➤ Dilatation

L'opération de dilatation utilise la notion d'élément structurant qui généralise le concept de fenêtre glissante. L'idée est que, pour pouvoir étudier la forme des objets, il faut pouvoir considérer des fenêtres glissantes avec des formes plus complexe qu'un carré : par exemple des segments, des cercles, des triangles.

La dilatation d'une image en niveaux de gris, par une image binaire (l'élément structurant) est notée  $f \oplus S$ . Après dilatation, la nouvelle valeur d'un pixel est donnée par la valeur maximale des pixels sous l'élément structurant.

Comparé au filtre médian, la fenêtre n'est plus carrée et on prend l'élément max au lieu du médian. Afin de bien visualiser le résultat de cette opération, voyons l'exemple binaire suivant (voir figure 17) :

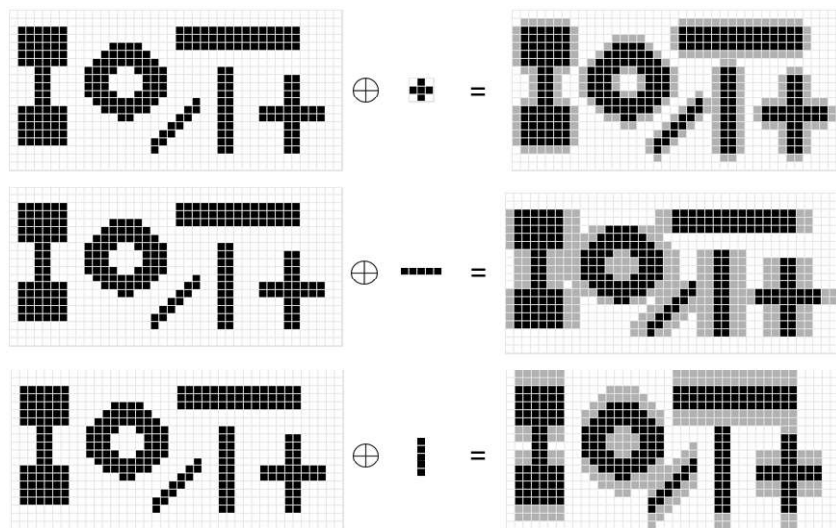


Figure 20: Différents exemple sur la dilatation [17]

## ➤ Érosion

L'érosion est obtenue de manière similaire à la dilatation en remplaçant l'élément Max par min. Soient une image à niveau de gris et une image binaire (l'élément structurant), l'érosion de  $f$  par  $S$  est notée  $f \ominus S$ .

Afin de bien visualiser le résultat de cette opération, voyons l'exemple suivant :

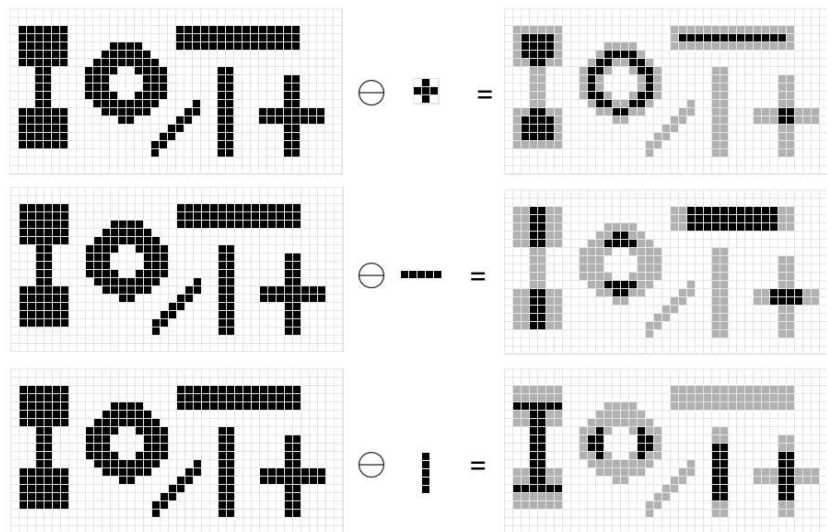


Figure 21: Différents exemple sur l'érosion [17]

L'érosion et la dilatation ont l'inconvénient de modifier fortement la taille des structures dans l'image. Pour réduire cet effet, on les utilise souvent en combinaison.

### ➤ Ouverture

On définit l'ouverture morphologique comme une érosion suivie d'une dilatation. L'ouverture de  $f$  par  $S$  est notée  $(f \ominus S) \oplus S$ , et a pour effets [17] :

- De faire disparaître les petites particules (dont la taille est inférieure à celle de l'élément structurant).
- De séparer les grosses particules aux endroits où elles sont plus fines.

### ➤ Fermeture

De manière duale on définit la fermeture morphologique comme une dilatation suivie d'une érosion. La fermeture de  $f$  par  $S$  est notée  $(f \oplus S) \ominus S$ , et a pour effets [17] :

- De faire disparaître les trous de petite taille dans les structures
- De connecter les structures proches

#### II.2.4. Avantage du filtrage

On citera comme avantage :

- Accentuer les variations d'intensité de l'image : renforcer les différences entre les pixels appartenant à des régions adjacentes (rehaussement de contraste).

- Détection de contours et préservation de la forme des régions.
- Renforcement de l'homogénéité à l'intérieur des régions (réduction de bruit).

## **II.3. Segmentation**

Les images sont composées de régions possédant des propriétés locales homogènes, lorsque nous segmentons une image, nous séparons les régions en fonction de diverses mesures, telles que la couleur, la texture, l'emplacement, etc. Tous les pixels de chaque région ont quelque chose en commun, en fonction de la métrique utilisée.

Il existe de très nombreuses méthodes permettant ce découpage, dont l'efficacité dépend avant tout de l'image.

### **II.3.1. Les types de segmentation d'images**

#### **II.3.1.1. Basée sur les pixels**

En utilisant les histogrammes de l'image l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image. On regroupe ainsi des endroits de l'image par couleur.

#### **II.3.1.2. Basée sur les régions**

Cette catégorie peut correspondre aux algorithmes d'accroissement de régions. Ce type de segmentation consiste à prendre un ensemble de petites régions uniformes dans l'image et à regrouper les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus possible. Cette catégorie peut aussi correspondre à des algorithmes de découpage de régions on part de l'image entière que l'on va subdiviser en plus petites régions tant que ces régions ne sont pas suffisamment homogènes. Les algorithmes dits « split and merge » sont un mélange de ces deux techniques. [18]

#### **II.3.1.3. Basée sur les contours**

Cette catégorie s'intéresse aux contours des objets dans l'image. Des filtres détecteurs de contours sont appliqués à l'image. Si l'image est très contrastée, le résultat est satisfaisant, mais dans le cas contraire, il faut utiliser des techniques de

reconstruction de contours. Cette technique est locale et en général trop limitée pour traiter des images complexes. [18]

Tableau 1 : avantage et inconvénients de ces trois types de segmentations

	Avantage	Inconvénients
<b>Segmentation basée sur les pixels.</b>	Plus homogènes Élimine les taches bruyantes.	Calculateur couteux. Sensible à l'initialisation.
<b>Segmentation basée sur les régions.</b>	Fermeture de frontières. Densité d'informations.	Difficulté de définir les critères pour fusionner ou diviser les pixels.
<b>Segmentation basée sur les contours.</b>	Rapidité. Précision.	Sensibilité au bruit.

#### II.4. Seuillage

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

Le seuillage d'image remplace un à un les pixels d'une image à l'aide d'une valeur seuil fixée (par exemple 123). Ainsi, si un pixel à une valeur supérieure au seuil (par exemple 150), il prendra la valeur 255 (blanc), et si sa valeur est inférieure (par exemple 100), il prendra la valeur 0 (noir). [19]

#### II.5. Détection de contours

La détection de contours a pour objectif l'acquisition d'une représentation compacte de toutes les propriétés significatives de la scène perçue. Elle est très utilisée comme étape de prétraitement pour la détection d'objets, de formes... afin de trouver les limites des régions.

La mise en évidence des points représentant ces contours dans une image peut servir à reconnaître des objets présents dans une scène, à différencier des zones de l'image, à faire de la segmentation d'images.

Du point de vue théorique, un contour est défini par une variation marquée de l'intensité d'un pixel à l'autre. Toute discontinuité n'est pas nécessairement située à la frontière entre deux objets. [15]

Exemples de discontinuités (Figure 22) :

- De profondeur.
- D'orientation de surface.
- De réflectance.
- D'illumination.



Figure 22 : exemple de discontinuité [18]

### II.5.1. Type de contours

Il existe plusieurs types de contours, on pourra citer :

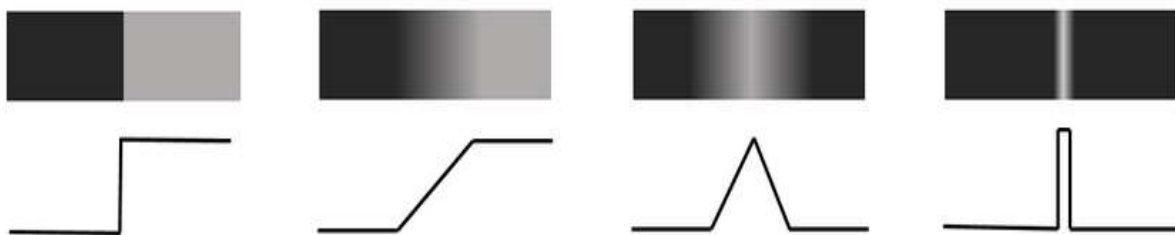


Figure 23 : Types de contours : Marche d'escalier, Rampe, Toit, Ligne [37]

## II.5.2. Méthode de détection des contours

Les contours des objets dans des images de dimension quelconques correspondent le plus souvent aux extrema locaux du gradient ou aux zéros du laplacien de la fonction de niveaux de gris.

Plusieurs méthodes existent pour détecter cette variation d'intensité, les unes plus ou moins complexes, les autres plus ou moins gourmandes en calculs. Dans la plupart des cas elle s'applique en deux étapes : la première permet de localiser les contours à partir :

- Des algorithmes basés sur le gradient (opérateurs du premier ordre).
- Des algorithmes basés sur le Laplacien (opérateurs du second ordre)

Ces algorithmes s'effectuent dans des directions privilégiées pour quantifier l'importance du contour. La seconde étape va permettre d'isoler les contours du reste de l'image à partir d'un seuillage judicieux.

### II.5.2.1. Méthode de gradient

On utilise la première dérivée pour calculer le gradient. Le gradient, en un pixel, est un vecteur caractérisé par un module et une direction.

- Le module du gradient permet de quantifier l'importance du contour mis en évidence, c'est-à-dire l'amplitude du saut d'intensité relevé dans l'image.
- La direction du gradient permet de déterminer l'arête présente dans l'image. En effet, la direction du gradient est orthogonale à celle du contour.

Le principe de la détection de contours par l'utilisation du gradient consiste à calculer d'abord le gradient de l'image dans deux directions orthogonales puis le module du gradient. Il s'agira ensuite d'effectuer une sélection des contours les plus marqués, c'est-à-dire les points de plus fort contraste par un seuillage adéquat, les directions des contours. [17] [19]

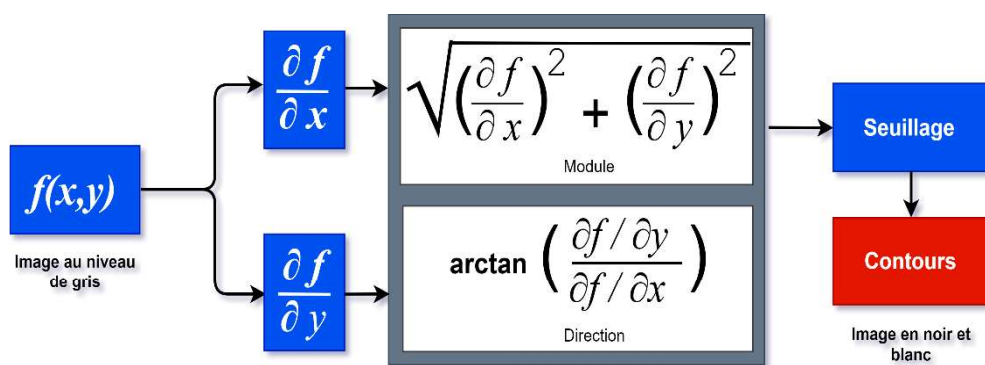


Figure 24 : principe des filtres médian

Sur l'exemple suivant, on peut observer une image et ses 2 dérivées partielles. On peut constater que les variations horizontales apparaissent dans la dérivée partielle  $\left(\frac{\partial f}{\partial x}\right)$ , et les variations verticales apparaissent dans la dérivée partielle  $\left(\frac{\partial f}{\partial y}\right)$  et le module.



Figure 25: Résultats d'un filtre moyennneur horizontal, vertical, et module [33]

En pratique, il n'existe pas de définition unique pour le calcul des dérivées partielles  $(\frac{\partial f}{\partial x})$  et  $(\frac{\partial f}{\partial y})$ . Plusieurs solutions exprimables sous forme de produit de convolution ont été proposées : les masques de Roberts, de Prewitt et de Sobel ...etc

Tableau 2 : les matrices de convolutions de Robert, Sobel et Prewitt

	TYPES DE MASQUE		AMPLITUDE	DIRECTION
Masque de Robert	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\sqrt{G_1^2 + G_2^2}$	$\theta = \frac{\pi}{4} + \arctan\left(\frac{G_2}{G_1}\right)$
	$G_1, G_2$			
Masque de Sobel	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	$\sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$
	$G_x, G_y$			
Masque de Prewitt	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$
	$G_x, G_y$			

### II.5.2.2. Méthode de laplacien

De la même façon, l'approximation par différences finies la plus simple de la dérivée seconde est la convolution par le noyau  $[1 \ -2 \ 1]$  pour l'approximation de  $(\frac{\partial^2 f}{\partial x^2})$  et  $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$  pour l'approximation de  $(\frac{\partial^2 f}{\partial y^2})$ . Le laplacien  $\Delta f = (\frac{\partial^2 f}{\partial x^2}) + (\frac{\partial^2 f}{\partial y^2})$  peut donc être approché par l'un des opérateurs linéaires suivants :

Laplacien discret - 4	Laplacien discret - 8	Laplacien de Robinson
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -2 & -1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$

Après le filtrage de l'image au moyen d'un de ce filtre, il s'agit de détecter les passages par zéro en conservant uniquement les passages les plus marqués. Mais cette

technique est particulièrement sensible au bruit, donc il suffit de ne pas le considérer. Ce bruit peut très bien se traduire par des oscillations autour de zéro. C'est le rôle du seuil S qui va être utilisé dans cette approche pour ne prendre en compte que les passages par zéro d'amplitude relativement élevée correspondant à des vrais contours de l'image. [19]

### III. Introduction au monde artificiel

#### III.1. Intelligence Artificielle (IA)

Historiquement, les débuts de l'IA remontent à Alan Turing dans les années 1950. [20]

L'Intelligence Artificielle (IA) est la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique. On y trouve donc le côté « Artificiel » atteint par l'usage des ordinateurs ou de processus électroniques élaborés et le côté « Intelligence » associé à son but d'imiter le comportement du cerveau humain. [21]

#### III.2. Machine Learning (ML)

Apparu dans les années 1980, le Machine Learning (ML) est l'application de méthodes statistiques aux algorithmes pour les rendre plus intelligents. Il repose donc sur la capacité des algorithmes à recevoir beaucoup de données et à « Apprendre » d'elle-même. [22]

Le machine learning est un domaine large, qui comprend de très nombreux algorithmes. Parmi les plus célèbres, on retrouve :

➤ Les régressions (linéaires, multivariées, polynomiales, régularisées, logistiques...) : ce sont des courbes qui approximent les données.

➤ L'algorithme de Naïve Bayes : l'algorithme donne la probabilité de la prédiction, sachant les événements antérieurs. [23]

➤ Les arbres de décision : en répondant à un certain nombre de questions sous la forme graphique d'un arbre et en suivant ses branches jusqu'aux réponses possibles

qui seront situées aux extrémités (les « feuilles » de l'arbre), on arrive à un résultat (avec un score de probabilité).

Ainsi que des algorithmes plus perfectionnés reposant sur plusieurs techniques de statistiques : Random Forest (une forêt d'arbres de décision qui votent), Gradient Boosting, Support Vector Machine...

### III.3. Deep Learning (DL)

En dépit de sa puissance, le ML a beaucoup de failles, on peut citer le fait qu'un expert humain doit, au préalable, faire du tri dans les données.

En 2010, L'idée du Deep Learning (DL) est née et s'inspire du fonctionnement du cerveau humain pour pouvoir pousser l'analyse plus loin et savoir extraire les données soi-même.

Le DL (qui est donc un sous-domaine du ML) repose sur ce qu'on appelle des réseaux de neurones artificiels, c'est-à-dire un ensemble de neurones qui s'envoient des nombres en fonction de leurs liaisons, jusqu'à des neurones de sortie. Grâce à cette architecture et une bonne approximation des données d'entrée, le DL est capable de reconnaître des visages, de synthétiser des textes ou encore de conduire une voiture autonome. [22]

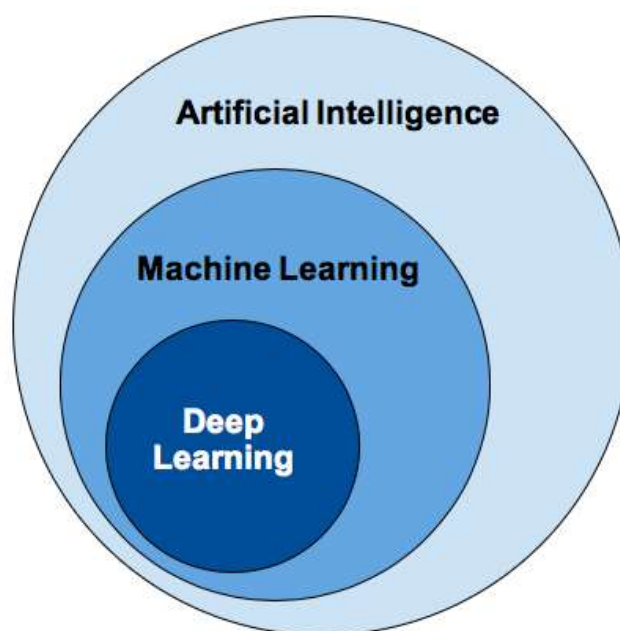


Figure 26: Relation entre IA, ML, DL [38]

En résumé : Le DL est du ML utilisé avec des réseaux de neurones. Le ML est de l'IA reposant sur des statistiques. L'IA est l'imitation d'une réponse d'un expert dans un domaine.

#### IV. Reconnaissance d'objet (plaques D'immatriculations)

Un système de reconnaissance automatique des plaques d'immatriculations robuste et efficace devient de plus en plus une nécessité dans la gestion de plusieurs domaines comme la sécurité routière, la gestion des parkings, contrôle routier, la poursuite des criminels.

Un système de reconnaissance automatique des plaques minéralogiques est une technologie qui trouve son essence dans ces 20 dernières années dans le développement des techniques de traitement d'image ainsi que dans les OCR (Reconnaissance Optique des Caractères).

##### IV.1. Anciennes méthodes de détection et de reconnaissance de plaque d'immatriculation

Les différents systèmes de détection de plaque d'immatriculation proposés sont basés sur des propriétés différentes. Certaines techniques utilisent des règles simples basées sur des méthodes déterministes tandis que d'autres emploient des systèmes d'apprentissage et de classification.

- L'approche de Hsieh, Yu et Chen réalisée 2002 utilise les opérations morphologiques sur l'image au niveau de gris. Cette approche se concentre sur les propriétés d'apparence locale des régions des plaques d'immatriculation telles que la luminosité, la symétrie et l'orientation. Les régions candidates sont comparées avec une image de plaque d'immatriculation donnée, elle se base sur la similarité de ces propriétés d'apparence. [24]
- Dans certains travaux, le problème de la localisation d'une plaque d'immatriculation est géré par la recherche des régions rectangulaires dans l'image. Les caractères d'une plaque d'immatriculation peuvent être directement identifiés en scannant la totalité de l'image d'entrée et en

cherchant les parties de l'images qui ne sont pas liés ou colées à d'autres parties de celle-là. Si un certain nombre de caractères se trouvent sur la même ligne droite, ils peuvent constituer une plaque d'immatriculation. Cette idée a été proposée par Lim, Yeo, Tan et Teo, en 1998. [25]

- Dans le travail de Jia, Zhang et He effectué en 2007, le déplacement moyen (mean shift) est utilisé pour filtrer et segmenter une image couleur du véhicule afin d'obtenir les régions candidates. Ces régions candidates sont ensuite analysées et classées afin de décider si une région candidate contient une plaque d'immatriculation ou non. [26]
- En 2010, Bayoumi et al ont utilisé la densité des contours et la couleur de fond pour trouver les numéros, les symboles et l'alphabet de la plaque d'immatriculation égyptienne. Leur système dépend des caractéristiques des numéros de la plaque. Puis, ils ont appliqué les fonctions de projection pour en extraire la plaque d'immatriculation. Ils ont utilisé un réseau de neurones afin de reconnaître les différents chiffres, symboles et alphabets. [27]

## **IV.2. Reconnaissance optique de caractères**

La reconnaissance optique de caractères est une technologie qui désigne les procédés informatiques permettant de reconnaître dans une image les caractères alphanumériques (lettre ou chiffre) composant un texte ou autre information textuelle.

En d'autres termes, le procédé OCR fournit aux systèmes d'imagerie et de numérisation (Appareil photo, scanner) la capacité de transformer des images de caractères imprimés à la machine en caractère lisible et compréhensible par la machine.

### **IV.2.1. Domaine d'application**

La technologie de reconnaissance optique des caractères, largement utilisée dans des applications commerciales depuis les années 1970. Elle permet aujourd'hui d'automatiser beaucoup de tâches, on peut citer quelques principales applications d'un OCR :

- Aide à la lecture pour les non-voyants.
- Lecture des tickets de transport.

- Lecture de formulaires. [28]
- Numérisation des documents.
- Lecture des passeports.
- Gestion automatique des chèques bancaires ou postaux.
- **Reconnaissance automatique des plaques d'immatriculation** : Pour diverses utilisations antérieures principalement pour des fins de sécurité, en effet, les OCR permettant la lecture des caractères d'une plaque ce qui permet l'identification du véhicule et par la suite de la personne possédant ce dernier ce qui permet la résolution de plusieurs fonctionnalités, à savoir la gestion automatique du trafic routier, des parkings, etc. [29]

#### IV.2.2.Principe de fonctionnement

Le principe de cette technologie est en effet, de lire un document (une image) et détecter les formes présentes dans ce dernier pour les comparer ensuite à une bibliothèque intégrée contenant des milliers d'échantillons possibles pour chaque forme, cette comparaison vise à faire correspondre chaque forme détectée à une forme de la bibliothèque pour en fournir le caractère alphanumérique correspondant.

Dans le cas où des erreurs surviennent lors de la reconnaissance, l'OCR fait recours à un autre procédé qui compare alors le mot entier au contenu d'un dictionnaire intégré pour en déduire l'équivalence la plus proche et ainsi corriger le caractère mal lu.

La technologie de reconnaissance optique de caractères (OCR) est la clé permettant d'accéder aux informations dans l'image d'un texte imprimé pour pouvoir l'exploiter ultérieurement.

Il existe plusieurs logiciels et implémentations d'OCR, ces derniers ont chacun une technologie et un raisonnement particulier, cependant ces derniers partagent tous un raisonnement quasi-identique et suivent des étapes d'une manière consécutive en utilisant leurs différentes technologies, ces étapes seront exposées dans ce qui suit :

#### **IV.2.2.1. Étape de prétraitement**

La première étape de reconnaissance par OCR consiste à numériser l'information destinée à la lecture, cette étape permet l'acquisition et la préparation des informations nécessaires pour représenter les données.

La qualité des résultats obtenus dépend amplement de cette dernière, il est donc nécessaire de prendre toutes les précautions permettant une bonne numérisation de la donnée.

##### **a) Redressement**

Il arrive parfois que l'image soit de travers et ceci pour multiple raisons à savoir : l'angle de prise de l'appareil d'acquisition, les conditions dans lesquelles se trouvait l'objet à numériser.

Le traitement à effectuer consiste à redresser l'image à l'aide d'algorithmes de correction d'orientation afin de rendre la lecture des caractères plus facile.

##### **b) Nettoyage de bruits**

Le bruit cause des ennuis considérables à la reconnaissance, en effet à cause de ce dernier des caractères peuvent être reconnus à la place d'autres ou encore un pixel noir peut être assimilé comme une virgule et bien d'autres contraintes.

Dans un souci d'optimisation des résultats de la reconnaissance, un traitement de réduction voire suppression des parasites "bruit" est nécessaire afin d'améliorer la netteté de l'image à traiter. Pour pallier à ces dégradations, faudra utiliser les filtres.

##### **c) Segmentation des caractères**

La segmentation permet d'isoler les éléments textuels, chiffre et caractères, pour la reconnaissance. Elle se base sur des mesures de plages blanches (interlignes et inter caractères) pour faire la séparation.

#### **IV.2.2.2. Reconnaissance des caractères**

La reconnaissance de caractères permet de se prononcer sur l'identité d'un caractère à partir d'un apprentissage de sa forme. Cette étape nécessite une étape

préalable de paramétrisation de la forme, définissant des données, des mesures, ou des indices visuels sur lesquels s'appuie la méthode de reconnaissance. Suivant la nature de ces informations, il existe plusieurs catégories de méthodes :

- Syntaxique (description par une grammaire),
- Structurale (description par un graphe),
- Statistique (description par partitionnement de l'espace).

Ces dernières ont de loin le plus grand intérêt avec les méthodes à base de réseaux de neurones, ou de modèles stochastiques. La complexité de la tâche vient de l'apprentissage qui nécessite, pour sa stabilité, d'un très grand nombre d'échantillons par classe, et de la recherche d'indices visuels discriminants.

### **IV.2.3. Quelques méthodes utilisées pour la détection des caractères**

#### **IV.2.3.1. L'algorithme K Nearest Neighbors (KNN)**

C'est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante "dis-moi qui sont tes voisins, je te dirais qui tu es".

Pour effectuer une prédiction, KNN n'a pas besoin de construire un modèle prédictif, l'algorithme KNN va se baser sur le jeu de données en entier. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les K instances du jeu de données les plus proches de notre observation. Ensuite pour ces n voisins, l'algorithme se basera sur leurs variables de sortie pour calculer la valeur de la variable de l'observation qu'on souhaite prédire. [30]

Par ailleurs :

- Si KNN est utilisé pour la régression, c'est la moyenne (ou la médiane) des variables des plus proches observations qui servira pour la prédiction.
- Si KNN est utilisé pour la classification, c'est le mode des variables des plus proches observations qui servira pour la prédiction.

## Limitations

- KNN est un algorithme assez simple à appréhender. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. Le contre coût est qu'il doit garder en mémoire l'ensemble des observations pour pouvoir effectuer sa prédiction. Ainsi il faut faire attention à la taille du jeu d'entraînement.
- Également, le choix de la méthode de calcul de la distance ainsi que le nombre de voisins peuvent ne pas être évident. Il faut essayer plusieurs combinaisons et faire des ajustements de l'algorithme pour avoir un résultat satisfaisant. [30]

### IV.2.3.2. Support Vector Machine (SVM)

C'est un algorithme d'apprentissage supervisé qui appartient à la catégorie des *classificateurs linéaires* (qui utilisent une séparation linéaire des données), et qui dispose de sa méthode pour trouver la frontière entre les catégories. Son but est de déterminer la catégorie à laquelle appartient un nouveau point de données.

Un algorithme SVM ne doit pas seulement placer les objets dans des catégories, mais avoir les marges entre eux aussi large que possible sur un graphique. [31]

Certaines applications de SVM incluent :

- Classification du texte.
- Classification des images.
- Reconnaître des caractères manuscrits.

## Limitations

- Avec une fonction de noyau appropriée SVM résout des problèmes complexe mais, ce n'est pas évident de Choisir une bonne fonction du noyau.
- Bien que les SVM aient de bonnes performances de généralisation, elles peuvent être extrêmement lentes en phase de test

- Difficile à comprendre et à interpréter le modèle final, poids variables et impact individuel.
- Comme le modèle final n'est pas si facile à voir, nous ne pouvons pas effectuer de petits étalonnages, ce qui rend difficile l'intégration de notre logique commerciale.
- Il n'est pas facile d'ajuster ces hyperparamètres. Il est difficile de visualiser leur impact. [31]

#### **IV.2.3.3. Les réseaux neuronaux**

Il s'agit d'un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Les réseaux de neurones tirent profit du partage des poids qui limite le nombre de coefficients à apprendre, et donc la complexité du modèle. Il peut donc être appliqué directement sur l'image, ce qui permet de s'affranchir de l'étape d'extraction de primitives.

C'est un atout intéressant, compte tenu du fait que le choix du jeu de primitives est souvent l'un des facteurs limitants d'un système de reconnaissance : le réseau de neurones apprend automatiquement sa propre extraction de caractéristiques en fonction des données d'apprentissage. [32]

### **Limitations**

- Les réseaux de neurones ne fournissent pas les explications concernant leurs résultats ce qui limite donc l'analyse des phénomènes existants. Cette limite est due à l'opacité des réseaux de neurones qui empêche une analyse pertinente des solutions obtenues.
- Les réseaux de neurones peuvent être assimilés à une boîte noire qui donne une réponse quand on lui fournit des données mais qui ne délivre pas toujours de justification simple à analyser.
- Les liens existants entre les variables du modèle ne sont pas toujours détectés. Ce système n'a donc qu'un pouvoir explicatif limité contrairement

à d'autres systèmes experts qui pourraient être capables de retracer le raisonnement suivi pour parvenir au résultat ce qui interpréterait donc les conclusions obtenues. [32]

#### **IV.2.3.4. Arbres de décision**

Un arbre de décision est un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles sont situées aux extrémités des branches (les feuilles de l'arbre), et sont atteints en fonction de décisions prises à chaque étape.

Pour la simplicité de sa compréhension et d'interprétation, il est utilisé dans des domaines variés tels que la sécurité, la fouille de données, la médecine, etc. Il a l'avantage d'être lisible et rapide à exécuter. Il s'agit de plus d'une représentation calculable automatiquement par des algorithmes d'apprentissage supervisé.

C'est un modèle boîte blanche : si l'on observe une certaine situation sur un modèle, celle-ci peut être facilement expliquée à l'aide de la logique booléenne, au contraire de modèles boîte noire comme les réseaux neuronaux, dont l'explication des résultats est difficile à comprendre. [33]

### **Limitations**

- Les algorithmes d'apprentissage par arbre de décision sont basés sur des heuristiques telles que les algorithmes gloutons cherchant à optimiser le partage à chaque nœud, et de tels algorithmes ne garantissent pas d'obtenir l'optimum global.
- L'apprentissage par arbre de décision peut amener des arbres de décision très complexes, qui généralisent mal l'ensemble d'apprentissage.
- Certains concepts sont difficiles à exprimer à l'aide d'arbres de décision (comme XOR ou la parité).
- Lorsque les données incluent des attributs ayant plusieurs niveaux, le gain d'information dans l'arbre est biaisé en faveur de ces attributs. Cependant, le problème de la sélection de prédicteurs biaisés peut être contourné par des méthodes telles que l'inférence conditionnelle. [33]

### IV.3. Systèmes existants de la reconnaissance des plaques d'immatriculations

Il existe de nombreux systèmes de reconnaissance de plaques d'immatriculations, tels que :

#### IV.3.1. Asia Vision Technology Limited (AVT)

Il s'agit du premier fournisseur mondial de solutions technologiques et de solutions de gestion intelligente des véhicules et des conteneurs. AVT a été un pionnier dans le développement et la fourniture de la technologie de reconnaissance optique de caractères (OCR). [34]

#### Particularités et avantages

- Le système a réussi à reconnaître les plaques d'immatriculation de 20 pays dans plusieurs langues.
- Le système détecte, reconnaît et vérifie automatiquement la plaque d'immatriculation.

#### IV.3.2.LAPI Engine

C'est un OCR de lecture de plaques d'immatriculation développé par la société ALPHA-Numéric permettant d'identifier et de tracer les véhicules passant sur une zone et se présentant devant une caméra, à partir d'un flux vidéo en direct ou de séquences vidéo d'événements détectés par les caméras.

Chaque événement (passage de véhicules) est tracé et enregistré dans un fichier xml accompagné d'une photo jpg horodatée de l'avant du véhicule. Les informations récupérées dans ce fichier peuvent ensuite être couplées avec d'autres données. [35]

#### Particularités et avantages

- La possibilité de crypter les données (images et numéro plaques) avec un système de gestion de clé unique.
- A lecture de tous types de plaques réfléchissantes aux infrarouges ou non.

### IV.3.3. SeeTec (AGL)

Il s'agit d'un module d'extension de SeeTec Cayuga qui permet la reconnaissance automatique de plaques d'immatriculation de véhicules à l'arrêt ou en mouvement. [36]

#### Particularités et avantages

- Il peut lire des formats de plaques internationales, même en caractères arabes et cyrilliques, jusqu'à huit voies de circulation par serveur.
- SeeTec peut être utilisé avec chaque caméra IP prise en charge par SeeTec et intégrée au système, même dans des conditions difficiles de luminosité :

#### V. Conclusion

Ce chapitre nous a permis de comprendre le cadre théorique général de la détection et reconnaissance d'objets. Nous avons présenté en particulier la reconnaissance optique de caractères (OCR) vu que c'est une étape primordiale pour la reconnaissance de plaque d'immatriculation.

Dans le chapitre suivant nous allons présenter des concepts qui nous a permis la réalisation de notre projet.

# CHAPITRE II

## OUTILS LOGICIELS ET MATÉRIELS

## CHAPITRE III : OUTILS LOGICIELS ET MATÉRIELS

### I. Introduction

Afin de réaliser notre projet, nous avons eu recours à plusieurs technologies logiciels et matériels notamment la bibliothèque de vision par ordinateur OpenCV, le langage de programmation python, une Raspberry pi 3b+, des moteurs ...etc. Dans ce présent chapitre, nous allons présenter les différentes bibliothèques, modules et matériels auxquels nous avons eu recours afin de concevoir notre véhicule intelligent qui, grâce à la reconnaissance des plaques numérogiques, peut reconnaître les véhicules suspects.

### II. Tesseract

#### II.1. Définition

Tesseract est un moteur de reconnaissance optique de caractères open source développé par les ingénieurs de HP (Hewlett Packard) entre 1984 et 1995, avant d'être abandonné.

Après 10 ans de purgatoire, son code est devenu ouvert en 2005, distribué sous Licence Apache, et son développement est repris sous la houlette de Google. [40]



*Figure 27 : moteur de reconnaissance optique de caractères Tesseract [75]*

Initialement limité aux caractères ASCII et il supportait parfaitement les caractères UTF-8, maintenant il est capable de reconnaître plus de 100 langues et propose un système d'apprentissage permettant d'en supporter plus. Ce moteur est compatible

avec un grand nombre de systèmes d'exploitation et possède un grand nombre de wrappers permettant sa manipulation depuis plusieurs langages de programmation.

Tesseract intègre également des algorithmes de pré-traitement tel le redimensionnement, la binarisation ou la réduction du bruit. Le moteur est également capable de fournir des résultats sous plusieurs format. [41]

## **II.2. Mode de fonctionnement**

### **II.2.1. Seuillage adaptatif**

Contrairement au seuil automatique<sup>1</sup>, le seuil adaptatif est calculé pour chaque pixel séparément. L'image est découpée en bloc NxN ensuite pour chaque bloc on calcule un seuil automatique puis on calcule le seuil de chaque pixel avec une moyenne pondérée des seuils locaux des blocs voisins en fonction de leurs distances.

### **II.2.2. Analyse des composants connectés**

L'étape est une analyse des composants connectés, dans laquelle les contours des composants sont stockés. En inspectant l'emboîtement des contours et le nombre de contours enfants et petits-enfants [42], il est simple de détecter le texte inversé ainsi que le texte blanc sur noir et de les reconnaître aussi facilement qu'un texte noir sur blanc. Les contours sont ensuite rassemblés, uniquement par imbrication, dans des Blobs<sup>2</sup>.

### **II.2.3. Recherche des lignes et des mots**

#### **II.2.3.1. Recherche des lignes**

L'algorithme de recherche de ligne est conçu de manière à ce qu'une page inclinée puisse être reconnue sans avoir à la dévier, évitant ainsi la perte de qualité d'image. Les éléments clés de ce processus sont le filtrage de blob et la construction de lignes.

---

<sup>1</sup> Seuil automatique est un seuil global donc calculé pour l'ensemble de l'image à la fois.

<sup>2</sup> Le blobs (binary large object) est un type de donnée permettant le stockage de données binaires ( image, son ou video) dans le champ d'une table d'une base de données.

### II.2.3.2. Ajustement de ligne de base

Une fois que les lignes de texte ont été trouvées, les lignes de base sont ajustées avec plus de précision à l'aide d'une spline quadratique<sup>3</sup>. Tesseract permet de gérer des pages avec des lignes de base courbes [5]. Les lignes de base sont ajustées en séparant les blobs en groupes avec un déplacement raisonnablement continu pour la ligne de base droite initiale.

### II.2.3.3. Détection de hauteur fix et fragmentation

Tesseract teste les lignes de texte pour déterminer si elles ont une hauteur fixe. Là où il trouve du texte à une hauteur fixe, Tesseract fragmente les mots en caractères utilisant la hauteur.

La figure suivante montre un exemple typique d'un mot à hauteur fixe.

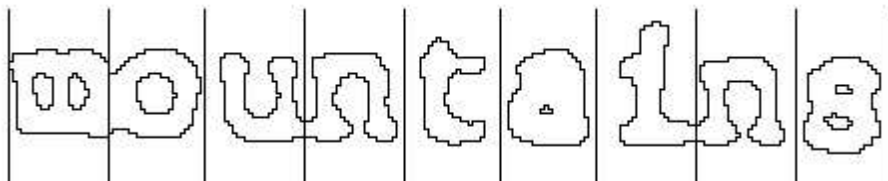


Figure 28 : Un mot coupé à hauteur fixe. [41]

### II.2.3.4. Recherche de mots proportionnels

L'espacement non vertical ou proportionnel du texte est une tâche hautement non triviale. Tesseract résout la plupart de ces problèmes en mesurant les écarts dans une plage verticale limitée entre la ligne de base et la ligne moyenne. Les espaces proches du seuil à ce stade sont rendus flous, de sorte qu'une décision finale puisse être prise après la reconnaissance des mots.

***fear of financialcollapse,***

Figure 29 : Un espacement difficile entre les mots.

---

<sup>3</sup> une spline est une fonction définie par morceaux par des polynômes.

## II.2.4. Reconnaissance des mots

Consiste à identifier comment un mot devrait être segmenté en caractères :

### II.2.4.1. Séparer les caractères collés

Tesseract tente d'améliorer le résultat de reconnaissance en coupant les blobs. Les points de découpage candidats sont trouvés à partir de sommets concaves d'une approximation polygonale du contour. Jusqu'à 3 paires de points de découpage peuvent être nécessaires pour séparer correctement les caractères joints du jeu ASCII.

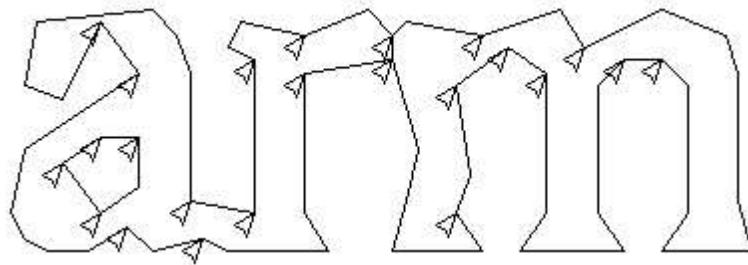


Figure 30 : Points candidats pour le découpage. [41]

### II.2.4.2. Associer les caractères brisés

Si le mot n'est toujours reconnu, il est donné à l'associateur. L'associateur effectue une recherche A\*<sup>4</sup> (best first) du graphe de segmentation des combinaisons possibles des blobs coupés au maximum en caractères candidats.

Grace à la recherche de segmentation A\*, la précision de Tesseract sur les caractères brisés était bien supérieure aux moteurs commerciaux de l'époque.

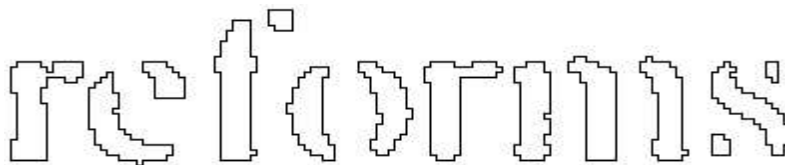


Figure 31 : Un mot facilement reconnu en utilisant Tesseract. [41]

---

<sup>4</sup> L'algorithme de recherche A\* : est un algorithme de recherche de chemin dans un graphe entre un nœud initial et un nœud final tous deux donnés.

## II.2.5. Classification des caractères

### II.2.5.1. Caractéristiques

Les caractéristiques topologiques tel que la police et de la taille ne suffise pas pour classifier un caractère, ces fonctionnalités ne résistent pas aux problèmes rencontrés dans les images réelles tellement varier. La solution révolutionnaire est l'idée que les caractéristiques du caractère inconnu ne doivent pas nécessairement être identiques à celles des données d'apprentissage. Pendant l'entraînement, les segments d'une approximation polygonale [41] sont utilisés pour les caractéristiques, mais, en reconnaissance, les entités de petite longueur fixe (en unités normalisées) sont extraites du contour et comparées à plusieurs par rapport aux prototypes en cluster de les données d'entraînement.

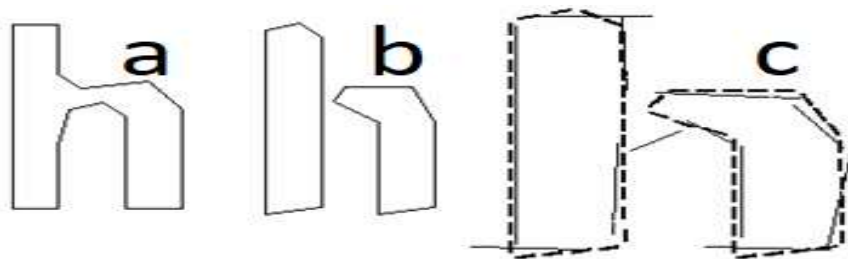


Figure 32 : a) "h" sain, b) "h" cassé, c) des caractéristiques adaptées aux prototypes. [41]

Sur la **figure 32**, les lignes courtes et épaisses représentent les caractéristiques extraites du caractère inconnu. Les lignes fines et plus longues représentent les segments en grappes de l'approximation polygonale utilisés en tant que prototypes. Un prototype reliant les deux pièces est totalement inégalé.

### II.2.5.2. Classification

Un sérateur de classe crée une liste de classes de caractères qui pourrait correspondre au caractère traité. Chaque caractéristique va chercher dans une table de correspondance quantifiée, un vecteur de bits de classes auquel elle pourrait correspondre, et les vecteurs de bits sont additionnés pour toutes les entités. Les classes avec les nombres les plus élevés deviennent la liste restreinte pour l'étape suivante.

### II.2.5.3. Données d'entraînement

Étant donné que le classificateur est capable de reconnaître facilement les caractères endommagés, il est bien entraîné pour cela, il a été formé pour plus de 60mille échantillons d'apprentissage.

### II.2.5.4. Analyse linguistique

Tesseract contient relativement peu d'analyses linguistiques. Chaque fois que le module de reconnaissance de mots envisage une nouvelle segmentation, le module choisit la meilleure chaîne de mots disponible dans chacune des catégories suivantes : Mots les plus fréquents, Mots du dictionnaire, Mots numériques les plus importants, Mots majuscules les plus importants, Mots minuscules les plus importants (avec majuscules initiales en option), Mots de choix du classificateur les plus importants. [41]

Résumé du principe de fonctionnement :

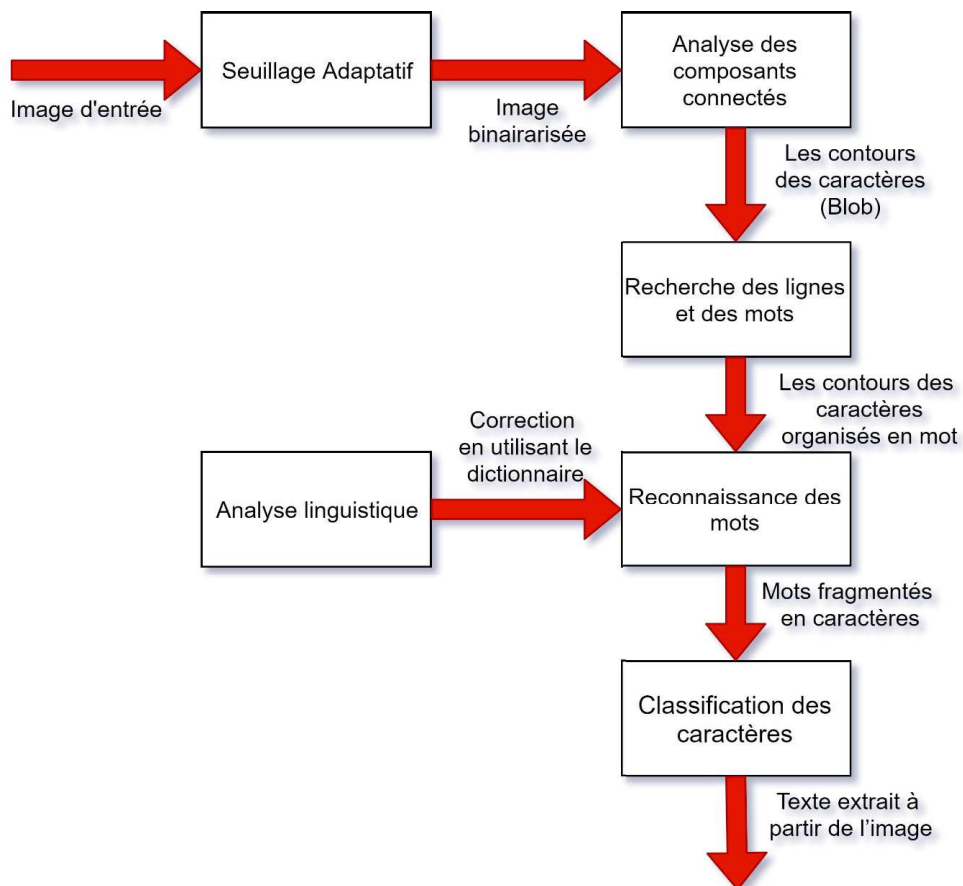


Figure 33 : schéma résumant le Mode fonctionnement de Tesseract.

### **III. Systèmes Embarqués**

#### **III.1.Introduction**

Les systèmes embarqués prennent une place de plus en plus importante dans notre vie, c'est souvent utilisé par le grand public dans la vie de tous les jours, ils ont la capacité de transformer tous les objets du monde physique du plus petit au plus grand, du plus simple au plus complexe en objets numériques, intelligents, autonomes et communicants. Par exemple, dans les smartphones (tel que Android et iOS), les systèmes de surveillance, le pilotage automatique des avions et des voitures...etc.

Comme notre travail consiste à concevoir une voiture autonome, on aura besoin d'un composant indispensable pour sa réalisation, la Raspberry Pi. On verra dans ce qui suit quelques points sur les systèmes embarqué.

#### **III.2. Définition**

Un système embarqué peut être défini comme un système électronique et informatique autonome dédié à une tâche bien précise et répondant souvent à des contraintes temps réel. Ses ressources disponibles sont généralement limitées, elles sont d'ordre spatial (taille limitée) et énergétique (consommation restreinte). [43]

Les systèmes embarqués sont désormais utilisés dans des applications diverses tel que le transport (avionique, espace, automobile, ferroviaire) dans les appareils électriques et électroniques (appareils photos, jouets, télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, etc.

L'attrait principal des systèmes embarqués vient du fait qu'ils permettent d'implémenter à faible coût des fonctions complexes dont la réalisation était inimaginable il y'a quelques années seulement. Aujourd'hui, la tendance générale est d'utiliser les systèmes embarqués pour incorporer des fonctionnalités complexes, précédemment considérées comme exotiques dans les produits de tous les jours.

### III.3. Caractéristiques spécifiques d'un système embarqué

Les principales caractéristiques d'un système embarqué sont les suivantes [44]:

- **Fonctionnement en temps réel :**

- Les applications embarquées doivent répondre rapidement aux évènements internes ou externes.
- Nécessaire dans les applications de système de contrôles.
- Le résultat peut être néfaste si le système ne réagit pas à l'immédiat à un évènement du système

- **Environnement :**

Un système embarqué est soumis à de nombreuses contraintes d'environnement, Il doit s'adapter et fonctionner avec eux. On peut citer :

- Température
- Vibrations
- Chocs
- Variations d'alimentation
- Corrosion
- Eau
- Feu
- Radiation
- etc.

- **Coût :**

Il est évident qu'un système aussi performant et moins couteux par rapport à un autre système sera considéré comme une solution plus attractive. Plusieurs aspects peuvent influencer le coût d'un système sur le marché :

- Surface du silicium
- Rendement de fabrication
- Packaging
- Effort de conception

- Coût de la programmation
- Temps de mise sur le marché...

- **Capacité mémoire adaptée :**

La mémoire est une ressource limitée dans un grand nombre de systèmes embarqués, et par conséquent une bonne utilisation de la ressource mémoire est cruciale pour la gestion de ces systèmes.

- **Consommation énergétique :**

Celle-ci est définie par la puissance consommée par le matériel au cours de l'exécution. Elle est liée à la surface occupée par l'application sur le circuit. Une consommation excessive augmente le prix de revient du système embarqué car il faut alors des batteries de forte capacité.

- **Fonctionnement critique pour la sécurité des personnes :**

- Le système doit toujours fonctionner correctement.
- Sûreté à faible coût avec une redondance minimale.
- Sûreté de fonctionnement du logiciel.
- Ses communications sont confidentielles et authentifiées.

### III.4. Domaine d'application des systèmes embarqués

Les systèmes embarqués sont introduits dans divers domaines, à savoir [45] :

*Tableau 3 : les divers domaines d'utilisation des systèmes embarqués*

<b>Le domaine grand public</b>	Smart phone Console de jeux Appareil photos Lecteur audio
<b>Les moyens de transport</b>	Ordinateur de bord GPS Système de navigation

<b>Les équipements médicaux</b>	Imagerie (rayon X, ultra-sons, IRM) Monitoring Lasers Chirurgie
<b>Les équipements de télécommunication</b>	Station mobile Routeur Gateway Satellite
<b>Les équipements industriels</b>	Productions automatisées Systèmes de commande d'énergie Équipements de stockage.
<b>Les équipements de bureautiques</b>	Répondeurs Copieurs Imprimantes.
<b>Les équipements de bâtiment</b>	Ascenseurs Système de surveillance Contrôle d'accès Systèmes d'éclairage

#### IV. Logiciel

##### IV.1. Python

Python est un langage de script de haut niveau, structuré et open source, multi-paradigme et multi-usage. Il est Développé à l'origine par Guido van Rossum en 1989. [45]

Le langage Python est un langage de programmation interprété, ce qui veut dire que les instructions qu'on lui envoie sont « transcrites » en langage machine au fur et à mesure de leur lecture. D'autres langages (comme le C / C++) sont appelés « langages compilés » car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine. On appelle cette étape la « compilation ». À chaque modification du code, il faut rappeler une étape de

compilation. Les avantages d'un langage interprété sont la simplicité (on ne passe pas par une étape de compilation avant d'exécuter son programme) et la portabilité.

Python est placé sous une licence libre proche de la licence BSD7 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et 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. [46]



Figure 34 : logo Python [73]

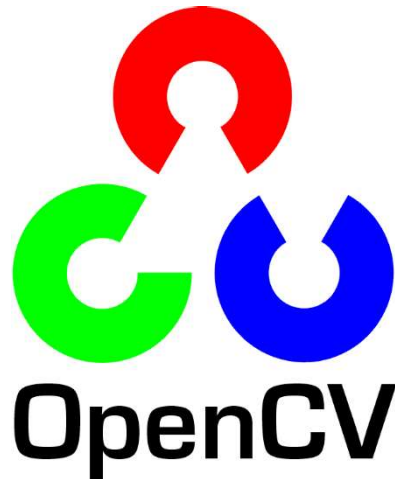
#### IV.2. Python IDLE

Autrement dit environnement de développement et d'apprentissage intégré) est un environnement de développement intégré (IDE) pour Python. Le programme d'installation Python pour Windows contient le module IDLE par défaut. IDLE n'est pas disponible par défaut dans les distributions Python pour Linux. Il doit être installé. IDLE fournit un éditeur de texte complet pour créer des scripts Python incluant des fonctionnalités telles que la coloration syntaxique, la complétion automatique et l'indentation intelligente. Il possède également un débogueur avec des fonctionnalités de pas à pas et de points d'arrêt. [47]

#### IV.3. OpenCV

OpenCV (Open-source Computer Vision library) est la librairie de référence pour le traitement d'image et d'apprentissage automatique. OpenCV a été conçu pour fournir

une infrastructure commune aux applications de vision par ordinateur et pour accélérer l'utilisation de la perception de la machine dans les produits commerciaux. OpenCV étant un produit sous licence BSD, il est facile pour les entreprises d'utiliser et de modifier le code.



*Figure 35 : logo OpenCV [74]*

La bibliothèque contient plus de 2500 algorithmes optimisés, qui incluent un ensemble complet d'algorithmes classiques et avancés de vision par ordinateur et d'apprentissage automatique. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, reconnaissance des chiffres et des caractères, identifier des objets, classer les actions humaines dans des vidéos, suivre des objets en mouvement. OpenCV compte plus de 47 000 utilisateurs et le nombre de téléchargements estimé à plus de 18 millions. La bibliothèque est largement utilisée dans les entreprises, les groupes de recherche et les organismes gouvernementaux.

Aux côtés de sociétés bien établies telles que Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda et Toyota qui utilisent la bibliothèque, de nombreuses startups telles qu'Applied Minds, VideoSurf et Zeitera utilisent énormément OpenCV. OpenCV associe des images de vues de rue, détecte des intrusions dans une vidéosurveillance, surveille des équipements, aide les robots à naviguer et ramasser des objets, détecte les accidents de noyade dans une piscine, exécute des œuvres d'art interactives en Espagne et à New York, inspectant le monde au Japon.

Il possède des interfaces C++, Python, Java et MATLAB et prend en charge Windows, Linux, Android et Mac OS. OpenCV s'appuie principalement sur les applications de vision en temps réel. Des interfaces CUDA et OpenCL complètes sont activement développées en ce moment. Il existe plus de 500 algorithmes et environ 10 fois plus de fonctions qui composent ou supportent ces algorithmes. OpenCV est écrit nativement en C++ et possède une interface basée sur un modèle qui fonctionne de manière transparente avec les conteneurs STL. [27]

OpenCV a une structure modulaire, ce qui signifie que le package comprend plusieurs bibliothèques partagées ou statiques. Les Principaux modules accessibles au travers de son API.

- **Core** : un module compact définissant les structures de données de base, y compris la matrice multidimensionnelle dense Mat et les fonctions de base utilisées par tous les autres modules.

- **Imgproc** : un module de traitement d'image comprenant un filtrage d'image linéaire et non linéaire, des transformations d'image géométriques (redimensionnement, affinage et déformation en perspective, remappage générique basé sur une table), une conversion d'espace couleur, des histogrammes, etc.

- **Vidéo** : un module d'analyse vidéo qui inclut des algorithmes d'estimation de mouvement, de soustraction d'arrière-plan et de suivi d'objets.

- **Calib3d** : algorithmes géométriques de base à vues multiples, calibrage de caméras simples et stéréoscopiques, estimation de la pose d'objets, algorithmes de correspondance stéréoscopique et éléments de reconstruction 3D.

- **Features2d** : détecteurs, descripteurs et appariements de descripteurs d'éléments saillants.

- **Objdetect** : détection d'objets et d'instances des classes prédéfinies (par exemple, les visages, les yeux, les tasses, les personnes, les voitures, etc.).

- **Highgui** : une interface facile à utiliser pour des fonctionnalités d'interface utilisateur simples.
- **Videoio** : une interface facile à utiliser pour la capture vidéo et les codecs vidéo.
- **GPU** : algorithmes accélérés par le GPU de différents modules OpenCV.

Certains autres modules auxiliaires, tels que les wrappers de test FLANN et Google, les liaisons Python, etc.

#### IV.4. Base de données

Une base de données abrégé en BDD, est une collection d'informations organisées et structurées reliées entre elles, afin d'être facilement consultables, gérables et mises à jour. Au sein d'une BDD, les données sont organisées en lignes, colonnes et tableaux et stockées sur supports numériques centralisés ou distribués. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées à l'aide d'un logiciel informatique. Chaque fois que de nouvelles informations sont ajoutées ou supprimées, les données sont mises à jour. [49]

#### Système de Gestion de Base de Données

Un SGBD est un logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance d'une base de données. Il est l'unique interface entre les informaticiens et les données (définition des schémas, programmation des applications), ainsi qu'entre les utilisateurs et les données (consultation et mise à jour). Parmi les SGBD les plus connus il est possible de citer [50] :

- **MySQL** est un SGBD relationnel libre (licence GPL et commerciale), simple d'accès et très utilisé pour la réalisation de sites Web dynamiques. Depuis la version 4 MySQL implémente la plupart des fonctions attendues d'un SGBD relationnel.
- **Oracle** est un SGBD relationnel (et relationnel-objet dans ses dernières versions) très reconnu pour les applications professionnelles.
- **PostgreSQL** est un SGBD relationnel et relationnel-objet très puissant qui offre une alternative open source aux solutions commerciales comme Oracle ou IBM.

- **Access** est un SGBD relationnel Microsoft, qui offre une interface graphique permettant de concevoir rapidement des applications de petite envergure ou de réaliser des prototypes.

#### **IV.5. Json**

JSON est l'acronyme de **JavaScript Object Notation**. Il s'agit d'un format de fichier open-standard, léger pour l'échange de données, et indépendant des langages qui l'utilisent, il permet de stocker les données de manière organisée et lisible par un humain tout en y facilitant l'accès. Les données sont présentées dans un format textuel constitué de paires « key : value ».

L'usage de JSON offre plusieurs avantages :

- Cette notation occupe moins de place que l'XML, réduisant ainsi la bande passante utilisée.
- Étant beaucoup plus simple dans sa structure, les opérations de lecture/écriture sur du JSON sont également beaucoup plus rapides.

#### **IV.6. Multiprocessing**

Dans un système à un seul processeur, un seul processus est exécuté à la fois. Le multitraitement consiste à utiliser deux processeurs ou plus au sein d'un même système d'ordinateur. Le terme fait également référence à la capacité d'un système à prendre en charge plusieurs processeurs au sein d'un même système informatique. Plusieurs processus peuvent être exécutés à la fois. Ces processeurs multiples partagent le bus informatique, parfois l'horloge, la mémoire et les périphériques. [51]

#### **IV.7. Multithreading**

Un thread est une unité d'utilisation de base du processeur. Le multi-threading est un modèle d'exécution qui permet à un processus unique de disposer de plusieurs segments de code (par exemple, des threads) s'exécutant simultanément dans le "contexte" de ce processus. Par exemple, le lecteur multimédia VLC, où un fil est utilisé pour ouvrir le lecteur multimédia VLC, un fil pour lire une chanson particulière et un autre pour ajouter de nouvelles chansons à la liste de lecture.

Le multi-threading est la capacité d'un processus à gérer son utilisation par plus d'un utilisateur à la fois et à gérer plusieurs demandes du même utilisateur sans avoir à avoir plusieurs copies du programme. [51]

### Différence entre Multiprocessing et Multithreading

Le module de threading utilise des threads, le module de multiprocessing utilise des processus. La différence est que les threads s'exécutent dans le même espace mémoire, alors que les processus ont une mémoire séparée. Cela rend un peu plus difficile le partage d'objets entre des processus multitraitement. Étant donné que les threads utilisent la même mémoire, des précautions doivent être prises ou deux threads écrivent dans la même mémoire en même temps.

#### IV.8. Socket

La notion de sockets a été introduite dans les distributions de Berkeley<sup>5</sup> (un fameux système de type UNIX, dont beaucoup de distributions actuelles utilisent des morceaux de code).

C'est la raison pour laquelle on parle parfois de sockets BSD (Berkeley Software Distribution).

Il s'agit d'un modèle permettant la communication inter processus (IPC - Inter Process Communication) afin de permettre à divers processus de communiquer aussi bien sur une même machine qu'à travers un réseau TCP/IP.

La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication :

- **Le mode connecté** (comparable à une communication téléphonique), utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.

---

<sup>5</sup> La Berkeley Software Distribution ou BSD est un système d'exploitation dérivé d'Unix et originaire de l'université de Californie à Berkeley.

- **Le mode non connecté** (analogue à une communication par courrier), utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné.

## **IV.9. Module python utilisés**

### **IV.9.1. Cv2**

Il s'agit de la bibliothèque OpenCV et contient les fonctions de traitement des images.

### **IV.9.2. Pickle**

Est le module qui permet de sauvegarder dans un fichier, au format binaire, n'importe quel objet Python. En clair, dans un script Python, vous avez besoin de sauvegarder, temporairement ou même de façon plus pérenne, le contenu d'un objet Python comme une liste, un dictionnaire, un tuple etc. Au lieu d'utiliser une base de données ou un simple fichier texte le module pickle est fait pour ça car il permet de stocker et de restaurer un objet Python tel quel sans aucune manipulation supplémentaire. Il fonctionne comme le module json mais n'est pas limité à un seul format d'objet. [52]

### **IV.9.3. Numpy**

Est une bibliothèque pour le langage de programmation Python, qui prend en charge de grands tableaux multidimensionnels. La représentation d'images sous forme de tableaux NumPy est très économe en ressources informatiques, c'est pour cela, de nombreuses autres bibliothèques de traitement d'images et d'apprentissage automatique utilisent également les représentations de tableaux NumPy. De plus, en utilisant les fonctions mathématiques intégrées de haut niveau de NumPy, nous pouvons rapidement et facilement effectuer une analyse numérique sur une image. [53]

### **IV.9.4. Pytesseract**

Python-tesseract ou simplement pytesseract est le module Python qui permet d'utiliser l'OCR avec Python. pytesseract est une bibliothèque qui encapsule le moteur Google Tesseract-OCR. Il peut lire tous les types d'image pris en charge par la

bibliothèque d'imagerie Python, y compris jpeg, png, gif, bmp, tiff et autres, alors que tesseract-ocr par défaut ne prend en charge que tiff et bmp. De plus, s'il est utilisé en tant que script, Pytesseract imprimera le texte reconnu au lieu de l'écrire dans un fichier. [54]

#### **IV.9.5.Struct**

Ce module effectue des conversions entre les valeurs Python et les structures C représentées sous forme d'objets octets Python. Cela peut être utilisé pour gérer des données binaires stockées dans des fichiers ou à partir de connexions réseau, entre autres sources. Il utilise des chaînes de format en tant que descriptions compactes de la disposition des structures C et de la conversion prévue en / à partir de valeurs Python.

#### **IV.9.6.Math**

Ce module donne accès aux fonctions mathématiques tel définies par le standard C, tel que les fonctions trigonométriques, les fonctions de représentation, les fonctions logarithmiques, les fonctions de conversion d'angle, etc... [55]

#### **IV.9.7.Concurrent.futures**

Le module `concurrent.futures` est un module qui fournit une interface de haut niveau pour l'exécution des tâches de manière asynchrone. L'exécution asynchrone peut être effectuée avec des threads, à l'aide de *ThreadPoolExecutor*, ou des processus distincts, à l'aide de *ProcessPoolExecutor*. Les deux implémentent la même interface, définie par la classe abstraite `Executor`. [56]

#### **IV.9.8.ThreadPoolExecutor**

`ThreadPoolExecutor` fournit une abstraction simple sur la création de plusieurs threads et l'utilisation de ces threads pour effectuer des tâches de manière simultanée. L'ajout de threads à votre application peut permettre d'améliorer considérablement la vitesse du programme exécuter. [56]

#### **IV.9.9.OS.fork**

Le module OS de Python permet d'utiliser les fonctionnalités dépendantes du système d'exploitation. Le fork est l'un des aspects les plus importants d'Unix et de Linux. Lorsqu'un processus se divise, il crée une copie de lui-même. Plus généralement, un fork dans un environnement multiprocessing signifie qu'un processus d'exécution est dupliqué, créant un processus enfant à partir du processus parent. Ils sont identiques mais peuvent être distingués grâce au PID (Process IDentifier). [57]

Fork crée un espace d'adressage distinct pour l'enfant, ce qui permet au processus enfant de s'exécuter en tant qu'instance indépendante, c'est-à-dire indépendante de son processus parent.

### **V. Matériel**

#### **V.1. Raspberry Pi**

##### **V.1.1. Introduction**

Le Raspberry Pi est un nano-ordinateur qui est capable d'effectuer un certain nombre d'applications embarquée. Il est devenu en quelques années de son apparition l'un des outils les plus populaires dans le domaine du développement. Les bricoleurs et les nouveaux programmeurs du monde entier utilisent le mini-ordinateur pour réaliser des projets très intéressants.

Ce petit ordinateur a été développé dans le but de faciliter le processus d'apprentissage informatique afin qu'un étudiant moyen puisse en tirer parti et anticiper ce que peut faire un ordinateur avancé. Dans ce qui suit, Nous allons traiter les microcontrôleurs de type Arduino, puis faire une comparaison entre la carte Arduino UNO et certaines versions de la carte Raspberry Pi. Nous allons ensuite nous attarder dans l'étude de cette dernière.

##### **V.1.2. Les microcontrôleurs**

Un microcontrôleur (en notation abrégée  $\mu\text{c}$ , ou  $\text{uc}$ ) est un circuit capable d'exécuter un programme. Il rassemble les éléments essentiels d'un ordinateur et

possède des circuits d'interface intégrés avec le monde extérieur. Les microcontrôleurs se caractérisent par un haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible et un coût réduit. [58]

Il existe beaucoup de microcontrôleurs différents sur le marché, mais les Arduino et les Raspberry sont les plus populaires car ils sont utilisés par une grande communauté et ils sont simples à prendre en main.

### **V.1.3. Arduino**

L'Arduino est une carte programmable simple sur laquelle on branche des composants électroniques, il possède un grand nombre d'entrées/sorties ce qui permet de brancher beaucoup de composants (capteurs, leds ...).

Il est donc principalement utilisé pour réaliser des interfaces électroniques avec des capteurs et des équipements. Une fois que le programme est chargé, l'Arduino l'exécute en boucle de manière autonome.

Les Arduino ont une mémoire de petite taille, une faible puissance de calcul, une faible consommation électrique (peut fonctionner facilement à partir de batteries) et un bon courant de sortie pour alimenter des modules externes. [59]

### **V.1.4. Raspberry Pi**

Le Raspberry Pi est un mini-ordinateur monocarte qui possède son propre système d'exploitation. On peut y brancher de l'audio, de la vidéo, un câble Ethernet, une carte SD, une clé USB, un câble HDMI, un clavier, un écran... exactement comme pour un ordinateur. C'est en réalité un micro-processeur. Il est 40 fois plus rapide que l'Arduino, a une plus grande mémoire, consomme plus de courant et possède moins d'entrées/sorties. Il peut interpréter des programmes en plusieurs langage tel que le C++, Java, python et beaucoup d'autres (contrairement à l'Arduino qui n'accepte que le C). [59]

Il existe plus modèle de Raspberry Pi qui sont au fur et à mesure améliorés pour une création de nouvelles versions. On peut citer par exemple le modèle zero W, modèle zero

W, modèle 1 A, modèle 1 A+, modèle 1 B, modèle 1 B+, modèle 2 A... jusqu'à la plus récente à peine sortie des usines, la Raspberry Pi 4 B.

Ces deux périphériques (Arduino et Raspberry Pi) ne sont donc pas destinés aux mêmes types de projets. Les Arduino sont utilisés pour des projets simples et plus au moins facile à réaliser, alors que les Raspberry sont consacrés pour des projets bien plus compliqué, plus puissants et moins simples à mettre en œuvre.

### V.1.5. Un tableau comparatif

Ce tableau illustre quelques différences entre Arduino UNO et certains modèles de la Raspberry Pi.

Tableau 4 : comparaison entre les propriétés de l'Arduino UNO et quelques Modèles de Raspberry Pi

	Arduino UNO Rev3	Modèle 3 B	Modèle 3 A+	Modèle 3 B+	Modèle 4 B
Prix de lancement (\$US)	20	30	25	35	35/45/55
Date de lancement	Mai. 2018	Fév. 2016	Nov. 2018	Mar. 2018	Jui. 2019
CPU	16 MHz ATmega328 P	1,2 GHz Quadricœur ARM Cortex-A53	1,4 GHz Quadricœur ARM Cortex-A53	1,4 GHz Quadricœur ARM Cortex-A53	1,5 GHz Quadricœur ARM Cortex-A72
Mémoire vive	2Ko	1 Go	512 Mo	1 Go	1 Go / 2 Go / 4 Go
N <sup>br</sup> de ports USB	1 (2.0)	4 (2.0)	1 (2.0)	4 (2.0)	2(2.0) - 2(3.0)
Sorties vidéo	Non	HDMI et Composite (Via un connecteur Jack)			2 Micro-HDMI
Carte / Connectique réseau	Non	10/100 Ethernet, Wifi 802.11n, Bluetooth 4.1	Wifi 802.11ac, Bluetooth 4.2	10/100 Ethernet(Max. 300Mbps), Wifi 802.11ac, Bluetooth 4.2	10/100/1000 Ethernet, Wifi 802.11ac, Bluetooth 5.0 BLE
Dimensions (mm)	74 x 53 x 15	86 x 54 x 17	65 x 56 x 13	86 x 54 x 17	85 x 56 x 16

Systèmes d'exploitation	Non	Debian GNU/Linux, Raspbian OS, Fedora, RISC OS, Arch Linux ARM1, FreeBSD, Plan 9, Kali Linux Snappy Ubuntu Core93, SolydX RPI94, Windows 10 IoT78
IDE	Arduino	Scratch, IDLE, toutes distributions de Linux
Capacité multitâche	Non	Oui

### V.1.6. Les Pins (brochage) de la Raspberry Pi

Un des grands points intéressants du Raspberry Pi est qu'il dispose d'un connecteur GPIO sur lequel on peut connecter du matériel externe (le model Pi b+ comporte 40 broches).

Le port **GPIO** (General Purpose Input/Output) du Raspberry Pi permet d'interagir avec le monde réel en communiquant avec des capteurs, des moteurs, et d'autres composants électroniques (sondes de température, LEDs, capteurs gyroscopiques, boussoles, GPS, etc.). Ces 40 broches peuvent être utilisées de manières différentes et se décompose en plusieurs types :

- **Alimentation (PWR)** : 3,3V ou 5V.
- **Ground (GND)** : masse 0V.
- **GPIO** : ce sont les port standard d'entrée/sortie et qui sont numérotés.
- **I<sup>2</sup>C** : Inter Integrated Circuit (Bus de circuit intégré) : permet la communication avec de multiples périphériques, un MCP23017 IO expander digital, un microprocesseur ATmega connecté à la Raspberry Pi, etc.
- **SPI** : Serial Peripheral Interface (Interface périphérique série) permet une communication entre un maître et un ou plusieurs esclaves.
- **UART** : Universal Asynchronous Receiver Transmitter (Emetteur récepteur asynchrone universel) c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série.

**PWM** : Pulse-width modulationc (Modul+ation de largeur d'impulsion). Les broches du port GPIO sont alimentées en 3,3V. Seules les deux broches d'arrivée de courant (en rouge) fournissent une tension de 5V.

L'I2C, le SPI, l'UART et le PWM permettent de communiquer avec des microcontrôleurs pour réaliser des opérations plus complexes telles que le contrôle de servos moteurs, la lecture de mesures prises par des capteurs, etc.

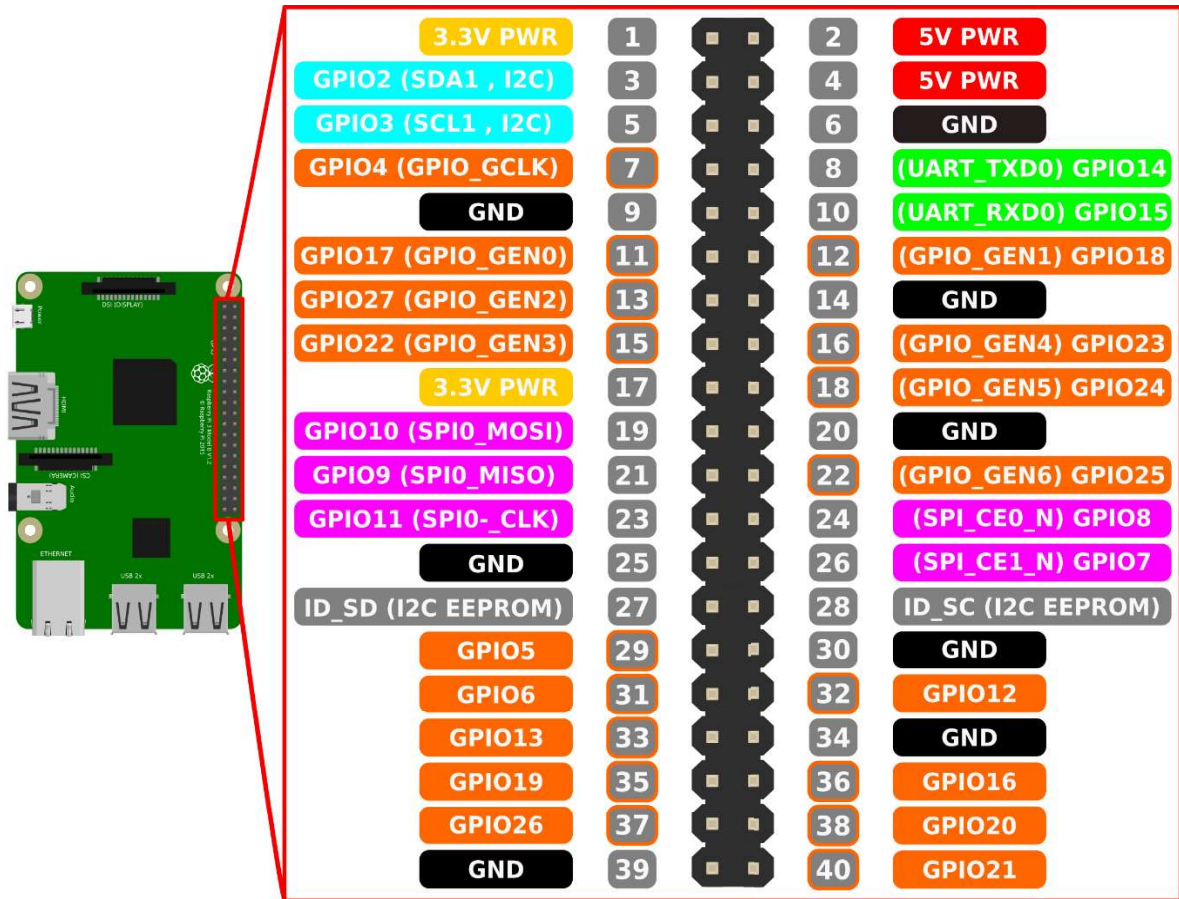


Figure 36: brochage de Raspberry Pi 3 B+ [68]

### V.1.7. Quelque exemple de projets nécessitant plutôt un Arduino ou un Raspberry

- Une caméra de sécurité connectée à internet – Raspberry.
- Une machine pour trier des skittles – Arduino.
- Un escalier-piano – Arduino.
- Un serveur web perso – Raspberry.
- Un cube de LED – Arduino.
- Un robot qui résout les rubiks cube – Raspberry.
- Un feu de circulation (rosuge orange rouge) – Arduino.

- Voiture autonome – Raspberry.

### V.1.8. Systèmes d'exploitation sur les Raspberry Pi

Les Raspberry Pi sont livrés par défaut vierges de tout système d'exploitation et de tout dispositif de stockage. C'est à l'utilisateur de choisir et d'installer le système d'exploitation qu'il veut ainsi que la carte de stockage de type microSD.

Il y a beaucoup de différents systèmes d'exploitation disponible pour la Raspberry Pi. Nous présentons quelques-uns des plus utilisés.

#### V.1.8.1. Raspbian

Raspbian est un système d'exploitation libre basé sur la distribution Linux Debian et optimisé pour le matériel de Raspberry Pi. Une première version du portage Debian a été publiée peu après la sortie de Raspberry Pi en juin 2012 et depuis lors, Raspbian est considéré comme le système d'exploitation officiel de Raspberry Pi. À cette époque, la distribution comprenait déjà plus de 35 000 progiciels, qui étaient mis à la disposition de l'utilisateur après l'installation.

Cependant, grâce à des développements continus, la taille du référentiel n'a cessé de croître et comprend maintenant même des programmes Office tels qu'un programme de traitement de texte ou un client de messagerie électronique. [60] [61]

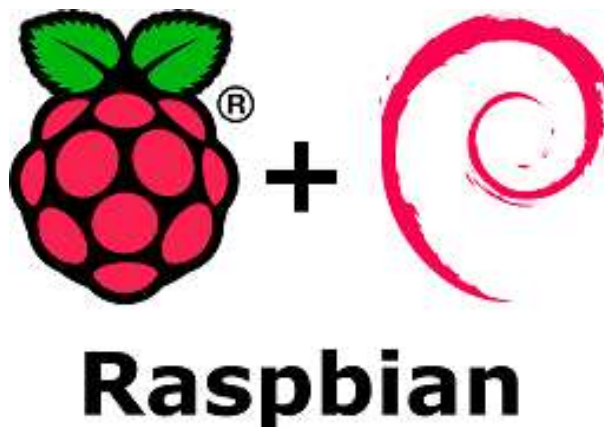


Figure 37: Logo du système d'exploitation Raspbian [69]

### V.1.8.2. Windows 10 IoT Core

Avec Windows 10 IoT Core, Microsoft a lancé son premier système d'exploitation pour les appareils IoT comme le Raspberry Pi (2 ou 3) en 2015. L'application s'adresse principalement aux développeurs et aux amateurs qui souhaitent mettre en réseau des objets du quotidien avec Internet ou bien créer de nouveaux objets en réseau. Windows 10 IoT Core s'appuie sur l'API interne « Universal Windows Platform » (UWP), qui

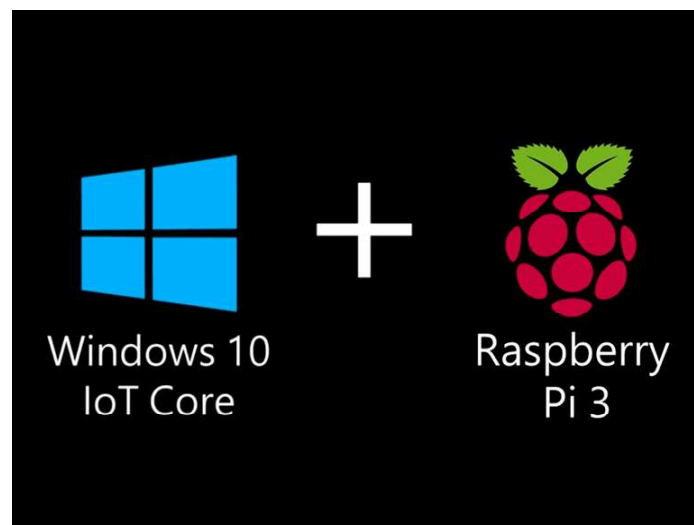


Figure 38: Logo du système d'exploitation Windows 10 IoT Core [71]

permet justement d'écrire des applications pour ses propres appareils. L'édition communautaire gratuite de

### V.1.8.3. Kali Linux

Un autre dérivé de Debian qui peut être utilisé dans son édition ARM comme distribution Raspberry Pi est Kali Linux. Le programme open source a été publié le 13 mars 2013 dans une première version. La distribution peut être utilisée comme un logiciel de système d'exploitation ordinaire, mais elle est principalement utilisée pour des tests de sécurité et des tests d'intrusion étendus des systèmes et réseaux informatiques. Pour cela, l'application contient plus de 600 outils : comme le scanner de ports libre Nmap, le Metasploit Framework (plateforme de test d'exploitation) ou le

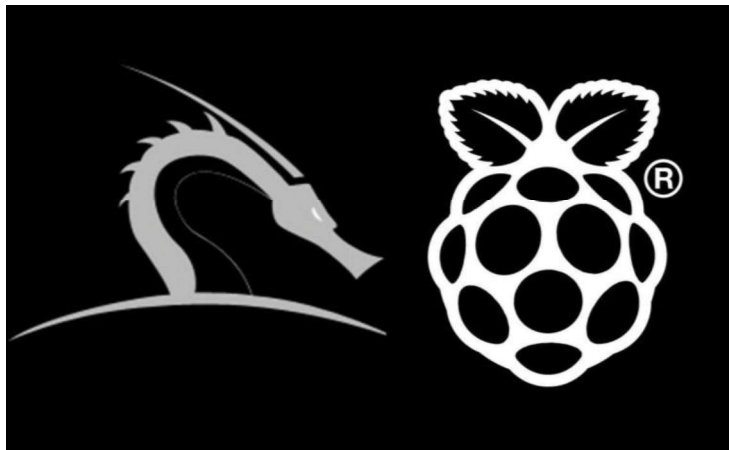


Figure 39 : Logo du système d'exploitation Kali Linux [63]

craqueur de mots de passe John the Ripper. Kali Linux est le successeur officiel de BackTrack, qui était basé sur la distribution Linux Ubuntu. [62]

## V.2.Camera

La caméra est un appareil de prise de vues destiné à enregistrer ou à transmettre des images ou un flux vidéo de haute définition, utilisée dans plusieurs domaine tels

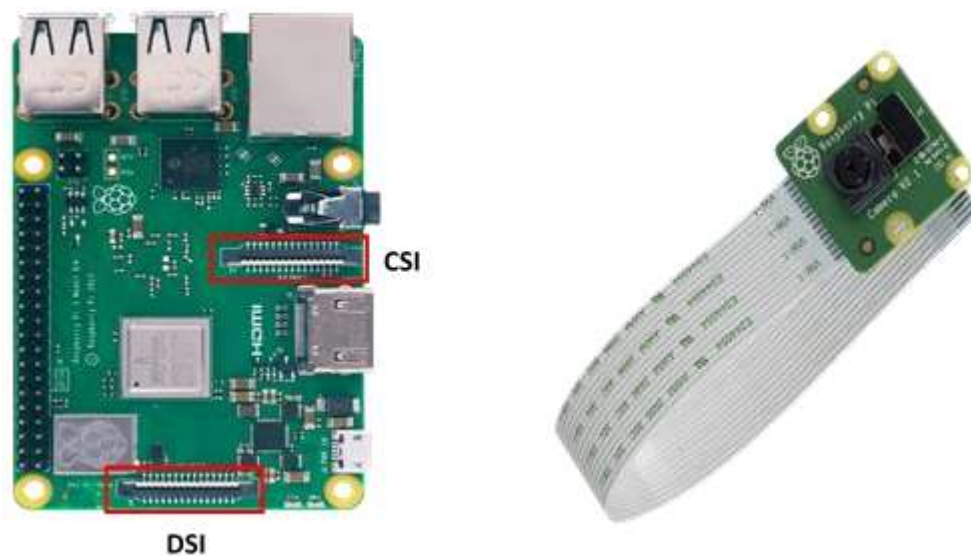


Figure 40 : Les ports DSI et CSI sur une raspberry Pi 3

que la recherche, la télésurveillance, l'imagerie industrielle et médicale, ou pour d'autres applications professionnelles ou domestiques. [64]

En plus de tous les composants que l'on peut relier aux GPIOs, le Raspberry possède deux connecteurs particuliers qui permettent de relier directement un écran (Display Serial Interface) et une caméra (Camera Serial Interface) au Raspberry Pi.

### V.3.Breadboard

La planche de montage expérimental (appelé en anglais breadboard et platine Labdec en France) est un outil essentiel pour l'électronique. Ce dispositif permet de réaliser le prototype d'un circuit électronique et de le tester. L'avantage de ce système est d'être totalement réutilisable, car il ne nécessite pas de soudure. De plus, il permet de brancher une grande variété de composants afin de réaliser des circuits électroniques et faire des tests très facilement et rapidement.

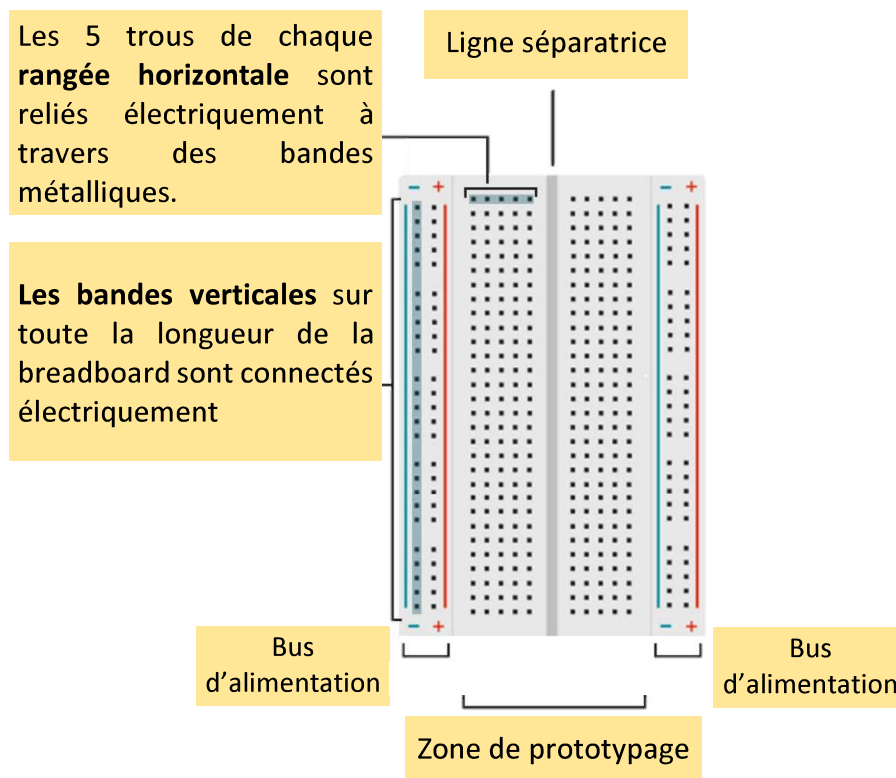


Figure 41 : Platine Labdec (Breadboard)

La planche est recouverte de trous dans lesquels on insère directement les broches des composants électroniques. [65]

#### V.4.Moteur

Un moteur à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. [66]

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique.

Composé de deux éléments un rotor et un stator :

- Le rotor : qui tourne et transmet le mouvement.
- Le stator : qui est alimenté par le courant électrique et qui est fixé au bâti.

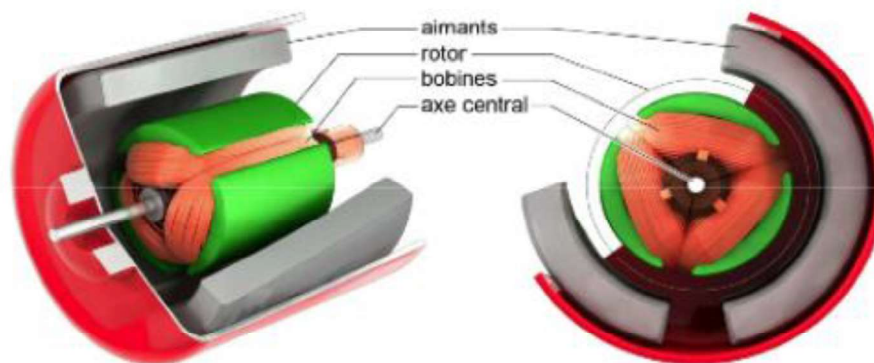


Figure 42 : une figure illustrant les composants d'un moteur à courant continu [66]

#### V.5.L298N

Le module L298N est un Double Pont-H destiné au contrôle de moteur (H-Bridge Motor Driver).

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il est conçu pour supporter des tensions plus élevées, des courants importants.

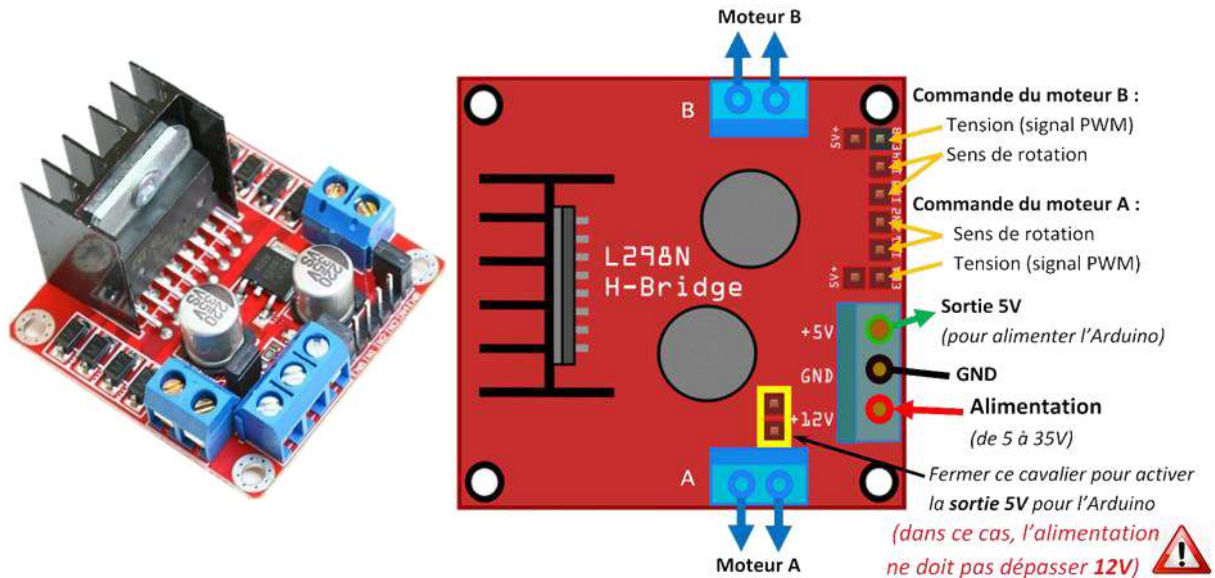


Figure 43: Double pont-H (L298n) [70]

Il peut piloter deux moteurs continus ou un moteur pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retours.

Tableau 5 : résumé du fonctionnement d'un L298n

	Arrêt (Moteur Libre)	Sens +	Sens -	Arrêt (Moteur Freiné)
IN1	LOW (0)	LOW (0)	HIGH (1)	HIGH (1)
IN2	LOW (0)	HIGH (1)	LOW (0)	HIGH (1)

## V.6. Ultrason (HC-SR04)

Un son est une vibration qui se propage sous forme d'onde. Les sons à fréquence très basse sont appelés « infrasons » alors que les sons à très haute fréquence sont appelés « ultrasons ».



Figure 44 : capteur ultrason [72]

Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur, puis la distance le séparant de la cible est calculé à base du temps écoulé entre l'émission du signal et la réception de l'écho.

Les ultrasons sont principalement utilisés, car ils sont inaudibles à l'oreille humaine et sont relativement précis sur de courtes distances. [67]

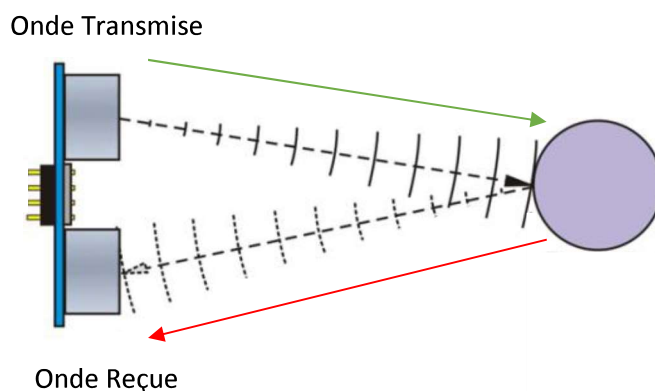


Figure 45 : fonctionnement du capteur ultrason [72]

## VI. Conclusion

Dans ce chapitre, nous avons présenté les différentes technologies sur lesquels on s'est appuyé pour concevoir notre projet. En premier lieu nous avons présenté l'OCR Tesseract ainsi que des généralités sur les systèmes embarqués puis, nous avons exposé les outils matérielles et logicielles utilisés pour sa réalisation.

Dans le prochain chapitre nous allons passer à la phase finale qui est la partie réalisation et tests.

# CHAPITRE IV

## IMPLÉMENTATION ET RÉALISATION

## CHAPITRE IV : Implémentation et Réalisation

### I. Introduction

Nous présentons dans ce chapitre les différentes étapes de réalisation de notre système et les détails d'implémentation de notre approche en utilisant les outils présentés dans le chapitre précédent.

### II. Fonctionnement de l'application

Dans ce qui suit, nous allons présenter les détails logiciels et matériels relatif à votre projet.

#### II.1. Software

Le programme principal se sépare en deux parties : client et serveur.

##### Partie client :

- S'exécute sur la Raspberry.
- Découpe le flux vidéo en frame.
- Réalise tous les traitements nécessaires pour l'extraction des plaques.
- Chaque plaque est envoyée vers le serveur afin de reconnaître les caractères présents.

##### Partie serveur :

- S'exécute sur une autre machine.
- Utilise le multiprocessing pour créer deux processus enfants afin de répartir les tâches pour but d'améliorer la vitesse du programme.
  - Le 1<sup>er</sup> Processus : s'occupe de la réception des images depuis le client et les range dans un vecteur « vect ». Ce vecteur représente une ressource partagée entre les deux processus créés.
  - Le 2<sup>ème</sup> Processus : son rôle est d'effectuer des traitements sur chaque élément (image de plaque) du vecteur « vect ». Pour pouvoir aboutir à un résultat rapide et en temps réel dans la reconnaissance qui est réalisée grâce à pytesseract, ce deuxième processus fait appel au multithreading

et va donc prendre en charge le traitement de plusieurs images simultanément.

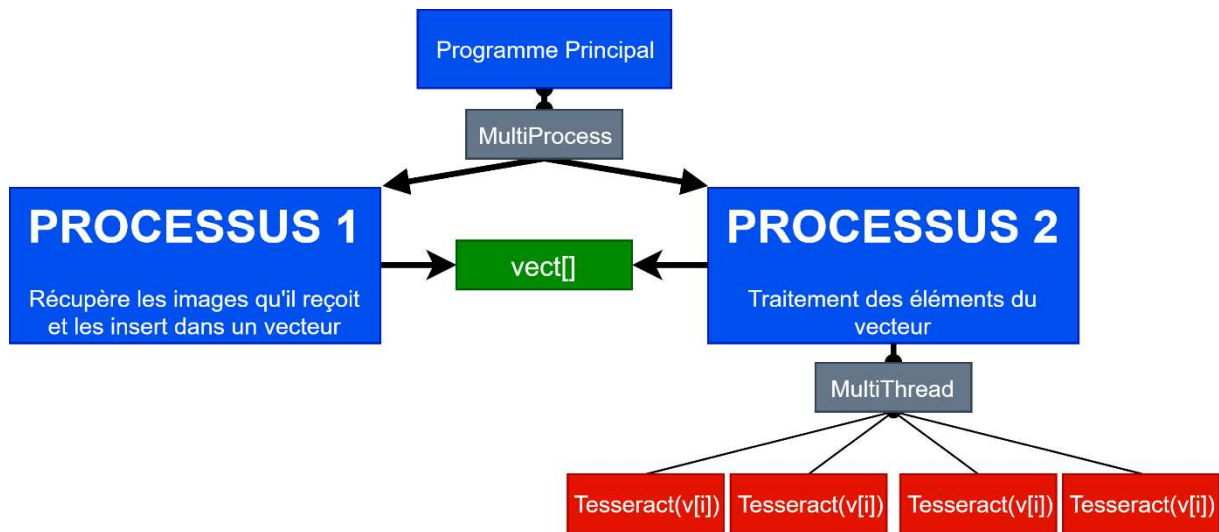


Figure 46 : schema resuming la decoposition des taches afin de reconnaitre les plaques d'immatriculations

### II.1.1. Reconnaissance de plaques

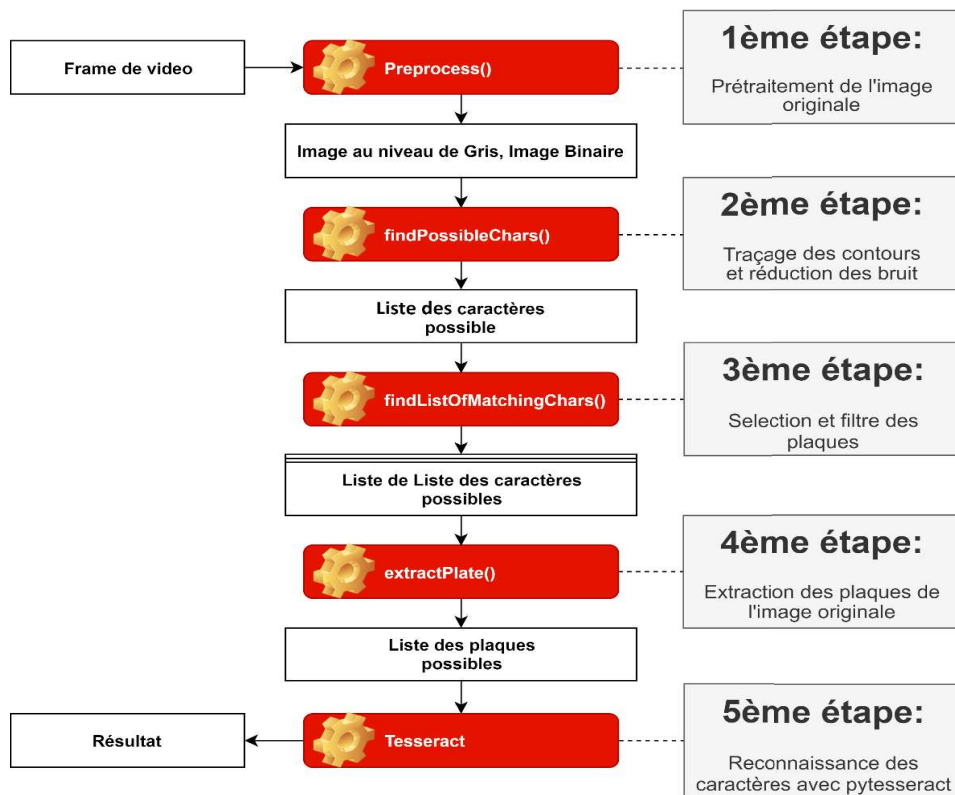


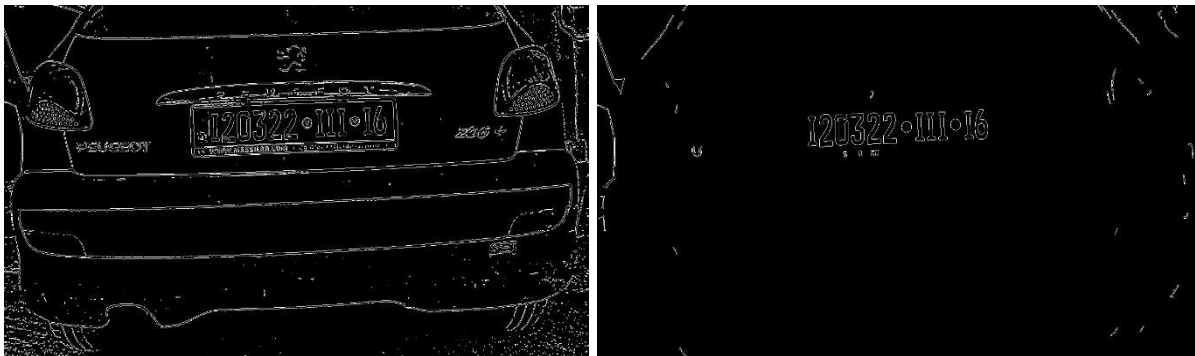
Figure 47 : Diagramme résumant les étapes de reconnaissance des plaques d'immatriculations

**Étape 1 :** Dans un premier lieu, la caméra capture un flux vidéo qui sera traité avec python et OpenCV, qui séparera la vidéo en plusieurs images. Chaque image devra passer par une étape de prétraitement, ou elle sera filtrée au niveau de gris puis binarisée.



*Figure 48 : resultat de l'image en noir et blanc et en binaire*

**Étape 2 :** Une fois l'image est en binaire, une fonction de détection de contour est appelée suivi par une fonction d'élimination de bruit qui se base sur l'aspect ratio des caractères. Cette étape permet de mettre en avant tous les contours ressemblants à des caractères, présents sur l'image.



*Figure 49 : detection de contours et elimination de bruits*

**Étape 3 :** Dans cette étape le but est de sélectionner les listes des caractères qui forment des structures semblables à des plaques d'immatriculations. En filtrant les contours déjà obtenu et en rassemblant ceux qui se trouvent dans la même zone qui suit différents paramètres d'espacement, d'orientation et de taille, entre les caractères,

on obtient alors toutes les régions candidates susceptible d'être des plaques d'immatriculation.

Sur la figure 49, chaque couleur représente une région candidate.



Figure 50 : detection des regions susceptible d'etre des plaques d'immatriculation

**Étape 4 :** On passe ensuite à l'extraction des plaques d'immatriculations. Cela signifie qu'on isole les régions les plaques de l'image originale, puis mise en noir et blanc pour faciliter leurs traitements avec Pytesseract.



Figure 51 : Isolation des plaques d'immatriculations

**Étape 5 :** toutes les plaques obtenues doivent alors passées par un système de reconnaissance de caractère, dans notre cas nous avons utilisé Tesseract, ou plus précisément la bibliothèque OCR de python « pytesseract ».

```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\the-s\Desktop\OpenCV_0\Memoire_modules\1- Licence Plate Recognition\Main.py
plaque numero = 12032211116
|
-----
>>>
```

Figure 52 : représentation des caractères détectés sur l'image en une chaîne numérique

Au final, le résultat est représenté par une chaîne numérique de caractères autrement dit le numéro de la plaque d'immatriculation du véhicule, et pour cela il doit impérativement être filtré pour ne répondre qu'avec des numéros de plaques correctes.



Figure 53 : représentation des caractères détectés sur l'image en une chaîne numérique

### II.1.2. Détection des marquages au sol des routes

Afin que notre véhicule soit capable de reconnaître le chemin à suivre sans sortir des routes, nous avons mis en point un système de détection des marquages sur les routes, pour cela nous avons tout d'abord filtré l'image pour obtenir une image binarisée. Une fois obtenue, on a utilisé un masque pour isoler la partie de la route de l'image originale qui se trouve dans la moitié basse de chaque image capturée.

Sur l'image résultante, on détecte toutes les lignes droites qui s'y trouvent, et pour cela on a utilisé une fonction d'OpenCV « HoughLinesP ».

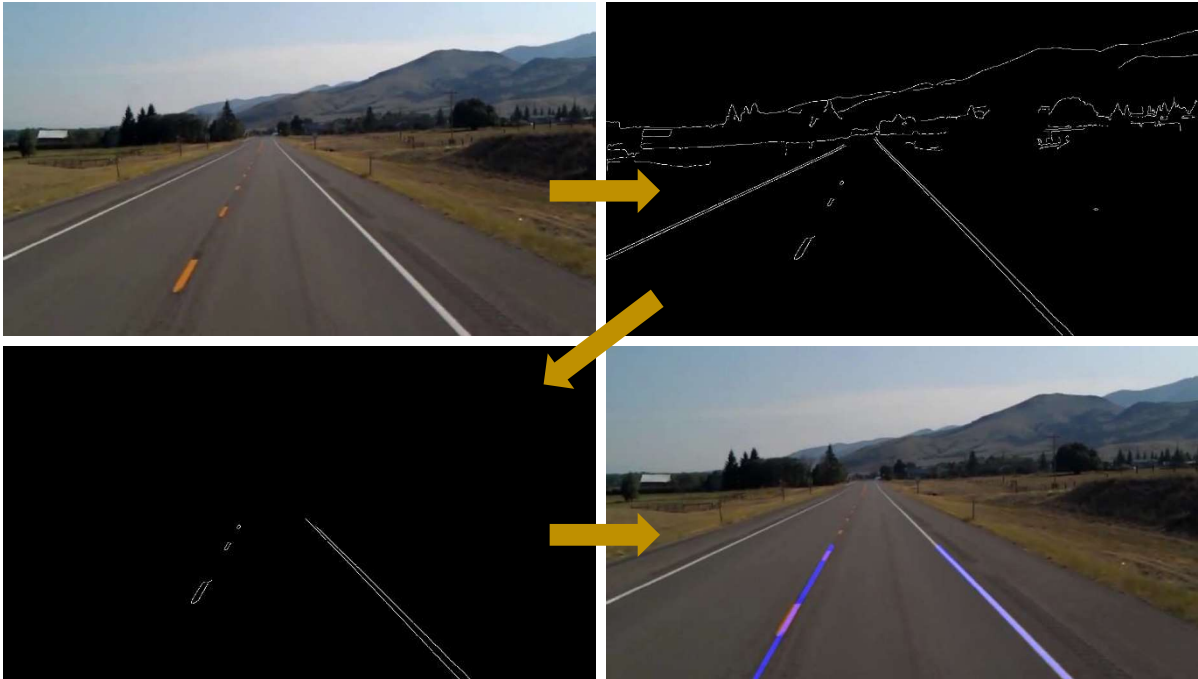


Figure 54 : Détection des marquages au sol des routes

### II.1.3. Détection des feux de signalisation

Pour détecter les feux de circulation qui sont normalement rouges ou verts dans notre cas, l'espace RVB n'est pas très approprié. RVB mélange les informations de couleur et d'intensité sur l'ensemble de ses canaux, Ce qui le rend sensible aux changements d'éclairage.

C'est pour cela que nous avons opter pour la conversion en un espaces colorimétriques qui séparent la chrominance ou les informations de couleur de la luminance. À part l'espace RVB, il y a plusieurs systèmes colorimétriques qui peuvent être utilisés, comme notamment l'espace HSV (Hue-Saturation-Value). Cet espace nous permet donc par simple modification sur les deux champs teinte et saturation d'extraire une plage de couleur de différente intensité.

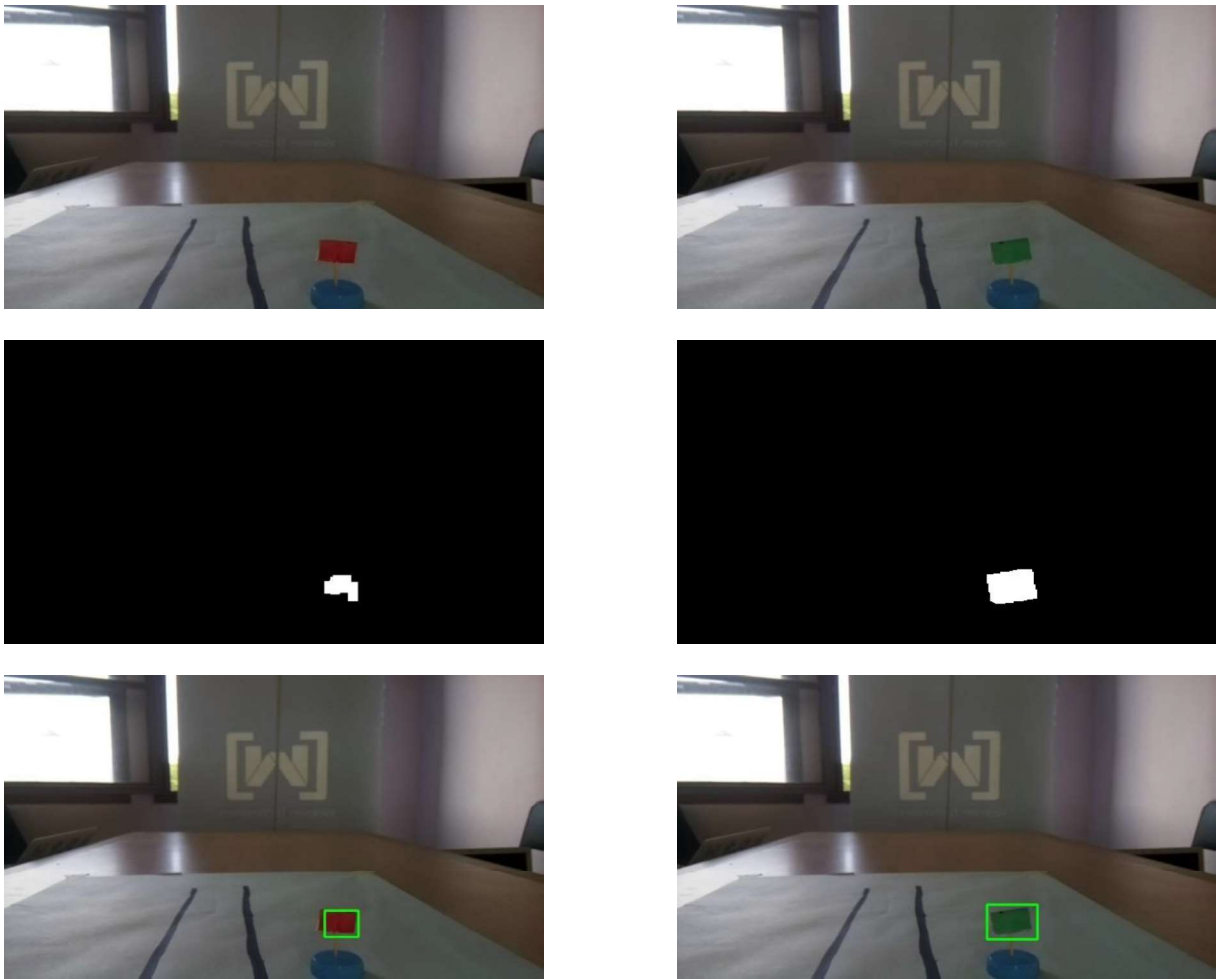


Figure 55 : détection des feux signalisation

## II.2. Hardware

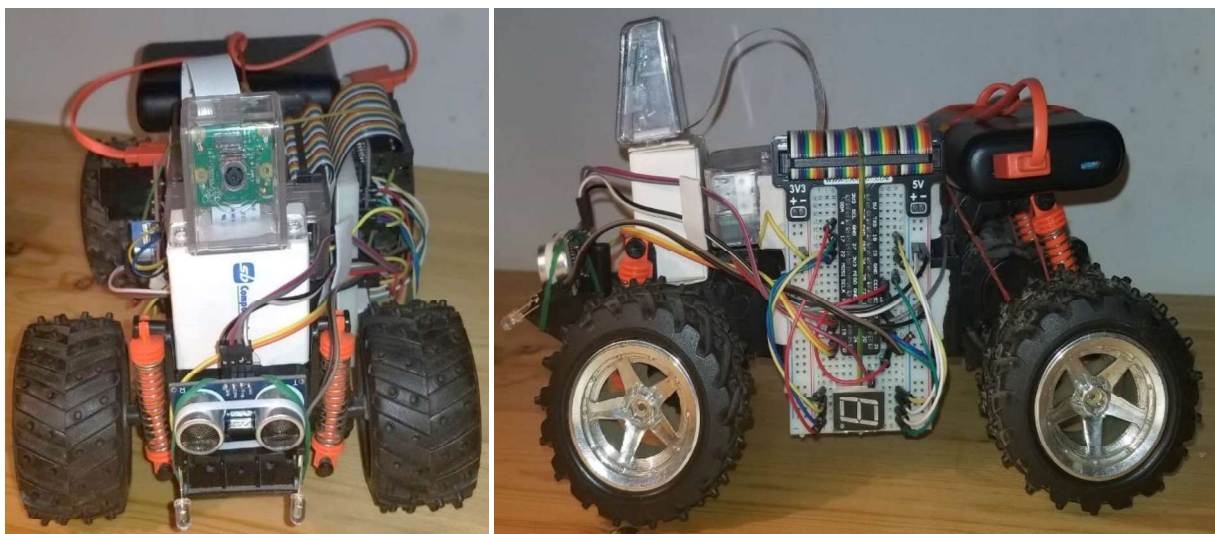


Figure 56 : le prototype du véhicule intelligent

La partie matérielle de notre projet se compose d'une Raspberry Pi 3 B+ qui représente le composant principale de la voiture, dont sont branchés les éléments suivants :

- Un ultrason (HC-SR04) qui va permettre d'identifier les obstacles se trouvant devant le véhicule dans un angle de 15°, et de connaître leurs distances à condition qu'ils se trouvent entre 2cm et 400 cm.
- Deux moteurs qui vont nous permettre de contrôler les 4 roues de la voiture. L'un d'eux se charge de faire tourner les deux roues arrière qui font avancer ou reculer le véhicule. Tandis que l'autre s'occupe des deux autres roues qui permettent de tourner vers la gauche ou vers la droite.
- La caméra permet une compréhension meilleure de l'environnement extérieur. Grâce au flux vidéo capturé, la Raspberry sera capable de reconnaître les plaques d'immatriculation et de détecter les marquages au sol dans les rues.
- Le Shield L298N quant à lui, permet de contrôler les deux moteurs à courant continu (CC).
- Les LEDs : dans un environnement nocturne les LEDs illumine l'itinéraire de la voiture ce qui permettra à la Raspberry et à la caméra de mieux fonctionner même dans la nuit.

### **II.2.1. Détection de distance et d'obstacles**

Notre système fait appel à un capteur ultrason de model HC-SR04 utilisent le principe de l'écho pour déterminer la distance entre le capteur et un obstacle se situant devant lui, qu'ils soient en mouvements ou immobiles.

Le capteur est placé en avant de la voiture et envoi des signaux, Le son est réfléchi par une surface et repart en direction du capteur, puis envoyer vers la Raspberry qui se chargera de calculer la distance. Une fois calculer le véhicule peut maintenant soit d'avancer s'il n'y a pas d'obstacles ou de réduire la vitesse s'ils sont proches, ou de s'arrêter sinon.

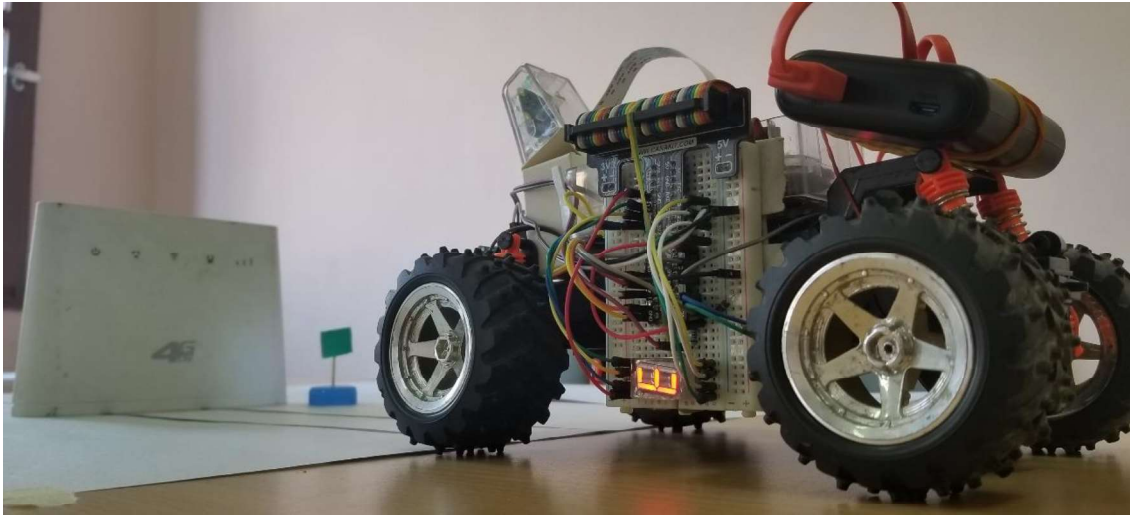


Figure 57 : obstacle detecte et affichage de la distance

Notre véhicule sera équipé d'un module camera contrôlé par la Raspberry, qui capturera son environnement, les images capturées seront ensuite traitées pour détecter les marquages au sol puis couplées aux données reçues de l'ultrason qui ont été récupérées, ce qui permettra à la Raspberry de comprendre l'environnement de la voiture pour prendre la bonne décision concernant son pilotage (avancer, tourner, freiner ou s'arrêter).

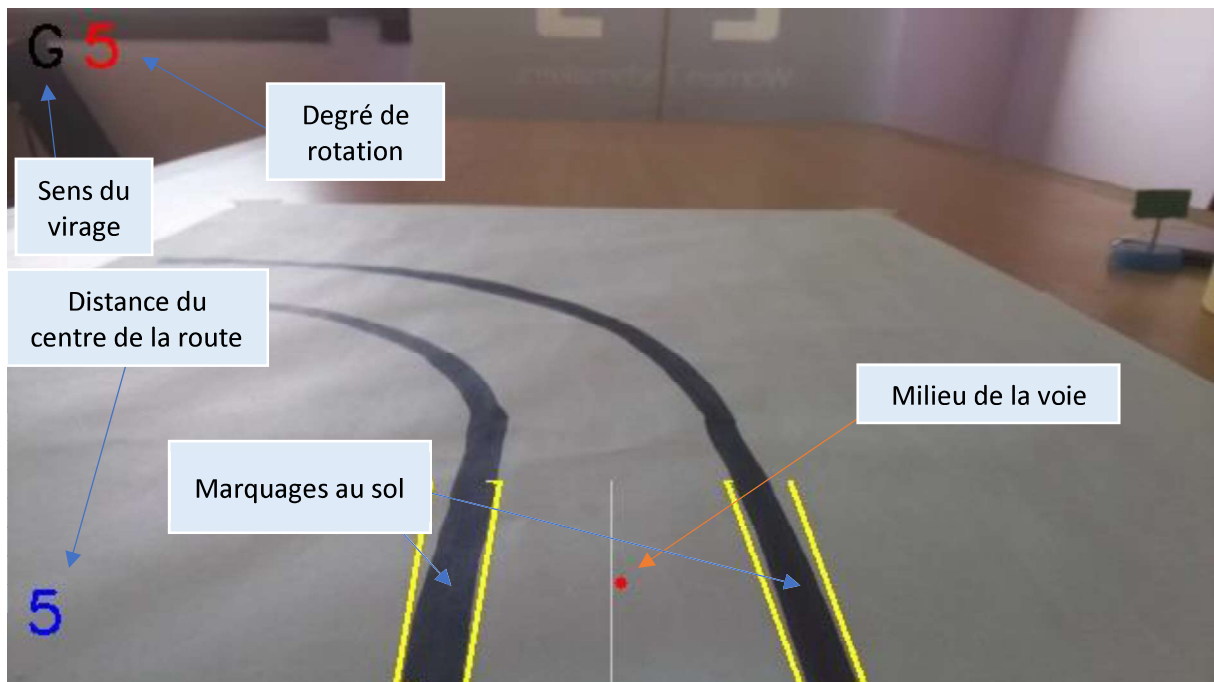


Figure 58 : route droite

- **Avancer** : Chemin libre et Route tracée.

Une fois le véhicule est sur un chemin, la Raspberry capture un flux vidéo avec le module camera puis détecte les lignes de marquage au sol, elle s'assure ensuite que le chemin est libre pour avancer en se servant du module ultrason. Si les lignes sont tracées au sol et que le chemin est libre alors le véhicule avancera.

- **Arrêter** : Obstacle très proche et/ou chemin non tracée.

Si les lignes sont indétectables et/ou un obstacle et très proche de la voiture alors le véhicule s'arrête et ne pourra pas avancer jusqu'à ce que le chemin se libère ou qu'il arrive à détecter les lignes de marquages.

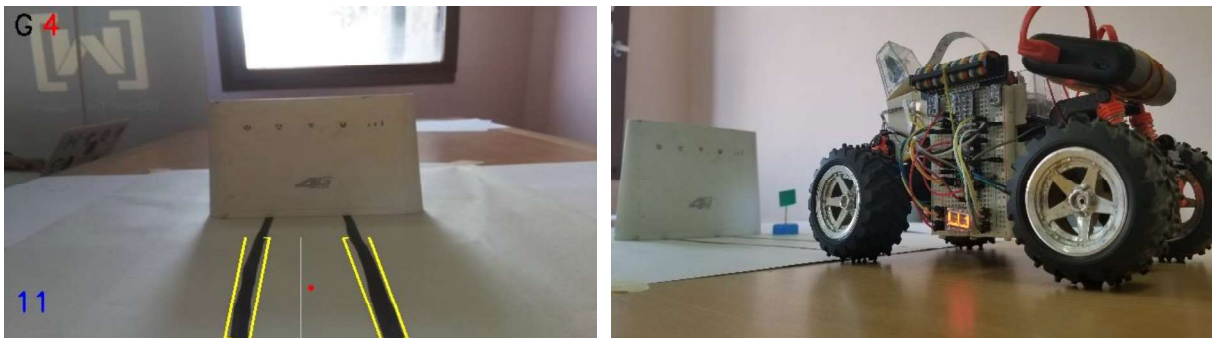


Figure 59 : route barrée par un obstacle

- **Tourner** : Virage à gauche ou virage à droite.

Si le véhicule se trouve sur un virage, les roues directionnelles (avant) sont en mesure de tourner pour permettre au véhicule de s'orienter vers la direction du virage.

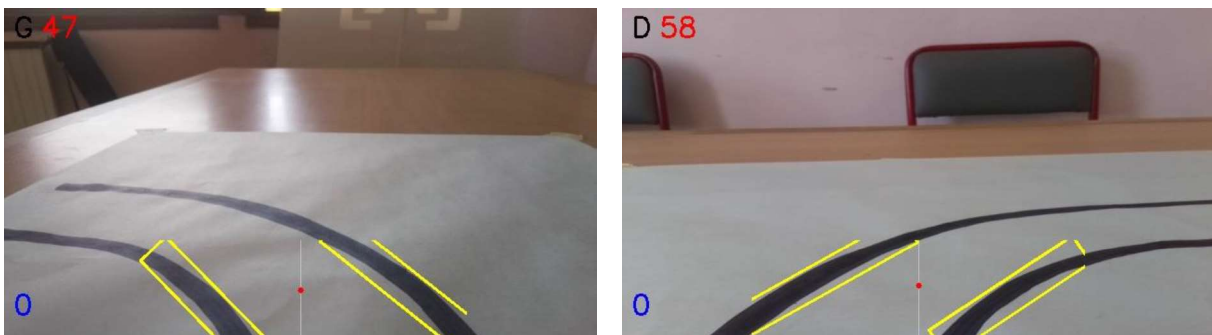


Figure 60 : Exemple de virage gauche et droite

### III. Test

Résultat de la détection de la plaque d'immatriculation :

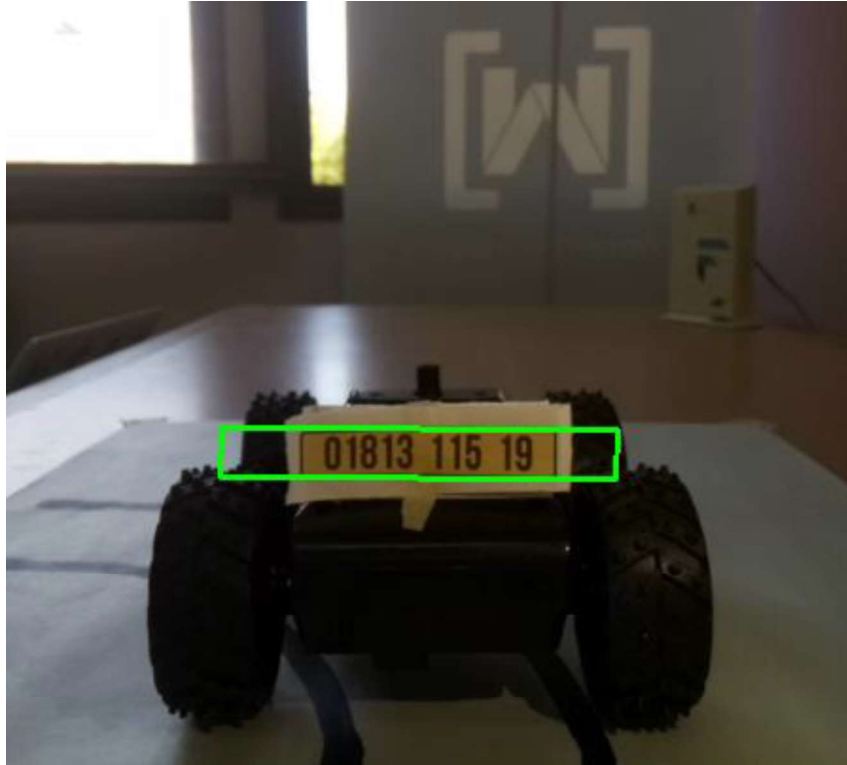


Figure 62 : détection d'une plaque d'immatriculation suspecte

```
Terminal - yacine@yacine-virtual-machine: ~/Bureau/M
Fichier  Édition  Affichage  Terminal  Onglets  Aide
keyboardinterrupt
yacine@yacine-virtual-machine:~/Bureau/Memoire_final$ python3 _server_Tess.py
Recuperation des immatriculations ...
# Recuperation reussie #
-----
working on yacine-virtual-machine with 192.168.43.29
starting up on 192.168.43.29 port 12335
waiting for a connection
connected from: "192.168.43.205"
tesseract starting ...
vectExtend starting ...
-----
nom_conducteur: n_cond1
prenom_conducteur: p_cond1
numero_peris: 1234515161
immatriculation: 0181311519
position:
position: Tizi Ouzou
date_heure: 2019-05-14
date_heure: 2019 09 02 15:38:02
-----
```

Figure 61 : résultat afficher lors de la détection d'un véhicule suspect

#### IV. Conclusion

Au cours de ce de dernier chapitre, nous avons décrit les étapes de réalisation de notre projet. Afin de valider notre approche, nous avons également présenté une évaluation expérimentale de notre système par l'étude d'un cas pratique qui valide la réalisation de notre système.

En premier lieu, nous avons dévoilé le mode de fonctionnement de la partie Software qui s'est basé principalement sur les traitements d'images (reconnaissance de plaques, détection des marquages au sol des routes et détection des feux de signalisation).

En second lieu, nous avons présenté la partie matérielle de notre projet, qui s'est basée sur les différents équipements présentés précédemment, puis nous avons tester et monter les résultats.

## Conclusion Générale

Notre travail avait pour but d'apporter une idée novatrice qui va permettre d'améliorer et de faciliter la gestion et la sécurité du réseau routier, cela en réalisant un véhicule intelligent doté d'un système de reconnaissance de caractères, permettant la lecture automatique des plaques d'immatriculation.

Pour ce faire, nous avons eu recours à plusieurs notions et concepts notamment en traitement d'images afin d'effectuer les prétraitements nécessaires sur les images extraites du flux vidéo capturé, afin de pouvoir détecter et localiser les plaques d'immatriculation et cela en utilisant diverses fonctions de la bibliothèque OpenCv.

Une fois les plaques minéralogiques sont localisées sur l'image, nous avons entamé la phase de Reconnaissance des Caractères et cela grâce à l'OCR Tesseract de Google.

Nous avons aussi pu toucher au côté matériel et cela pour la réalisation de notre véhicule intelligence à l'aide d'une Raspberry Pi 3 model B+ et quelques composants électroniques.

Ainsi la réalisation de ce projet a fait objet d'une expérience intéressante qui nous a permis d'acquérir de nouvelles connaissances tant se le plan théoriques que pratiques, de nous familiariser avec certains concepts de traitement d'images et les systèmes embarqués ...

Toutes fois, nous avons envisagé quelque **perspective**, qui vont nous permettre d'apporter certaines améliorations et perfectionner notre travail. On peut citer :

- Réduire encore plus le temps de traitement des images dans le flux vidéo capturé.
- Ajouter d'autres contraintes et certains prétraitements d'images permettant une meilleure détection (cas des condition défavorables, mauvaise qualité de l'image fournie ...).
- Renforcer l'application avec du Machine Learning pour détecter les véhicules ou les personnes et cela afin améliorer le pilotage.

## Bibliographie

- [1] D. M. E. V. G. Prateek Joshi, OpenCV By Example, Packt Publishing Ltd, 2016.
- [2] Imedias, «bases dessin vectoriel» [En ligne]. Available: <https://www.imedias.pro/cours-en-ligne/logiciels/illustrator/bases-dessin-vectoriel/>. [Accès le 25 04 2019].
- [3] T. Dehesdin, «formats profil couleur et profondeur 8 ou 16 bits d'une image» 2013. [En ligne]. Available: <https://blog.dehesdin.com/workflow/formats-profil-couleur-et-profondeur-8-ou-16-bits-dune-image/>. [Accès le 27 04 2019].
- [4] D. Metz, «Qu'est ce que la luminance ?», 2017. [En ligne]. Available: <http://www.blog-couleur.com/?qu-est-ce-que-la-luminance-96>. [Accès le 01 05 2019].
- [5] J.-M. Courty, «phys chap2» 2018. [En ligne]. Available: <http://www.edu.upmc.fr/physique/lp326/dossiers/phys-chap2.pdf>. [Accès le 10 05 2019].
- [6] IRIV, «Outils fondamentaux en traitements d'images», 2018. [En ligne]. Available: <http://miv.u-strasbg.fr/mazet/ofiti/bases.pdf>. [Accès le 01 05 2019].
- [7] A. Rosebrock, Practical Python and OpenCV 2nd Edition, pyimagesearch, 2016..
- [8] almereyda, «mapschool», 2015. [En ligne]. Available: <https://mapschool.io>. [Accès le 05 05 2019].
- [9] A. Branco, « qu'est-ce que le bruit numérique,» 2014. [En ligne]. Available: <https://www.01net.com/astuces/quest-ce-que-le-bruit-numerique-632881.html>. [Accès le 30 04 2019].
- [10] B. CHARNAY, «the photometrie» 2019. [En ligne]. Available: <https://leclairage.fr/knowledge-base/th-photometrie/>. [Accès le 05 05 2019].
- [11] M. Guillard, «Analyse d'images Détection de contour,» 2014. [En ligne]. Available: <https://slideplayer.fr/slide/1296291/>. [Accès le 20 05 2019].
- [12] M. Fatiha, «Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images,» université Oran U. S. T. O, 2012.
- [13] J.-F. C. Christophe Guyeux, «Les Filtres,» 2012. [En ligne]. Available: <http://cours-info.iut-bm.univ-fcomte.fr/wiki/pmwiki.php/Imagerie/LesFiltres#toc2>. [Accès le 25 05 2019].
- [14] X. Philippeau, «Les filtres usuels en traitement d'images,» 2012. [En ligne]. Available: [https://xphilipp.developpez.com/articles/filtres/?page=page\\_3](https://xphilipp.developpez.com/articles/filtres/?page=page_3). [Accès le 25 05 2019].
- [15] f. mooc, «Notions de traitement et d'analyse d'image,» Agence universitaire de la Francophonie, 2015.
- [16] j. matt, «Traitement d'image,» Université de Rennes 2, 2016.

- [17] B. Perret, «Morphology Mathématique,» 2017. [En ligne]. Available: <https://perso.esiee.fr/~perretb/I5FM/TAI/morpho/index.html>. [Accès le 27 05 2019].
- [18] W. b. salem, «ségmentation d'image,» faculté des sciences de bizerte, 2018.
- [9] M. Bergounioux, «Quelques méthodes de filtrage en Traitement d'Image,» Université d'Orleans, 2010.
- [20] V. BLANCHOT, «Brève histoire de l'intelligence artificielle,» 2018. [En ligne]. Available: <https://siecledigital.fr/2018/08/20/histoire-intelligence-artificielle/>. [Accès le 05 06 2019].
- [21] A. Dusolier, «L' Intelligence Artificielle,» 2013. [En ligne]. Available: <http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/definition-de-l-intelligence-artificielle.html>. [Accès le 07 06 2019].
- [22] R. Lambert, «Différence entre Intelligence Artificielle, Machine Learning et Deep Learning,» 2019. [En ligne]. Available: <http://penseeartificielle.fr/difference-intelligence-artificielle-machine-learning-deep-learning/>. [Accès le 10 06 2019].
- [23] Y. BENZAKI, «Naive Bayes Classifier pour Machine Learning,» 2017. [En ligne]. Available: <https://mrmint.fr/naive-bayes-classifier>. [Accès le 05 06 2019].
- [24] J.-W. Y. S.-H. & C. Y.-S. Hsieh, «Morphology-based license plate detection from complex scenes. Proceedings. 16th International Conference on Pattern Recognition,» 16th International Conference on Pattern Recognition , 2002.
- [25] B. Y. W. T. K. & T. C. Lim, «A novel DSP based real-time character classification and recognition algorithm for car plate detection and recognition,» Fourth International Conference on Signal Processing Proceedings, 1998.
- [26] W. Z. H. & H. X. Jia, «Region-based license plate detection,» Journal of Network and Computer Application, 2007.
- [27] S. K. E. & F. S. Bayoumi, «License Plate Recognition system for Egyptian Car Plates.,» Proc of International Conference on Image and Video Processing and Computer Vision , 2010.
- [28] P. T. Wright, «On-line recognition of handwriting,» GEC Journal of research, 1992.
- [29] wikipedia, «Reconnaissance optique de caractères,» 2017. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Reconnaissance\\_optique\\_de\\_caract%C3%A8res](https://fr.wikipedia.org/wiki/Reconnaissance_optique_de_caract%C3%A8res). [Accès le 05 06 2019].
- [30] T. SRIVASTAVA, «Introduction to k-Nearest Neighbors,» 2018. [En ligne]. Available: <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>. [Accès le 05 06 2019].
- [31] melepe, «Un peu de Machine Learning avec les SVM,» 2017 . [En ligne]. Available: <https://zestedesavoir.com/tutoriels/1760/un-peu-de-machine-learning-avec-les-svm/>. [Accès le 02 06 2019].

- [32] wikiversity, «Réseaux de neurones : Points faibles et limites,» 2017. [En ligne]. Available: [https://fr.wikiversity.org/wiki/Réseaux\\_de\\_neurones/Points\\_faibles\\_et\\_limites](https://fr.wikiversity.org/wiki/Réseaux_de_neurones/Points_faibles_et_limites). [Accès le 02 06 2019].
- [33] B. Perret., «Convolution,» 2017. [En ligne]. Available: <https://perso.esiee.fr/~perretb/I5FM/TAI/convolution/index.html>. [Accès le 02 06 2019].
- [34] D. J. Lee, «Asia Vision Technology Limited,» 2016. [En ligne]. Available: <https://asiavisionweb.com/>. [Accès le 18 06 2019].
- [35] A. Vision, «LAPI-ENGINE,» 2019. [En ligne].
- [36] Q. GmbH, «Reconnaissance Automatisée de Plaques d'Immatriculation,» 2019. [En ligne]. Available: Qognify GmbH 2019. [Accès le 20 05 2019].
- [37] M. Mélançon, «Détection de contours,» DocPlayer, 2016.
- [38] S. Dangare, «Data processing using ML Supervised classification algorithm to find accuracy,» 2019. [En ligne]. Available: <https://blog.knoldus.com/data-processing-and-using-ml-supervised-classification-algorithm-and-find-accuracy/amp/>. [Accès le 07 06 2019].
- [39] A. H, « NUMÉRISATION DE DOCUMENTS ET EXTRACTION (OCR) DES INDEX DE RECHERCHE » 2018. [En ligne]. Available: <http://www.appic-documents.com/numerisation-de-documents/>. [Accès le 01 07 2019].
- [40] R. G, «Tesseract OCR,» 2019. [En ligne]. Available: <https://doc.ubuntu-fr.org/tesseract-ocr>. [Accès le 09 07 2019].
- [41] R. Smith, «An Overview of the Tesseract OCR Engine,» Google Inc, 2015.
- [42] univercité Lille 1, « analyse en composantes connexes,» univercité Lille 1, 2017.
- [43] B. Estibals, « Conception et Commande de Systèmes Electriques Embarqués,» Univ Paul Sabatier, 2017.
- [44] I. D. Briki, «Cours Systeme Embarqué Et Systeme D'Exploitation Mobile,» ESEN, 2017.
- [45] P. PEZZOTTA, «Etude sur l'évolution des métiers et des besoins en formation,» OPIIEC, 2014.
- [46] WIKIBOOKS, «Programmation Python/Introduction,» 2018. [En ligne]. Available: [https://fr.wikibooks.org/wiki/Programmation\\_Python/Introduction](https://fr.wikibooks.org/wiki/Programmation_Python/Introduction). [Accès le 30 06 2019].
- [47] ToturialsTeacher, «Python - IDLE,» 2019. [En ligne]. Available: <https://www.tutorialsteacher.com/python/python-idle>. [Accès le 30 06 2019].
- [48] O. team, «About OpenCV,» 2019. [En ligne]. Available: <https://opencv.org/about/>. [Accès le 30 06 2019].
- [49] B. L, «Base de données : qu'est-ce que c'est ? Définition et présentation,» 2019. [En ligne]. [Accès le 02 07 2019].

- [50] J.-F. Pillou, «Qu'est ce qu'un SGBD ?», 2003 . [En ligne]. Available: <https://stph.scenari-community.org/idl-bd/mob/rss/co/idl-bd7-web/co/priUC002.html>. [Accès le 02 07 2019].
- [51] D. L., «Difference between multitasking, multithreading and multiprocessing,» 2018. [En ligne]. Available: <https://www.geeksforgeeks.org/difference-between-multitasking-multithreading-and-multiprocessing/>. [Accès le 10 07 2019].
- [52] R. Quennec, «Python: Le module pickle,» 2018. [En ligne]. Available: <http://www.quennec.fr/trucs-astuces/langages/python/python-le-module-pickle>. [Accès le 10 02 2019].
- [53] A. Rosebrock, «Practical Python and OpenCV 2nd Edition Dr,» Pyimagesearch, 2016.
- [54] M. Lee, «pytesseract 0.2.9,» 2019. [En ligne]. Available: <https://pypi.org/project/pytesseract/>. [Accès le 10 02 2019].
- [55] TutorialsTeacher, «Python - Math Module,» 2019. [En ligne]. Available: <https://www.tutorialsteacher.com/python/math-module>. [Accès le 04 07 2019].
- [56] Foundation, Python Software, «concurrent.futures — Launching parallel tasks,» 2019. [En ligne]. Available: <https://docs.python.org/3/library/concurrent.futures.html>. [Accès le 11 07 2019].
- [57] B. Bernd Klein, «Fork und Prozesse,» 2018. [En ligne]. Available: <https://www.python-course.eu/forking.php>. [Accès le 05 07 2019].
- [58] S. G., «Microcontrôleur,» 2015. [En ligne]. Available: <http://forge.blogspot.com/2015/07/microcontroleur.html>. [Accès le 12 07 2019].
- [59] G. geek, «Arduino Vs Raspberry pi,» 2019. [En ligne]. Available: <http://www.lafabriquediy.com/tutoriel/arduino-vs-raspberry-pi-86/>. [Accès le 12 07].
- [60] THOMAS, «RASPBerry PI3,» ENCHANTED GUIDE FOR STARTERS, 2017.
- [61] M. Legouge, «Quel OS installer sur un Raspberry Pi,» 2019. [En ligne]. Available: <https://www.clubic.com/raspberry-pi/article-850017-1-os-installer-raspberry-pi.html>. [Accès le 01 07 2019].
- [62] K. Skaarhoj, «systeme exploitation raspberry pi,» 2018. [En ligne]. Available: <https://www.ionos.fr/digitalguide/serveur/know-how/systeme-exploitation-raspberry-pi/> . [Accès le 15 07 2019].
- [63] H. LEAVE, «Kali Linux Raspberry Pi,» 2018. [En ligne]. Available: <https://techooid.com/install-kali-linux-on-raspberry-pi-3/kali-linux-raspberry-pi>. [Accès le 15 07 2019].
- [64] L. A. SIDI YKHLEF Asma, «Mise au point d'une Application de télésurveillance,» université de tlemcen, 2015.
- [65] P. Labdec, «Breadboard,» 2018. [En ligne]. Available: <http://www.zpag.net/Electroniques/Breadboard/Breadboard.htm>. [Accès le 17 07 2019].

- [66] B. Camille, «Fonctionnement du moteur électrique,» 2017. [En ligne]. Available: <http://tpevoitureelectrique.e-monsite.com/pages/introduction/fonctionnement-du-moteur-electrique.html>. [Accès le 15 07 2019].
- [67] Microsonic GmbH, «Capteurs à ultrasons,» 2019. [En ligne]. Available: <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm>. [Accès le 19 07 2019].
- [68] F. MOCQ, «scratch-raspberry-pi-composants,» 2019. [En ligne]. Available: <https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/>. [Accès le 15 07 2019].
- [69] Phil, «installer raspbian sur une carte sd depuis windows,» 2013. [En ligne]. Available: <http://raspberrypis.net/installer-raspbian-sur-une-carte-sd-depuis-windows/>. [Accès le 15 07 2019].
- [70] CFAURY, «Pont en H L298N,» 2018. [En ligne]. Available: <http://arduino.blaise-pascal.fr/pont-en-h-l298n/>. [Accès le 18 07 2019].
- [71] G. Bonvoisin, «Windows 10 : une version IoT Core compatible Raspberry Pi 3,» 2016. [En ligne]. Available: <https://www.cnetfrance.fr/news/windows-10-une-version-iot-core-compatible-raspberry-pi-3-39833468.htm>. [Accès le 15 07 2018].
- [72] semageek, «CAPTEUR DE DISTANCE ULTRASON HC-SR04,» 2019. [En ligne]. Available: <https://boutique.semageek.com/fr/372-capteur-de-distance-ultrason-hc-sr04.html>. [Accès le 18 07 2019].
- [73] Python Software Foundation, «Python,» 2019. [En ligne]. Available: <https://www.python.org/>. [Accès le 28 06 2019].
- [74] OpenCV team, «Open CV,» 2019. [En ligne]. Available: <https://opencv.org/>. [Accès le 28 06 2019].
- [75] S. Algun, «Review for Tesseract and Kraken OCR for text recognition,» 2018. [En ligne]. Available: <https://medium.com/datadriveninvestor/review-for-tesseract-and-kraken-ocr-for-text-recognition-2e63c2adedd0>. [Accès le 28 07 2019].