



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

THEME

**Systeme de localisation «indoor» en
utilisant des techniques intelligentes à
base de traitement de l'image**

Réalisé par : MESSAI Yacine

Soutenu le : 15/09/2020 *Devant le jury composé de:*

Mr. DJAMAI Djmoui
Mr. DOUAK Fouzi
Mme. FRIHA Souad

Président *Université Abbes Laghrour-Khenchela*
Encadreur *Université Abbes Laghrour-Khenchela*
Examineur *Université Abbes Laghrour-Khenchela*

Promotion 2019/2020

Je dédie ce travail à :

Ma mère,

Mon père,

Mes frères et mes sœurs,

Tous mes amis sans exception.

Yacine

Remerciements

Je tiens à remercier ALLAH tout puissant qui ma a donné la santé, le courage, et la volonté pour réaliser ce travail.

Je remercie mon encadreur Monsieur Fouzi Douak, enseignant à l'université de Khenchela, pour son suivi, pour son aide précieux,

J'adresse mes plus sincères remerciements aux examinateurs, pour avoir accepté de juger ce travail, et qui à font l'honneur d'être membres de ce Jury.

Je tiens à remercier particulièrement tous les enseignants et les responsables de notre département, qui ont veillé sur notre formation.

Mes remerciements vont également à tout le personnel du département du génie industriel pour leur gentillesse et leur disponibilité.

Enfin, j'exprime mon remerciement à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Résumé

Dans ce projet, nous avons étudié le problème de la localisation «indoor» qui permet à une entreprise de créer de nouveaux services digitaux grâce au guidage à l'intérieur des murs, là où les signaux GPS ne passent plus. Avec une précision plus ou moins grande, on parle d'une précision de quelque mètre à quelque dizaine de mètre.

A cet effet et à l'aide de l'intelligence artificielle, précisément l'apprentissage profond qui a considérablement amélioré les résultats dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole et la traduction automatique. Les techniques d'apprentissage profond permettent, à l'aide de données, de résoudre de nombreux problèmes dans de nombreux domaines tels que la géolocalisation indoor ou la précision est un facteur lourdement indispensable.

Afin de minimiser les erreurs de précision, nous avons proposé un système de localisation dans les environnements intérieurs «indoor» en utilisant les méthodes de l'intelligence artificielle, cette solution est basée sur l'apprentissage profond (deep learning) et les différentes techniques de traitement de l'image (seuillage, filtrage et classification). Notre proposition est soutenue par des tests qui ont donné des résultats acceptables.

Mots clés: GPS, indoor, Apprentissage profond, Seuillage, CNN, Filtrage.

Abstract

In this project, we studied the problem of indoor location which allows a company to create new digital services using guidance inside the walls, where GPS signals no longer pass. With bad precision, we are talking about a precision of a few meters to ten meters.

For this purpose and with the help of artificial intelligence, precisely deep learning that has dramatically improved results in many areas such as computer vision, speech recognition and machine translation. Deep learning techniques allow, with the help of data, to solve many problems in many fields such as indoor geolocation or precision is a heavily essential factor.

In order to minimize precision errors, we have proposed a localization system in indoor environments using artificial intelligence methods; this solution is based on deep learning and the different image processing technique (thresholding, filtering, and classification). Our proposal is supported by tests which have given acceptable results.

Keywords: GPS, Indoor, Deep learning, Thresholding, CNN, Filtering.

ملخص

في هذا المشروع، بحثنا في مشكلة تحديد الموقع الجغرافي داخل هياكل المؤسسات الذي يسمح للشركة بإنشاء خدمات رقمية جديدة باستخدام التوجيه داخل الجدران، حيث لم تعد إشارات GPS تمر. بدقة أكبر أو أقل، نحن نتحدث عن دقة تتراوح من بضعة أمتار إلى عشرة أمتار.

لهذا الغرض وبمساعدة الذكاء الاصطناعي، خاصة التعلم العميق الذي أدى إلى تحسين النتائج بشكل كبير في العديد من المجالات مثل رؤية الكمبيوتر والتعرف على الكلام والترجمة الآلية. تسمح تقنيات التعلم العميق، بمساعدة البيانات، بحل العديد من المشكلات في العديد من المجالات مثل تحديد الموقع الجغرافي الداخلي أين تكون الدقة عامل أساسي للغاية. ولتقليل أخطاء الدقة، اقترحنا نظام تحديد الموقع في البيئات الداخلية باستخدام أساليب الذكاء الاصطناعي، يعتمد هذا الحل على التعلم العميق والتقنيات المختلفة في معالجة الصور (العتبة، التصنيف والتصفية). اقترحنا مدعوم باختبارات أعطت نتائج مقبولة.

الكلمات المفتاحية : GPS، داخلي، التعلم العميق، العتبة، CNN، التصنيف.

Liste des tableaux

Tableau III. 1. Fonctions de transfert $a=f(n)$	42
Tableau IV. 1. Résultats de la première architecture de notre modèle d'approche par apprentissage profond sur le jeu de test.....	57
Tableau IV. 2. Configuration de l'architecture 2.	58
Tableau IV. 3. Résultats de la deuxième architecture de notre modèle d'approche par apprentissage profond sur le jeu de test.....	59
Tableau IV. 4. Validation sur le jeu de validation des classifieurs 28×28 et 48×48.....	60

Liste des figures

Figure I. 1. Principe de la trilatération.....	6
Figure I. 2. Principe de fonctionnement du système Active Bat.....	9
Figure I. 3. Localisation par UltraSon.....	9
Figure I. 4. Localisation par Active BAT.....	10
Figure I. 5. Système Active BAT: Les récepteurs ultrasons.	11
Figure I. 6. Principe d'émission – réception.....	12
Figure II. 1. Raspberry <i>Pi</i> 3.....	16
Figure II. 2. Pins GPIO et Leur Fonctions.	17
Figure II. 3. Logo du système d'exploitation Raspbian et Windows 10 IoT.....	18
Figure II. 4. Adresse IP de la carte Raspberry <i>Pi</i>	18
Figure II. 5. Fichier de configuration réseau.....	19
Figure II. 6. Logiciel Putty.....	20
Figure II. 7. Invite de commande via SSH.....	21
Figure II. 8. Connexion ordinateur et Raspberry <i>Pi</i>	22
Figure II. 9. L'environnement de programmation pour Raspberry <i>Pi</i> par l'intermédiaire du support package Simulink.	23
Figure II. 10. Raspberry <i>Pi</i> 3 et camera.	24
Figure III. 1. La relation entre l'intelligence artificielle, le ML et le deep learning.....	31
Figure III. 2. Comparaison entre (a) ML classique et (b) deep learning.....	33
Figure III. 3. Les réseaux de neurones convolutifs (CNN).....	33
Figure III. 4. Pooling avec un filtre 2×2 et un pas de 2.	34
Figure III. 5. Exemple de max pooling (2×2).	36
Figure III. 6. L'apprentissage supervisé.....	38
Figure III. 7. La classification avec KNN.....	39
Figure III. 8. L'hyperplan optimal (en rouge) avec la marge maximale. Les échantillons entourés sont des vecteurs supports.....	40

Figure III. 9. Modèle d'un neurone artificiel.	41
Figure III. 10. Couche de S neurones.	43
Figure III. 11. Structure d'un réseau de neurone multicouche.	44
Figure IV. 1. Les 5 classes des marqueurs	48
Figure IV. 2. Exemples sur les effet appliqués sur les marqueurs originals.	48
Figure IV. 3. Images de test.	49
Figure IV. 4. Image de Test 1 segmentée par les méthodes basées sur l'entropie.	49
Figure IV. 5. Image de Test 1 segmentée par les méthodes basées sur la similarité.	50
Figure IV. 6. Image de Test 1 segmentée par les méthodes basées sur la forme de l'histogramme.	50
Figure IV. 7. Image de Test 2 segmentée par les méthodes basées sur l'entropie.	51
Figure IV. 8. Image de Test 2 segmentée par les méthodes basées sur la similarité.	51
Figure IV. 9. Image de Test 2 segmentée par les méthodes basées sur la forme de l'histogramme.	52
Figure IV. 10. Etiquetage en composantes connexes de l'image de test 1.	53
Figure IV. 11. Etiquetage en composantes connexes de l'image de test 2.	53
Figure IV. 12. Détection de marqueurs Aruco de l'image de test 1.	54
Figure IV. 13. Détection de marqueurs Aruco de l'image de test 2.	54
Figure IV. 14. Diagramme block d'identification des marqueurs.	55
Figure IV. 15. Résultats d'apprentissage de l'architecture 1 , en termes de (a) precision (Accuracy) et (b) d'erreur (Loss).	58
Figure IV. 16. Résultats d'apprentissage de l'architecture 2 en termes de precision (Accuracy) et d'erreur (Loss).	59
Figure IV. 17. Matériel utilisé dans le test.	60
Figure IV. 18. Système de localisation.	61
Figure IV. 19. Les captures des étapes du test.	61

Table des matières

Introduction Générale	1
------------------------------------	---

Chapitre I

Généralités sur la localisation

I.1. Introduction	4
I.2. La localisation indoor.....	4
I.3. Principales technologies utilisées pour la localisation	5
I.3.1. Localisation par ondes radio.....	5
I.3.1.1. GPS	5
I.3.1.2. Ultra Wide Band	7
I.3.1.3. Bluetooth	7
I.3.1.4. Wifi	7
I.3.1.5. Rayonnement infrarouge	8
I.3.1.6. Ultrason	9
I.3.1.7. Radio-identification	11
I.3.1.8. La radio fréquence	12
I.3.2. Capteurs	13
I.3.2.1. Systèmes micro électromécaniques	13
I.3.2.2. Caméra.....	13
I.4. Conclusion.....	13

Chapitre II

Traitement d'images sur Raspberry Pi

II.1. Introduction.....	15
II.2. Le Raspberry <i>Pi</i>	15
II.2.1. Raspberry <i>Pi</i> 3	15
II.2.2. Les Ports GPIO.....	17
II.2.3. Aspects Logiciel	17
II.3. Accès à distance au Raspberry <i>Pi</i>	18

_____ II.3.1. Choix de l'adresse IP du Raspberry Pi.....	18
_____ II.3.2. Connexion à distance via SSH	20
II.4. Raspberry Pi basé sur des langages de programmations.....	21
_____ II.4.1. Analyse des données à partir la carte Raspberry Pi	21
_____ II.4.2. Programmation Raspberry Pi	22
_____ II.4.3. Développer des algorithmes Simulink	22
II.5. Segmentation utilisant Raspberry Pi.....	23
_____ II.5.1. Binarisation basée sur le seuillage	24
_____ II.5.1.1. Méthodes de seuillage global	24
_____ II.5.1.2. Méthodes basées sur l'entropie	24
_____ II.5.1.2.1. Méthode de Kapur	25
_____ II.5.1.2.2. Méthode de seuillage par entropie croisée (Li et Lee)	25
_____ II.5.1.2.3. Méthode de Yen et al.	26
_____ II.5.1.3. Méthodes basées sur les méthodes de similarités.....	26
_____ II.5.1.3.1. Méthode de Tsai et Lin.....	26
_____ II.5.1.3.2. Méthode de Huang et Wang.....	26
_____ II.5.1.3.3. Méthode de Hertz et Schafer	27
_____ II.5.1.4. Méthodes basées sur la forme de l'histogramme	27
_____ II.5.1.4.1. Méthode d'Otsu	27
_____ II.5.1.4.2. Méthode de Ridler et Calvard	28
_____ II.5.1.4.3. Méthode de Lloyd	29
II.6. Conclusion	29

Chapitre III

Apprentissage profond (deep learning)

III.1. Introduction	31
III.2. Définition de l'apprentissage profond (deep learning).....	31
III.3. Architectures de réseaux de neurones profonds	32
_____ III.3.1. Principe d'architecture d'un CNN	32
_____ III.3.2. Les blocs de construction.....	33
_____ III.3.2.1. La couche de convolution (CONV)	33
_____ III.3.2.2. Couche de pooling (POOL)	34

III.3.2.3. Couche de correction (ReLU) :	35
III.3.2.4. Couche entièrement connectée (FC)	35
III.3.2.5. Couche de perte (LOSS)	35
III.3.3. Choix des hyperparamètres	36
III.3.3.1. Nombre de filtres	36
III.3.3.2. Forme de filtres	36
III.3.3.3. Forme de Max pooling	36
III.3.4. Avantages de CNNs	37
III.3.5. La classification	37
III.3.5.1. L'objectif de la classification	37
III.3.5.2. Domaines d'application de la classification	37
III.3.5.3. Les différentes méthodes de la classification et l'apprentissage	38
III.3.5.3.1. Méthodes supervisées	38
III.3.5.3.2. Méthodes non-supervisées	38
III.4. Les algorithmes d'apprentissage automatique	39
III.4.1. Le voisin K le plus proche (K-Nearest Neighbours : KNN)	39
III.4.2. Machine à vecteurs de support	39
III.4.2.1. Principe général	40
III.4.3. Les Réseaux de Neurones	40
III.4.3.1. Fonctions de transfert	41
III.4.3.2. Architecture de réseau	43
III.4.3.3. Le perceptron multicouche (Multi Layer Perceptron MLP)	43
III.4.4. Réseaux Neurones Convolutionnels	44
III.5. Conclusion	45

Chapitre IV

Tests et résultats

IV. 1. Introduction	47
IV. 2. Description des bases de données	47
IV. 3. Evaluation des algorithmes de segmentation implémentés	48
IV. 4. Evaluation des algorithmes d'apprentissage profond implémentés	54
IV. 5. Environnement de développement	56

IV.5.1. Résultat de l'apprentissage	56
IV. 6. Réalisation finale et tests	60
IV. 7. Conclusion.....	62
Conclusion générale.....	64
<i>Bibliographie</i>	66

Introduction générale

Introduction générale

La localisation joue un rôle important dans la vie quotidienne, elle devenue indispensable dans le cadre de positionnement des locaux, des véhicules et des tracés des chemins...etc. les GPS sont des dispositifs électroniques fréquemment dans le cadre de la localisation, leur limitation résident dans le fait que les environnements fermés ne sont pas bien définis sa dû à la mauvaise pénétration des signaux GPS à l'intérieur des bâtiments à cause des obstacles [1]. Nombreuses technologies sans fil permettent la géolocalisation «indoor» : le Li-Fi qui se base essentiellement sur la lumière pour la localisation, l'ultrason qui utilise la géolocalisation sonore inaudible pour l'homme, l'étiquette RFID composée de puces pour identifier les objets,... etc. [2, 3]. En plus, beaucoup de technologies complémentaires sont aujourd'hui disponibles pour répondre à des besoins différents.

La plupart des solutions du problème de la localisation «indoor» se basent sur un mixage entre des signaux Wi-Fi et Bluetooth. La localisation «indoor» par le Wi-Fi s'appuie essentiellement sur le réseau Wi-Fi existant, Il offre la possibilité de déterminer, grâce à la mesure de l'intensité du signal reçu et d'une méthode d'empreinte, la position d'un utilisateur dans un espace défini.

De nombreux usages et avantages sont donnés par la mise en place d'une géolocalisation «indoor» [4]. D'abord, elle permet à la personne qui utilise la technologie sur son smart phone de se repérer dans les locaux ou elle se trouve et de s'orienter vers l'espace souhaité. En plus, d'avoir une fonction d'itinéraire personnalisé.

Le type de structures déterminera les besoins et les solutions à chercher dans la géolocalisation « indoor ». Son interaction avec les objets connectés présents décuplent les possibilités de services. Par exemple, pour les centres commerciaux, permet de trouver un magasin par une application qui se comporte comme un jeu d'enfant pour le client. L'application lui permettra également d'avoir un itinéraire ou des offres personnalisées (soldes, pop-up store, etc.).

Dans ce cadre notre travail a pour objectif de mettre en œuvre un système de positionnement à l'intérieur d'un espace fermé, où l'utilisateur est menu d'une caméra et une carte Raspberry Pi. Cela est effectué en basant sur les méthodes de l'intelligence artificielle. La méthode proposée dans ce mémoire s'inspire de l'algorithme de la détection et la classification basée sur l'apprentissage profond des marqueurs dans un environnement «indoor» [5].

Le système de navigation «indoor» développé a pour but de détecter des marqueurs étiquettes. Dans la littérature scientifique, plusieurs types des marqueurs utilisés tels que les codes QR

(Quick Response) [6, 7], les marqueurs AR (Augmented Reality) [8, 9], et les marqueurs Aruco [10]. Dans notre système nous avons utilisé les marqueurs Aruco, ce type de marqueur se forme d'un carré où le processus de détection est effectué en trois étapes principales : la segmentation, la détection des marqueurs et l'analyse de codification.

Cependant, les images capturées dans de nombreuses applications réelles peuvent présenter des problèmes tels que le flou dû au mouvement, la distorsion du marqueur et l'occlusion du marqueur, ce qui rend l'identification difficile. Pour cela, nous avons proposé un système de navigation pour améliorer l'identification des marqueurs dans des conditions difficiles en utilisant les réseaux de neurones convolutifs (CNN). Le problème d'identification est converti en un problème de classification. Dans ce mémoire, on présente un système de localisation pour aider les utilisateurs à naviguer à l'aide des marqueurs. Dans ce système, les marqueurs ont été imprimés sur des papiers et installés dans des endroits intérieurs. Lorsqu'un marqueur est détecté pendant la navigation à l'aide d'une caméra et d'un Raspberry Pi; le système utilise ce marqueur comme position dans un plan ou Map. Pendant la navigation, lorsque les marqueurs sont détectés, les commandes de navigation sont envoyées à un PC connecté par l'intermédiaire d'une interface de communication WiFi.

L'organisation générale du mémoire est décrite ci-dessous :

Le chapitre I est consacré à la présentation de l'état de l'art des systèmes de la localisation.

Le chapitre II traite deux parties. Dans la première partie nous présentons l'élément principal de notre système qui est la carte Raspberry Pi, dans la deuxième partie, nous détaillons les différentes méthodes de segmentation utilisée dans notre travail.

Le chapitre III une étude générale sur les méthodes de classification en utilisant l'apprentissage profond (deep learning).

Dans le dernier chapitre, on va implémenter les méthodes de segmentation et de classificateur CNN pour la détection et l'identification des marqueurs et faire les tests et l'analyse des résultats obtenus.

La dernière partie de ce mémoire est consacrée à la conclusion générale.

Chapitre I

Généralités sur la localisation

I.1. Introduction

Le déplacement est une nécessité vitale pour chaque personne. De tout temps, l'homme a développé des moyens techniques pour faciliter ses déplacements et augmenter son autonomie. Jusqu'à maintenant, les signaux de navigation GPS qui sont utilisés pour le positionnement à l'extérieur ne peuvent pas être bien captés à l'intérieur des bâtiments. Dans un tel environnement, le système GPS fonctionne en mode dégradé, ou ne fonctionne plus. Ces limites proviennent de : (i) Atténuation produite par les obstacles, les signaux reçus ont un faible SNR [Signal-to-Noise Ratio] (30db de moins qu'à l'extérieur). En outre, à chaque obstacle traversé correspond une atténuation de la puissance du signal. Ce qui implique une faible précision de positionnement. (ii) le positionnement à l'intérieur est soumis aux problèmes de multi-trajets, de masquage, des interférences... La superposition de différents signaux qui ont suivi différents chemins à l'antenne de réception, produit une distorsion de signal original.

I.2. La localisation indoor

Egalement connue sous les appellations «indoor location » ou «indoor positioning», la géolocalisation indoor permet de connaître avec une précision plus ou moins grande, la localisation d'une personne ou d'un produit dans un espace ou un lieu fermé c'est-à-dire à l'intérieur des bâtiments. On trouve plusieurs applications de ce type de localisation. Parmi ceux-ci, on peut citer [1]:

- **Aide à la navigation** : Dans le cas d'une navigation allant de l'outdoor à l'indoor (ex : Un client utilisant son application GPS pour se rendre de chez lui à un point d'intérêt précis situé à l'intérieur d'un centre commercial), celle-ci se doit d'être «sans coutures» : le mobile devra immédiatement détecter que l'utilisateur est rentré dans un bâtiment, et passer de manière transparente en mode navigation intérieure.
- **Sites industriels** : Une fois un problème identifié sur un site industriel, une application mobile pourrait permettre de guider l'équipe technique mobile la plus proche vers la source du problème. Ensuite, une fonctionnalité d'aide à la résolution de problème type Réalité augmentée ou Télé-assistance pourrait prendre le relai si besoin.
- **Transports** : Permettre à un voyageur de naviguer non seulement en extérieur, mais aussi à l'intérieur d'un lieu de type aéroport, gare ou station de métro.

- **Santé** : Permettre au personnel de santé de retrouver des patients ou du matériel est un besoin souvent formulé. Concrètement, une application peut permettre la géo-localisation des nourrissons, des personnes souffrant d'Alzheimer, ou des appareils spécialisés égarés.
- **Marketing** : Mais la principale application marketing serait le géomarketing: optimisation des revenus au mètre carré, en fonction des données de visite des clients (flux, temps de visite, zones de passages, zones de transformation, etc.). Il est possible de personnaliser des offres marketing, non seulement selon les données connues sur le client par la marque (carte de fidélité) mais aussi selon le contexte géographique d'un client en magasin.

I.3. Principales technologies utilisées pour la localisation

On peut distinguer deux catégories. La première met en œuvre un bouquet d'émetteurs et un récepteur (GPS, UWB, Wifi, Bluetooth, RFID, ... etc.) tandis que la seconde s'appuie sur un ou plusieurs capteurs autonomes (caméra) sans dépendance à l'infrastructure.

I.3.1. Localisation par ondes radio

Les systèmes communicants, ou encore les systèmes basés sur le principe émetteur récepteur, sont les premiers à être exploités pour assurer la localisation. Dans ce paragraphe nous allons présenter les principaux systèmes communicants utilisés dans la localisation indoor.

I.3.1.1. GPS

Le GPS est un système qui offre le service de positionnement civil avec une précision de 5 à 15 *m*. Il comprend au moins 24 satellites orbitant à 20 200 *km* d'altitude. Ces satellites transmettent régulièrement les informations nécessaires au calcul de leur position au récepteur qui, grâce à la connaissance de la distance qui le sépare des satellites, peut accéder à ses coordonnées. Une des améliorations du GPS, qui a été proposée afin d'augmenter sa précision, est le GPS différentiel (Differential Global Positioning System DGPS). Ce dernier utilise un réseau de stations fixes de référence qui transmet l'écart entre les positions indiquées par les satellites et leurs positions réelles connues. En recevant la référence entre les pseudo-distances mesurées par les satellites et les véritables pseudo-distances, le récepteur peut corriger ses mesures de positions. Cependant, cela n'apporte pas de solutions pour les inconvénients du GPS à savoir : le non opérabilité à l'intérieur des bâtiments, la dépendance à l'infrastructure ainsi que le manque de précision qui reste toujours faible même avec le DGPS pour la localisation indoor.

La trilatération est la méthode la plus basique et intuitive pour de calcul de la position d'un nœud par l'intersection de trois cercles, comme le montre la Figure I.1. Cette méthode est basée sur la résolution d'un système de trois équations à trois inconnues par rapport à trois satellites de positions connues (équations I.1, I.2, I.3). Par exemple, si on considère que la position inconnue d'un récepteur en deux dimensions (2D) est (x, y) et que les coordonnées des trois satellites sont $(x_i, y_i, i = 1, 2, 3)$, nous obtenons le système de trois équations à résoudre ci-dessous :

$$R_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (\text{I.1})$$

$$R_2^2 = (x - x_2)^2 + (y - y_2)^2 \quad (\text{I.2})$$

$$R_3^2 = (x - x_3)^2 + (y - y_3)^2 \quad (\text{I.3})$$

ou R_i : La distance entre le nœud à localiser et le $i^{\text{ème}}$ satellites.

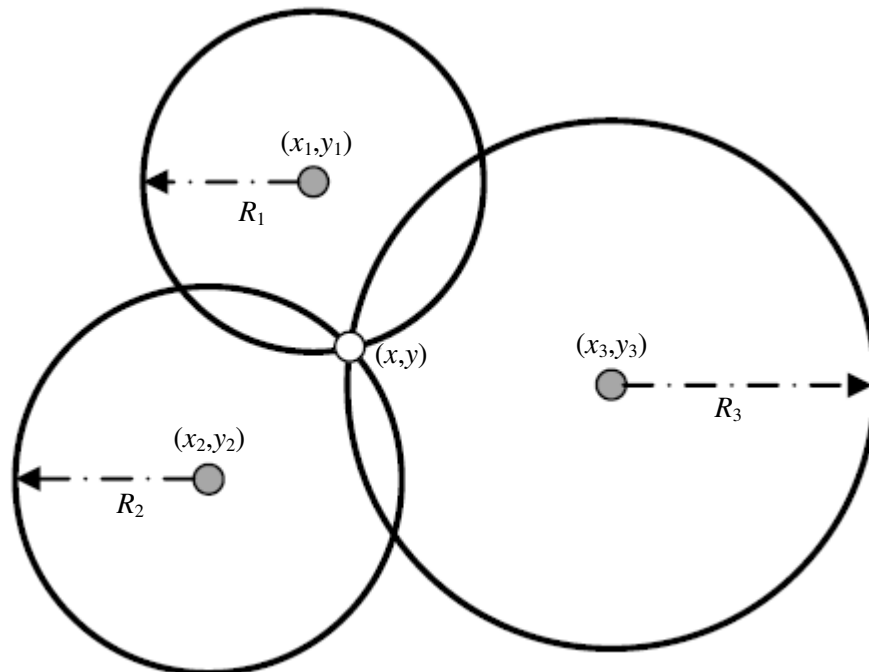


Figure I. 1. Principe de la trilatération.

En soustrayant l'équation (II.3) des deux premières équations, on obtient un système linéaire avec deux équations à deux inconnus (x, y) qui peut donner une seule solution [11]:

$$x(x_3 - x_1) + y(y_3 - y_1) = \frac{1}{2}(x_3^2 - x_1^2 + y_3^2 - y_1^2 + R_1^2 - R_3^2) \quad (\text{II.4})$$

$$x(x_3 - x_2) + y(y_3 - y_2) = \frac{1}{2}(x_3^2 - x_2^2 + y_3^2 - y_2^2 + R_2^2 - R_3^2) \quad (\text{II.5})$$

Dans ce système, les distances R_i sont calculées par le temps de vol ou la durée du trajet reliant le satellite au récepteur (T_i) [$R_i = c \cdot T_i$, $c \approx 3 \cdot 10^8 \text{ m/s}$] que met le signal GPS pour faire le trajet entre le récepteur et les satellites. Si le temps T_i est mesuré avec précision, la résolution de ce système suffit pour déterminer la position du récepteur. Cependant, en réalité, le temps de vol est entaché d'erreur de synchronisation entre l'horloge du récepteur et celle du satellite. Cette incertitude sur le temps est considérée comme une quatrième inconnue, d'où la nécessité d'une quatrième équation (donc d'un quatrième satellite) pour qu'un récepteur puisse déterminer sa position. C'est pour cette raison que le nombre minimal de 4 satellites, est fixé pour le positionnement 2D [12].

I.3.1.2. Ultra Wide Band

Cette technologie a été développée en 1960 pour des applications radars utilisant une communication sans fil. C'est une technique de modulation radio qui est basée sur la transmission d'impulsions de très courte durée souvent inférieure à la nanoseconde.

L'UWB peut être utilisé pour la communication comme il peut être également utilisé pour le positionnement de mobiles. En effet, les récepteurs, qui sont répartis dans l'environnement, peuvent évaluer la distance des balises UWB situées sur les mobiles, par mesure du temps de propagation des signaux émis par ces balises. Ceci permet au système de calculer la position des mobiles par triangulation.

I.3.1.3. Bluetooth

Le Bluetooth ou standard IEEE 802.15, est un protocole de communication à courte portée de données. Il utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. La position d'un appareil mobile à l'aide de cette technologie est considérée comme la même que celle de la cellule individuelle qu'il est en communication avec. L'inconvénient majeur d'un tel système de localisation est que sa précision dépend fortement du nombre de cellules installées et de leurs tailles.

I.3.1.4. Wifi

Le Wifi aussi connu sous le nom du standard IEEE 802.11, est un protocole de communication plus sophistiqué que le Bluetooth. La localisation par Wifi ressemble à celle par Bluetooth: elle approche la position de la personne en se basant sur certaines caractéristiques de la propagation des signaux. L'inconvénient de la localisation par Wifi est qu'elle dépend d'une infrastructure relativement coûteuse dans l'endroit où on veut assurer le suivi d'une personne.

Le processus de localisation se fait en deux phases, une phase offline et une phase online :

- **La Phase « Offline »:** Durant cette phase, le système RADAR mesure les puissances des signaux reçus par le mobile en provenance des points d'accès. Deux méthodes sont utilisées : une méthode théorique et une méthode empirique. En effet, théoriquement, en espace libre, la puissance d'un signal radio est donnée par une fonction logarithmique de propagation de l'onde radio [13]. Mais, à l'intérieur des bâtiments, le signal radio subit plusieurs perturbations (réflexions, trajets multiples...). Le système RADAR réalise alors un ensemble de mesures expérimentales dans un nombre de points de mesure [14] réalisant ainsi une carte de couverture radio (radio map). Cette carte sera stockée dans une base de données qui catégorise les niveaux de signaux reçus à partir de chacun des points d'accès en un point du plan.
- **La Phase « Online »:** pendant laquelle, le système RADAR mesure la puissance de signaux reçus par les points de mesure. En consultant la base de données construite pendant la phase offline, le système localise le point auquel on reçoit des puissances de signaux similaires à celles enregistrées dans la base de données.

Le principal avantage du système RADAR est son coût réduit. En effet, seulement un nombre réduit de stations de base est nécessaire (trois stations de base sont déployées). Outre ce fait, la mise en place de ce système ne nécessite pas un coût supplémentaire puisque RADAR utilise la même infrastructure (réseau local sans fil) existante dans un bâtiment [15].

I.3.1.5. Rayonnement infrarouge

C'est un rayonnement électromagnétique qui a été exploité dans plusieurs utilisations comme le chauffage, la vision nocturne, la communication, le contrôle d'authenticité de billets de banque, les détecteurs d'intrusions, etc. Il a été utilisé également pour la localisation : tout comme la localisation à ultrason, on se sert de la communication entre les émetteurs et le récepteur à infrarouge pour assurer la localisation.

Comme dans le système Active Bat/Cricket, le système infrarouge émet une série de pulses. Cette technologie a été retenue à cette période car elle est peu coûteuse. De plus, la portée des capteurs utilisés est de 6m. Pour des utilisations dans de petites pièces, de nombreuses réflexions sont présentes et facilitent la détection. Le désavantage par rapport aux technologies radio, c'est que les signaux ne traversent pas les murs, ce qui réduit la portée du système [16].

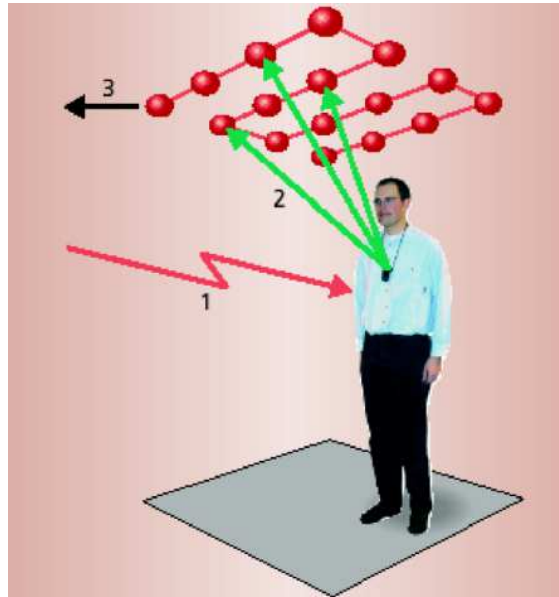


Figure I. 2. Principe de fonctionnement du système « Actif Bat ».

I.3.1.6. Ultrason

Une des applications des ultrasons (onde mécanique et élastique, diffusée par un objet ou un corps dont la fréquence est supérieure à 20 kHz) est la localisation d'un mobile: le récepteur fixé sur le mobile à localiser reçoit les ultrasons émis par plusieurs émetteurs, la différence des temps d'arrivée des signaux ultrasons lui permet d'estimer la distance qui le sépare aux émetteurs. En réitérant cette même mesure avec plusieurs émetteurs, on détermine précisément la position du mobile (qui est celle du récepteur) dans l'environnement.

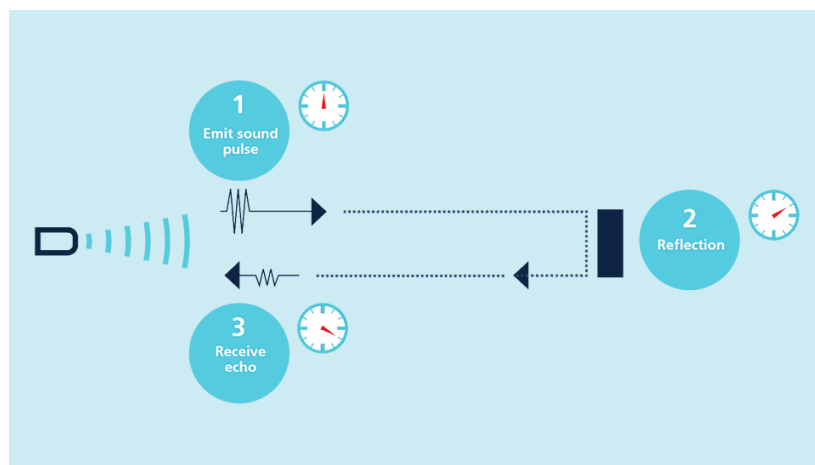


Figure I. 3. Localisation par UltraSon.

Le principal avantage de la technologie d'ultrasons découle de la lenteur de propagation du son par comparaison à celle des ondes radio ce qui donne une meilleure précision de la mesure du temps de propagation et ainsi des estimations de distances plus fiables. Généralement, les systèmes de localisation par ultrasons font appel à une autre technologie afin d'obtenir une estimation de la distance entre l'émetteur et le récepteur [16].

Un des systèmes de localisation indoor conçus en utilisant la technologie ultrason est un système opère d'une manière semblable à celle de l'Active Badge, mais au lieu d'employer des impulsions infrarouge, un message radio et une impulsion ultrason sont émis par un émetteur en même temps et d'une façon périodique. La technique de trilatération basée sur le temps d'arrivée d'un signal ultrason offre une précision centimétrique ce qui fournit une localisation plus fine et plus précise comparée aux performances de l'Active Badge. Les objets à localiser portent des tags Active Bat [17] comme le montre la Figure suivante.

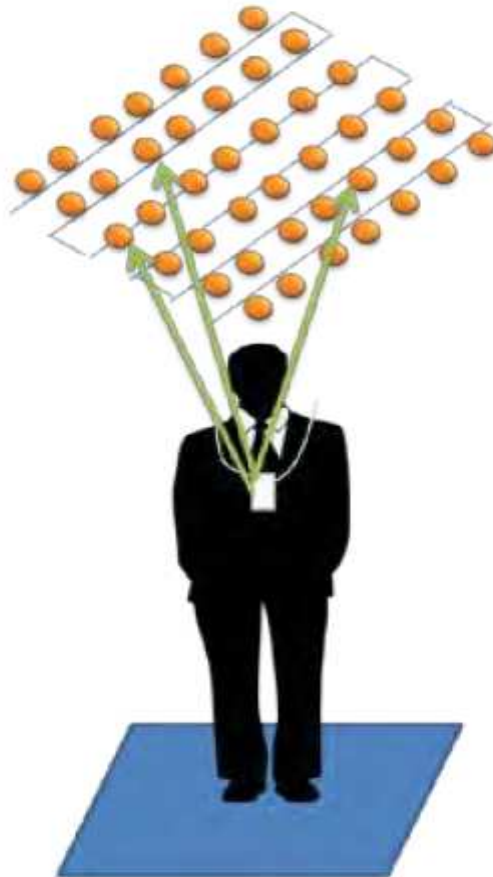


Figure I. 4. Localisation par « Active BAT ».

Chaque Active Bat dispose d'un identifiant unique. L'active Bat s'appuie sur plusieurs récepteurs à ultrasons situés dans le plafond d'un bâtiment et reliés à l'aide d'un réseau à haute vitesse (Figure I.5). En réponse à une requête envoyée par le biais d'une onde radio à

courte-distance par un contrôleur, un Bat envoie une pulse ultrason vers cette grille de récepteurs.



Figure I. 5. Système Active BAT: Les récepteurs ultrasons.

Ainsi, on peut calculer la position d'un objet si l'on trouve les positions relatives des bats qui lui sont attachés. En outre, on peut déduire des informations sur la direction d'un individu même s'il porte seulement un bat, par l'analyse du motif des récepteurs qui ont détecté les signaux ultrasons de cet émetteur ainsi que la puissance de ces signaux.

I.3.1.7. Radio-identification

La radio-identification connue sous l'acronyme RFID est une méthode d'identification automatique qui permet de mémoriser et de récupérer des données à distance en utilisant des marqueurs appelés «radio-étiquettes».

Un système RFID est composé essentiellement des deux éléments suivants [15]:

- ✚ **Tag RFID**, aussi appelé étiquette radio, marqueur radio, identifiant, transpondeur,... Il s'agit d'un récepteur incorporé sur des objets, des animaux ou mêmes des humains. Chaque «tag» est muni d'une puce qui sert à mémoriser les données. Cette puce est reliée à une antenne qui assure les échanges de ces données. Selon l'intégration ou non d'un émetteur radio, les tags RFID peuvent être classés en deux catégories :
 - **Tag RFID passif:** ce type de tag n'intègre pas d'émetteur radio. Ne comportant pas une source d'énergie, le tag passif utilise souvent l'onde radio reçue de l'interrogateur afin d'alimenter son circuit électronique. Ce type tag est utilisé dans plusieurs applications de capture simple de données sans nécessité de les émettre (capture de température, etc.).
 - **Tag RFID actif:** ce tag comporte un émetteur radio. Il embarque généralement une source d'alimentation en énergie (batterie, pile..). Cette source d'énergie

assure la fonction diffusion du signal vers un lecteur RFID. Ce type de tag est largement utilisé dans des applications de localisation de personnes, de véhicules et la traçabilité d'objets de tout genre.

- **Interrogateurs RFID**, ont comme vocation l'activation des tags RFID en leur fournissant l'énergie dont ils ont besoin.

I.3.1.8. La radio fréquence

Les radiofréquences sont utilisées dans de nombreux domaines, mais dans le contexte des technologies de l'information et des communications, il s'agit de la bande de fréquence à laquelle les signaux de télécommunication sans fil sont transmis et diffusés. La bande de fréquences est divisée en différentes parties, qui sont ensuite attribuées à différentes industries technologiques. Ceci est connu sous le nom de spectre radio. Par exemple, la bande VHF (très haute fréquence), qui va de 30 à 300 MHz, est utilisée pour la radio FM, les émissions de télévision, la radio amateur et ses homologues. Pour beaucoup d'appareils de communication électroniques, la bande ultra haute fréquence (UHF) est utilisée. Il s'agit de l'espace utilisé par les téléphones mobiles, le réseau local sans fil, Bluetooth, la télévision et la radio terrestre. Pour établir une telle liaison il faut un émetteur et un récepteur. L'émetteur reçoit un signal électrique qu'il convertit en signal électromagnétique, lequel est conduit vers l'antenne. L'antenne émet le signal dans la nature où il peut être capté par une ou plusieurs autres antennes. Ceci constitue une différence fondamentale avec les liaisons filaires pour lesquelles un signal ne peut être reçu que par celui qui se trouve au bout du fil. Partout et est donc susceptible.

L'antenne réceptrice conduit le signal reçu vers le récepteur d'où est extrait le même signal électrique que celui qui était en entrée de l'émetteur.



Figure I. 6. Principe d'émission – réception

I.3.2. Capteurs

Les capteurs exploités dans la localisation sont principalement les systèmes micro électromécaniques (MEMS) et la caméra avec ses différents types : perspective, omnidirectionnelle et fish-eye.

I.3.2.1. Systèmes micro électromécaniques

Connus aussi avec l'acronyme anglais MEMS, ils sont caractérisés par leur structure micrométrique et assurent la fonction de capteur et/ou d'actionneur. Les capteurs qui utilisent cette technologie sont généralement les accéléromètres, les magnétomètres, les gyroscopes ou les baromètres. Les MEMS sont indépendants de toute infrastructure et ils sont fonctionnels dans tout environnement que ce soit à l'intérieur ou bien à l'extérieur. Cependant, leur inconvénient majeur c'est que leur performance est affectée par de nombreuses erreurs (biais et bruit) qui s'accumulent rapidement au cours du temps. Pour cette raison, ils sont généralement couplés avec d'autres technologies pour assurer une localisation fiable.

I.3.2.2. Caméra

C'est un appareil électronique dont sa fonction principale est la prise de vues et/ou de séquences vidéo, mais qui peut être aussi exploitée pour assurer la localisation indoor: grâce à une caméra embarquée et une cartographie, il est possible d'obtenir une localisation et une orientation instantanées et précises du porteur le long d'un itinéraire, à quelques dizaines de centimètres ou de degrés près. La cartographie peut être préenregistrée formant ainsi une mémoire visuelle de l'environnement comme elle peut être construite au fur et à mesure de l'avancement du porteur de la caméra l'environnement. Cette technique s'appelle la localisation par vision artificielle.

I.4. Conclusion

Nous avons présenté dans ce chapitre une étude bibliographique qui a introduite les techniques et les technologies de localisation indoor. L'idée est alors de proposer un système de localisation indoor en utilisant les techniques d'intelligence artificielle (IA). Avant de parler sur l'IA, nous vous présentons d'abord la carte Raspberry Pi dans le chapitre suivant, cette carte est le principal élément hardware utilisé dans notre système de localisation.

Chapitre II

Traitement d'images sur Raspberry Pi

II.1. Introduction

Les cartes de prototypage rapide (Arduino, Raspberry Pi...) sont des outils faciles à mettre en œuvre pour le “grand public” et deviennent des outils indispensables au développement de systèmes embarqués simples.

La carte Raspberry Pi est vite devenue un phénomène mondial. Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry *Pi* est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles.

Le chapitre présent est divisé en deux parties, dans la première partie nous présentons brièvement la carte Raspberry *Pi*, dans la deuxième partie, nous présentons les méthodes de segmentation utilisé dans notre travail.

II.2. Le Raspberry *Pi*

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (Matlab, python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry *Pi* est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles.

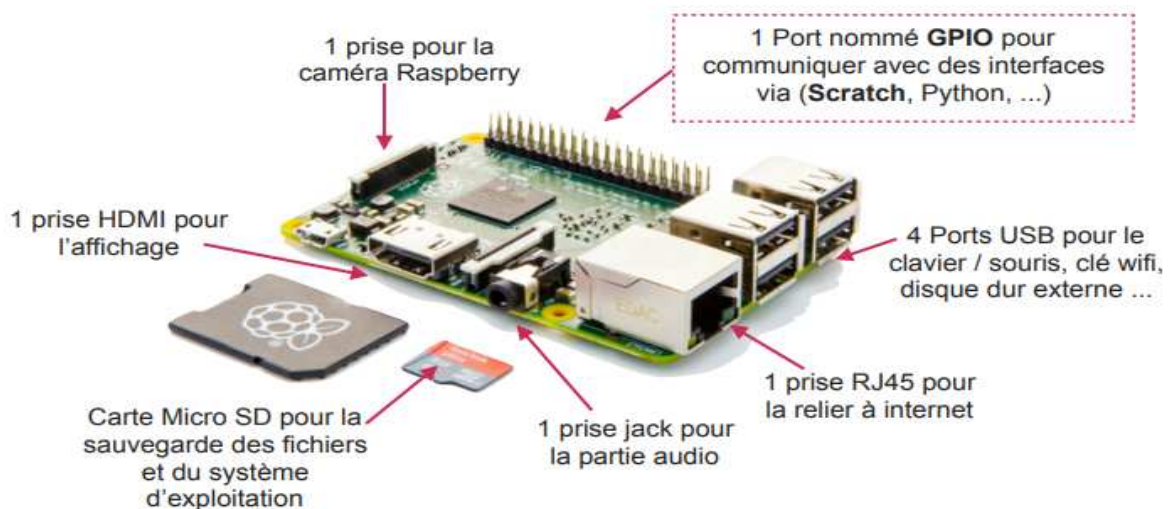
II.2.1. Raspberry *Pi* 3

La carte Raspberry Pi 3 est une ordinateur mono-carte pouvant se connecter à un téléviseur, à un clavier et disposant d'une connectivité Wi-Fi et Bluetooth, possède 4 ports USB, un port micro-SD, un connecteur d'E/S 40 broches et un port HDMI. Cette version est physiquement identique à la version Pi 2 la rendant compatible avec les boîtiers et autres accessoires de la version *Pi* 2. La version 3 est basée sur un processeur ARM Cortex-A53 64 bits quatre cœurs à 1,2 GHz (environ 10x plus rapide que le *Pi* 1 et 50% plus performante que le modèle *Pi* 2) et possède 1 GB de mémoire RAM.

Le Raspberry *Pi* peut effectuer des tâches d'un PC de bureau (feuilles de calcul, traitement de texte, jeux). Il peut également diffuser des vidéos en haute définition grâce à son circuit Broadcom Video Core IV (permet le décodage des flux Blu-ray full HD).

Caractéristiques:

- CPU: Processeur SoC de noyau de Cortex-A53 de quadruple BCM2837 64bit de Broadcom fonctionnant à 1.2GHz
- Mémoire: 1 Go de RAM LPDDR2
- 4 ports USB2.0 avec une sortie jusqu'à 1.2A
- En-tête GPIO étendu à 40 broches
- Sortie vidéo / audio via un connecteur 3,5 mm 4 broches, HDMI, caméra CSI ou LCD brut (DSI)
- Stockage: Support pour cartes micro-SD
- Port Ethernet 10/100 (RJ45)
- WiFi: 2,4 GHz, 802.11n (Broadcom BCM43438)
- Bluetooth 4.1 (Broadcom BCM43438)
- Périphériques de bas niveau:
 - 27 x GPIO
 - UART
 - Bus I2C
 - Bus SPI avec deux choix de puces
 - + 3,3 V
 - + 5V
 - GND
- Alimentation : 5 Vcc/maxi 2.5 A * via prise micro-USB (* intensité maxi si toutes les fonctions sont utilisées)
- Bus: SPI, I2C, série.

**Figure II. 1.** Raspberry Pi 3.

II.2.2. Les Ports GPIO

Le Raspberry *Pi* possède, en plus des connectiques classiques USB, HDMI, etc... un connecteur GPIO. GPIO signifie en anglais «General Purpose Input Output» et pourrait être traduit en français par entrées/sorties numériques. Ces entrées/sorties permettent d'étendre les fonctionnalités du Raspberry *Pi* en lui donnant la possibilité d'agir sur des LEDs ou des afficheurs LCD par exemple, lire l'état d'un interrupteur, d'un capteur, etc... Ce connecteur GPIO dispose de différents types de connexion:

- Des broches utilisables en entrée ou sortie numérique tout ou rien.
- Des broches pour une interface I2C.
- Une interface SPI.
- Les broches *Rx* et *Tx* pour la communication UART avec les périphériques séries.
- Des broches pouvant être utilisé en PWM ("Pulse Width Modulation") permettant le contrôle de puissance ou PPM.

PIN #	NAME	NAME	PIN #
	3.3 VDC Power	5.0 VDC Power	
8	SDA0 (I2C)	DNC	
9	SCL0 (I2C)	DV (Ground)	
7	GPIO 7	TxD (UART)	15
	DNC	RxD (UART)	16
0	GPIO 0	GPIO1	1
2	GPIO2	DNC	
3	GPIO3	GPIO4	4
	DNC	GPIO5	5
12	MOSI	DNC	
13	MISO	GPIO6	6
14	SCLK	CE0	10
	DNC	CE1	11

Figure II. 2. Pins GPIO et Leur Fonctions.

II.2.3. Aspects Logiciel

Toutes les applications compatibles avec le système d'exploitation et le processeur ARM, ou utilisant un environnement d'exécution virtuel (Java, émulateurs...) sont susceptibles de fonctionner : Python, MATLAB,... Les principales contraintes portent sur les performances du processeur et la mémoire vive disponible (256 Mo). Ce dernier point a toutefois été corrigé avec l'arrivée des versions embarquant 512 Mo de mémoire vive ce qui est suffisant pour lancer le processus du contrôle et le serveur web sur le système d'exploitation Raspbian.

La plupart des systèmes qui fonctionnent sur Raspberry *Pi* sont des versions du système d'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter pour des buts spécifiques. La distribution recommandée est Raspbian. C'est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry *Pi*. Cependant

Raspbian fournit plus qu'un simple système d'exploitation : il est livré avec plus de 35.000 paquets, des logiciels précompilés qui nous facilitent le développement soft. Il y a autre système d'exploitation créé par Microsoft spécialement pour le Windows 10 IoT qui peut être installé sur notre carte Raspberry *Pi* 3.



Figure II. 3. Logo du système d'exploitation Raspbian et Windows 10 IoT.

II.3. Accès à distance au Raspberry *Pi*

On peut accéder à la carte Raspberry *Pi* via le protocole SSH et/ou VNC, si on connaît l'adresse IP de la carte.

II.3.1. Choix de l'adresse IP du Raspberry *Pi*

Pour trouver l'adresse IP de notre Raspberry *Pi*, nous avons utilisé Advanced IP Scanner qui est un scanner de réseau local utilisé sous Windows. La figure suivante montre l'interface de cet outil et l'information fournie par ce dernier concernant le Raspberry *Pi* :

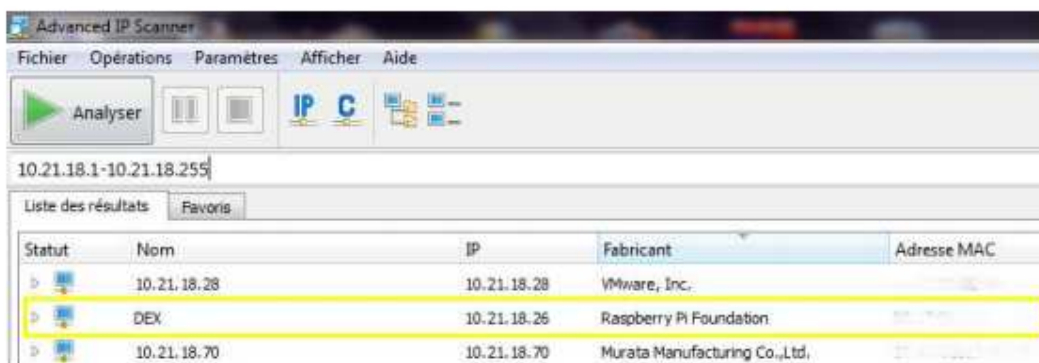


Figure II. 4. Adresse IP de la carte Raspberry *Pi*.

Après que nous avons trouvé l'adresse IP, il faut la fixer afin de nous permettre l'accès à distance au Raspberry *Pi*, pour cela on va suivre les étapes suivantes :

Configuration réseau sous « Raspbian For Robots ». Par défaut, il contient les valeurs suivantes :

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet manual wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

La signification de chaque ligne de ce fichier est la suivante :

- auto lo : va démarrer l'interface automatiquement lors de la séquence de boot.
- Iface lo inet loopback : définition de l'interface loopback (interface virtuelle indispensable qui permet de supprimer les paquets erronés)
- iface eth0 inet dhcp : l'interface eth0 (le port RJ45 du Raspberry) sera configurée en envoyant une requête DHCP sur le réseau. Donc IP dynamique.

Les quatre lignes suivantes permettent de configurer le Wi-Fi via le fichier de configuration de wpa_supplicant.

Cette configuration permet de placer une adresse IP statique pour cela nous allons changer l'interface Ethernet (eth0) en mettant à jour la configuration de l'interface : IP, masque et passerelle comme indique la figure suivante :

```
root@dex:/etc# cd network/
root@dex:/etc/network# ls
if-down.d      if-pre-up.d  interfaces    run
if-post-down.d if-up.d      interfaces.bkp
root@dex:/etc/network# cat interfaces
auto lo

iface lo inet loopback
iface eth0 inet static
address 10.21.18.26
netmask 255.255.248.0
gateway 10.21.16.100
dns-nameservers 8.8.8.8

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface wlan0 inet dhcp
```

Figure II. 5. Fichier de configuration réseau.

La ligne après gateway (passerelle) sert à définir les serveurs de noms de domaine (DNS) pour qu'on puisse se connecter à une ressource réseau externe (convertir un nom alphanumérique « par exemple www. Google.com » en une adresse réseau numérique « 172.217.19.164 »). Le DNS 8.8.8.8 est un DNS Google

Finalement le redémarrage de Raspberry Pi.

Alors à cette étape nous avons donné à notre carte une adresse IP fixe qui va nous servir à la manipulation de la carte à distance via le protocole SSH et ou VNC.

II.3.2. Connexion à distance via SSH

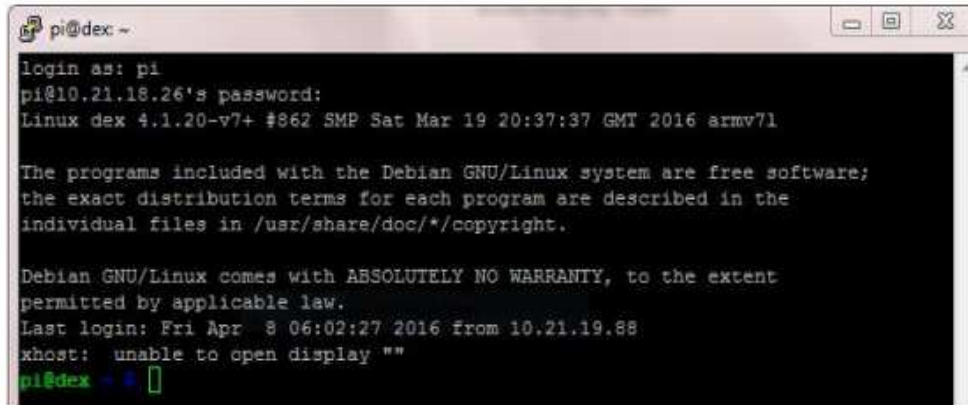
Secure Shell est un protocole de réseau crypté pour initier des sessions Shell textuelles sur des machines distantes de manière sécurisée. Cela permet à un utilisateur d'exécuter des commandes sur l'invite de commande d'une machine sans qu'ils soient physiquement présents à proximité de la machine. SSH a été créé en 1995 pour le principal but est de permettre la prise de contrôle à distance d'une machine à travers une interface en lignes de commande.

Pour faire le contrôle à distance sur le Raspberry Pi on doit installer le SSH côté d'un serveur SSH sur notre Raspberry Pi et d'un autre d'un client SSH sur notre ordinateur, mais heureusement le serveur SSH est déjà installé et activé par défaut sur la Raspberry Pi, ce qui reste seulement d'installer un client SSH sur notre ordinateur ; pour cela on va installer le logiciel Putty sous Windows qui est un client SSH.



Figure II. 6. Logiciel Putty.

Après son installation on entre l'adresse IP de la carte puis on clique sur «Open», et après l'authentification (en entrant le nom d'utilisateur pi et notre mot de passe), on se connectera directement sur l'invite des commandes de notre système d'exploitation Raspbian For robots comme l'indique la figure suivante:



```
pi@dex ~$ ssh pi@10.21.18.26
login as: pi
pi@10.21.18.26's password:
Linux dex 4.1.20-v7+ #862 SMP Sat Mar 19 20:37:37 GMT 2016 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 8 06:02:27 2016 from 10.21.19.88
xhost:  unable to open display ""
pi@dex ~$
```

Figure II. 7. Invite de commande via SSH.

II.4. Raspberry Pi basé sur des langages de programmations

La programmation d'une carte Raspberry *Pi* implique souvent de travailler avec des images, de la vidéo, de l'audio et autres données de capteurs.

MATLAB et Simulink permettent aux utilisateurs d'analyser et visualiser rapidement ces données et de programmer leur carte Raspberry *Pi* en conséquence. Les produits prennent en charge deux méthodes de travail [16]:

- ❖ Lecture, écriture et analyse des données à partir des capteurs et caméras Raspberry *Pi*.
- ❖ Développement d'algorithmes s'exécutant de manière autonome sur la carte Raspberry *Pi*.

II.4.1. Analyse des données à partir la carte Raspberry *Pi*

Le support package MATLAB pour Raspberry *Pi* permet d'écrire des programmes MATLAB qui communiquent avec la carte Raspberry *Pi* et acquièrent des données à partir des ports GPIO de la carte, des caméras et autres appareils connectés. MATLAB étant un langage interprété de haut niveau, il permet de facilement prototyper et affiner des algorithmes pour notre projet basée sur le Raspberry *Pi*. MATLAB inclut des milliers de fonctions mathématiques et de tracés que nous pouvez utiliser dans le cadre de la programmation de Raspberry *Pi*, couvrant des domaines tels que le traitement vidéo et le traitement d'image, l'optimisation, les statistiques et le traitement du signal.

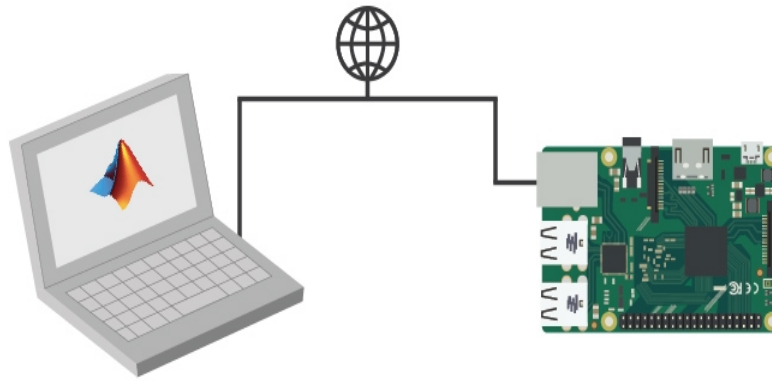


Figure II. 8. Connexion ordinateur et Raspberry *Pi*.

Avec le support package MATLAB pour Raspberry *Pi*, la carte Raspberry *Pi* est connectée à un ordinateur exécutant MATLAB. Le traitement est effectué sur l'ordinateur avec MATLAB.

II.4.2. Programmation Raspberry *Pi*

L'utilisation de langage de programmation MATLAB dans la carte Raspberry *Pi* permet d'effectuer les opérations suivantes :

- ❖ Analyse des données du capteur Raspberry *Pi* à l'aide de milliers de fonctions pré-intégrées pour le traitement d'image, le traitement du signal, la modélisation mathématique, etc.
- ❖ Visualisation rapide des données grâce au vaste choix de tracés MATLAB.
- ❖ Utilisation du même logiciel pour programmer d'autres cartes matérielles, notamment Arduino.

II.4.3. Développer des algorithmes Simulink

Le support package Simulink pour Raspberry *Pi* permet de développer des algorithmes dans Simulink, un environnement de programmation par blocs pour la modélisation de systèmes dynamiques et le développement d'algorithmes, avec une exécution autonome sur la carte Raspberry *Pi*. Le support package intègre des blocs Simulink pour configurer la carte Raspberry *Pi*, envoyer et recevoir des paquets UDP et lire et écrire des données à partir des capteurs.

Une fois le modèle Simulink créé, nous pouvons le simuler, affiner les paramètres de notre algorithme pour arriver au bon réglage, et télécharger l'algorithme final pour une exécution autonome sur la carte. Utilisez le bloc MATLAB Function pour incorporer du code MATLAB dans un modèle Simulink.

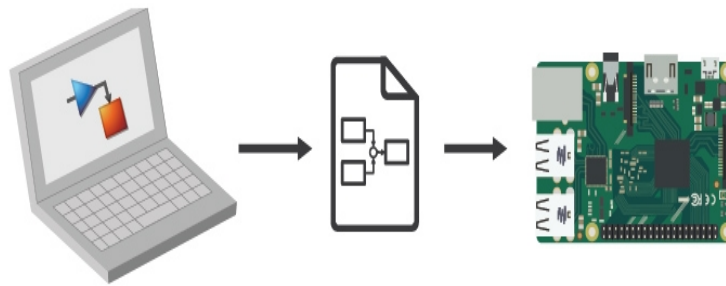


Figure II. 9. L'environnement de programmation pour Raspberry *Pi* par l'intermédiaire du support package Simulink.

Avec le support package Simulink pour Raspberry *Pi*, on peut développer des algorithmes dans Simulink et le déployer sur la carte Raspberry *Pi* à l'aide de la génération de code automatique. Le traitement est alors effectué sur la carte Raspberry *Pi*.

Utiliser Simulink dans le cadre de la programmation Raspberry *Pi* permet d'effectuer les opérations suivantes :

- ❖ Développement et simulation des algorithmes dans Simulink et utilisation de la génération de code automatique pour exécuter les algorithmes sur la carte.
- ❖ Incorporation du traitement du signal, de la conception de contrôleurs, de la logique de contrôle, et d'autres programmes avancés pour les mathématiques et l'ingénierie dans les projets de programmation Raspberry *Pi*.
- ❖ Réglage et optimisation interactive des paramètres durant l'exécution des algorithmes sur la carte Raspberry *Pi*.

II.5. Segmentation utilisant Raspberry Pi

La caméra prend des photos à intervalles de temps réguliers (figure II.10). On effectue alors une segmentation, qui permet d'obtenir une image ne contenant que du blanc et du noir (Objet/ arrière-plan). On définit pour cela un seuil et on compare la luminance de chaque pixel à ce seuil. Dans un cas on remplace ce pixel par du blanc, dans l'autre par du noir. L'image binaire obtenue est noire aux endroits qui n'ont pas d'objet, et blanche où il y a eu un objet.



Figure II. 10. Raspberry Pi 3 et camera.

Les méthodes de segmentation sont très nombreuses, elles constituent la grande majorité des méthodes de seuillage et sont généralement basées sur le seuillage. Nous présenterons dans ce qui suit quatre catégories de segmentation: méthodes de seuillage global, les méthodes basées sur l'entropie, les méthodes basées sur la similarité, et les méthodes basées sur la forme de l'histogramme. Dans ce travail, nous intéressons par les trois dernières méthodes.

II.5.1. Binarisation basée sur le seuillage

Notons I l'image en niveaux de gris dont les intensités varient de 0 (noir) à 255 (blanc), et I_b l'image binaire résultante. $I(x,y)$ et $I_b(x,y)$ dénotent l'intensité du pixel (x,y) dans l'image I et I_b respectivement.

II.5.1.1. Méthodes de seuillage global

Les méthodes de seuillage global calculent un seul seuil global TH pour séparer l'image en deux classes Objet et arrière-plan. Ainsi, le seuillage s'effectue comme suit :

$$I_b(x, y) = \begin{cases} 255, & \text{si } I(x, y) > TH \\ 0, & \text{si } I(x, y) \leq TH \end{cases} \quad (\text{II.1})$$

II.5.1.2. Méthodes basées sur l'entropie

Ces méthodes exploitent l'entropie de distribution des niveaux de gris pour l'estimation d'un seuil global de binarisation.

Notons $\mu_1(TH)$ et $\mu_2(TH)$ les moyennes des deux classes objet et arrière-plan en fonction du niveau de gris TH . $L=256$ est l'intensité maximal de l'image.

$$\mu_1(TH) = \frac{\sum_{i=0}^{TH} i \times h_i}{\sum_{i=0}^{TH} h_i} \quad (\text{II.2})$$

$$\mu_2(TH) = \frac{\sum_{i=TH+1}^{L-1} i \times h_i}{\sum_{i=TH+1}^{L-1} h_i} \quad (\text{II.3})$$

II.5.1.2.1. Méthode de Kapur

La méthode de Kapur [18], ou méthode du seuillage entropique, est basée sur le principe de la maximisation de l'entropie totale de Shannon. La méthode suppose que deux objets d'une même image ont des densités de probabilité indépendantes.

La méthode prend en compte la distribution de probabilité de l'objet P_1 et la distribution de probabilité d'arrière-plan P_2 dans la détermination de l'entropie. Le seuil optimal est celui qui permet de maximiser l'entropie de l'image partitionnée. Dans le cas de deux classes, les entropies partielles et totales sont données par :

$$H_1(TH) = - \sum_{i=0}^{TH} \frac{P_i}{P_1} \log_2 \left(\frac{P_i}{P_1} \right) \quad (\text{II.4})$$

$$H_2(TH) = - \sum_{i=TH+1}^{L-1} \frac{P_i}{P_2} \log_2 \left(\frac{P_i}{P_2} \right) \quad (\text{II.5})$$

$$H_T(TH) = H_1(TH) + H_2(TH) \quad (\text{II.6})$$

Notons $p_i=p(i)$, la probabilité d'occurrence du niveau de gris i dans l'image.

Le problème de seuillage entropique est alors défini par :

$$TH^* = \text{Arg max}(H_T(TH)) \quad (\text{II.7})$$

II.5.1.2.2. Méthode de seuillage par entropie croisée (Li et Lee)

Li et Lee ont proposé une méthode de seuillage où le regroupement des niveaux de gris en deux classes (objet et arrière-plan) est fondé sur la minimisation de l'entropie croisée [19]. Le seuil optimal TH^* est calculé de façon à minimiser:

$$\eta_T(TH) = \eta_1(TH) + \eta_2(TH) \quad (\text{II.8})$$

où :

$$\eta_1(TH) = \sum_{i=0}^{TH} i \times h_i \times \log_2 \left(\frac{i}{\mu_1(TH)} \right) \quad (\text{II.9})$$

$$\eta_1(TH) = \sum_{i=TH+1}^{L-1} i \times h_i \times \log_2 \left(\frac{i}{\mu_2(TH)} \right) \quad (\text{II.10})$$

II.5.1.2.3. Méthode de Yen et al.

Considérant objet et l'arrière-plan de l'image comme deux sources de signaux différents, Yen et al. [20] choisissent le seuil comme le niveau de gris qui maximise la corrélation entropique (H_C). Cette dernière est donnée par :

$$H_c(TH) = H_1(TH) + H_2(TH) = -\log_2 \left(\sum_{i=0}^{TH} \left(\frac{P_i}{P_1} \right)^2 \right) - \log_2 \left(\sum_{i=TH+1}^{L-1} \left(\frac{P_i}{P_2} \right)^2 \right) \quad (\text{II.11})$$

II.5.1.3. Méthodes basées sur les méthodes de similarités

Les méthodes de cette classe choisissent la valeur du seuil en se basant sur des mesures de similarité entre l'image originale et sa l'image binarisée.

II.5.1.3.1. Méthode de Tsai et Lin

Tsai [21] a considéré l'image en niveaux de gris comme une version floue d'une image binaire idéale. Il propose ainsi de déterminer le seuil optimal comme le niveau de gris avec lequel l'image binarisée préserve le mieux les trois moments mathématiques de l'image en niveaux de gris. Ainsi, le seuil optimal TH^* est le niveau de gris qui garantie :

$$m_1 = b_1(TH), m_2 = b_2(TH), m_3 = b_3(TH), \quad (\text{II.12})$$

Tel que m_k est le $k^{\text{ème}}$ moment de l'image en niveaux de gris et b_k le $k^{\text{ème}}$ moment de l'image binaire, et sont définis par :

$$\begin{cases} m_k(TH) = \sum_{i=0}^{L-1} p_i \times i^k \\ b_k(TH) = P_1 \times m_1^k + P_2 \times m_2^k \end{cases} \quad (\text{II.13})$$

$$\begin{cases} m_1^k = \sum_{i=0}^{TH} p_i \times i^k \\ m_2^k = \sum_{i=TH+1}^{L-1} p_i \times i^k \end{cases} \quad (\text{II.14})$$

II.5.1.3.2. Méthode de Huang et Wang

Cette méthode basée sur la minimisation d'une mesure de floue [22]. La mesure de floue de l'image entière peut être obtenue par l'entropie de Shannon ou par la mesure de Yager. L'entropie de Shannon pour une image de taille $M \times N$ est définie comme suit :

$$H = \frac{1}{MN \log_2(2)} \sum_{i=0}^{L-1} S(\varphi(i)) \times h_i \quad (\text{II.15})$$

Avec :

$$S(TH) = \sum_{i=0}^{TH} h_i \quad (\text{II.16})$$

$\varphi(i)$: la fonction d'appartenance d'un pixel ayant le niveau de gris i et elle est donnée comme suit :

$$\varphi(i) = \begin{cases} \frac{1}{1 + \frac{|i - \mu_1|}{C}} & \text{si } i \leq TH \\ \frac{1}{1 + \frac{|i - \mu_2|}{C}} & \text{si } i > TH \end{cases} \quad (\text{II.17})$$

Où C : la distance entre le niveau de gris minimum et le niveau de gris maximum dans l'image.

II.5.1.3.3. Méthode de Hertz et Schafer

Dans cette méthode, le seuil global est choisi comme le niveau de gris maximisant la similarité entre les contours obtenus à partir de l'image originale et les contours dérivés de l'image binarisée [23]. Les contours des deux images sont obtenus en utilisant un opérateur de Sobel.

II.5.1.4. Méthodes basées sur la forme de l'histogramme

Cette catégorie de méthodes effectue un seuillage en se basant sur les propriétés de la forme de l'histogramme.

II.5.1.4.1. Méthode d'Otsu

Elle est considérée comme la méthode de référence dans le domaine du seuillage d'histogrammes. Dans cette méthode [24], l'opération de seuillage est vue comme une séparation des pixels d'une image en deux classes C_1 (arrière plan), C_2 (objet) à partir d'un seuil TH . La classe C_1 regroupe tous les pixels ayant un niveau de gris inférieur au seuil TH alors que la classe C_2 contient tous les pixels de niveau de gris supérieur à TH . Ces deux classes peuvent être désignées en fonction du seuil TH comme suit :

$$\begin{cases} C_1 = \{0, 1, \dots, TH\} \\ C_2 = \{TH + 1, \dots, L - 1\} \end{cases} \quad (\text{II.18})$$

$$\begin{cases} \sigma_B^2 = P_1 P_2 (\mu_2 - \mu_1)^2 \\ \sigma_T^2 = \sum_{i=0}^{L-1} p_i (i - \mu)^2 \\ \sigma_w^2 = \sum_{i=0}^{TH} P_1 (i - \mu_1)^2 + \sum_{i=TH+1}^{L-1} P_2 (i - \mu_2)^2 \end{cases} \quad (\text{II.19})$$

Soient σ_B^2 est la variance d'interclasse, σ_T^2 la variance totale, et σ_w^2 la variance d'une classe, p_i la probabilité de pixel i .

$$\left\{ \begin{array}{l} \sigma_T^2 = \sigma_B^2 + \sigma_w^2 \\ \mu_1 = \sum_{i=1}^{L-1} i \cdot \frac{p_i}{P_1} \\ \mu_2 = \sum_{i=1}^{L-1} i \cdot \frac{p_i}{P_2} \\ \mu = \sum_{i=1}^{L-1} i \cdot p_i \end{array} \right. \quad (\text{II.20})$$

μ_1 , μ_2 , μ désignent respectivement les niveau de gris moyen des classes C_1 , C_2 et de l'image.

P_1 et P_2 représentent respectivement les probabilités des classes C_1 , C_2 tels que :

$$\left\{ \begin{array}{l} P_1 = \sum_{i=0}^{TH} p_i \\ P_2 = \sum_{i=TH+1}^{TH} p_i \\ P_1 + P_2 = 1 \end{array} \right. \quad (\text{II.21})$$

Le seuil optimum TH^* peut être déterminé en maximisant un des trois critères suivantes :

$$\left\{ \begin{array}{l} TH^* = \arg \max \left(\frac{\sigma_B^2}{\sigma_w^2} \right) \\ TH^* = \arg \max \left(\frac{\sigma_B^2}{\sigma_T^2} \right) \\ TH^* = \arg \max \left(\frac{\sigma_T^2}{\sigma_w^2} \right) \end{array} \right. \quad (\text{II.22})$$

II.5.1.4.2. Méthode de Ridler et Calvard

Cette méthode [25] est basée sur une mixture de gaussienne de deux classes. A la $n^{\text{ième}}$ itération, un nouveau seuil TH_n est établi en utilisant la moyenne de l'objet (μ_1) et la moyenne de l'arrière-plan (μ_2):

$$TH^* = \lim_{n \rightarrow \infty} TH_n \quad (\text{II.23})$$

$$TH_{n+1} = \frac{\mu_1(TH_n) + \mu_2(TH_n)}{2} \quad (\text{II.24})$$

II.5.1.4.3. Méthode de Lloyd

Lloyd [26] a supposé que l'image est caractérisée par deux fonctions de densité gaussienne à variance égale. Le seuil de segmentation est choisi comme le niveau de gris qui minimise la fonction η .

$$\eta = \frac{\mu_1(TH) + \mu_2(TH)}{2} + \frac{\sigma^2}{\mu_1(TH) - \mu_2(TH)} \log_2 \frac{P_2(TH)}{P_1(TH)} \quad (\text{II.25})$$

Où σ^2 est la variance de l'image entière.

II.6. Conclusion

Nous avons présenté dans ce chapitre les composants de la carte Raspberry Pi, et nous avons expliqué les portes GPIO et l'aspect logiciel de l'adresse IP du Raspberry Pi 3, pour utiliser ce dernier à distance avec une application via SSH.

Ensuite, nous avons présenté quelques méthodes sur la segmentation d'images. Les méthodes de segmentation basées sur le seuillage sont nombreuses, nous avons détaillé que quelques une d'entre elles, celles qui seront utilisées pour la détection des marqueurs, et ensuite l'étape d'identification ou classification des marqueurs en utilisant les méthodes d'apprentissage profond (deep learning) pour l'objectif de la localisation dans un environnement intérieur. L'apprentissage profond est étudié dans le prochain chapitre.

Chapitre III

Apprentissage profond (deep learning)

III.1. Introduction

Afin de bénéficier des avantages des applications d'intelligence artificielle nous voulons réaliser une plateforme qui utilise l'apprentissage automatique pour prédire la localisation intérieure. Cela est dû à la limite des GPS dans les environnements intérieurs à savoir la mauvaise pénétration de ses signaux à l'intérieur des immeubles, ainsi le cout considérablement cher de l'intégration des récepteur GPS dans chaque objet.

Dans ce chapitre nous allons présenter les méthodes de l'apprentissage profond, et exposer les méthodes les plus utilisées pour l'apprentissage de ce type de réseaux qui nous permettront d'entamer notre étude.

III.2. Définition de l'apprentissage profond (deep learning)

Le Deep Learning est un nouveau domaine de recherche du ML (machine learning), qui a été introduit dans le but de rapprocher le ML (machine learning) de son objectif principal: l'intelligence artificielle (Figure III.1). Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

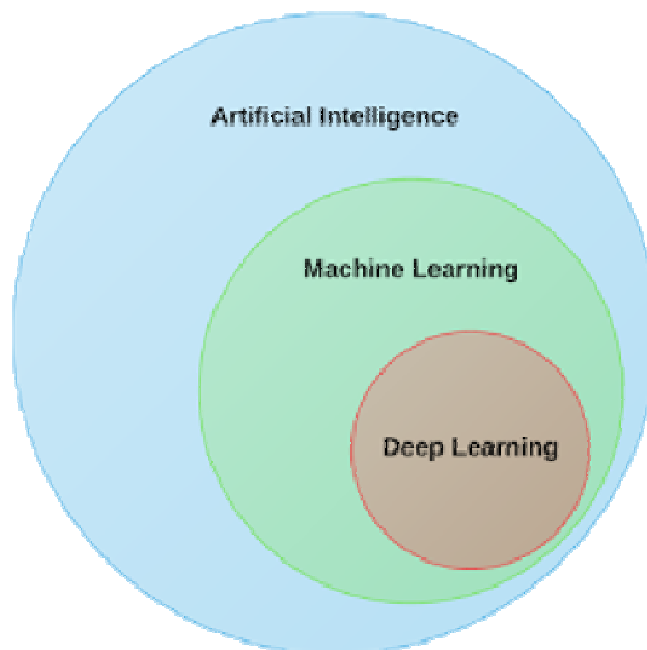


Figure III. 1. La relation entre l'intelligence artificielle, le ML et le deep learning.

L'apprentissage profond «deep learning» est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années.

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données.

L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des «réseaux de neurones artificiels», composés de milliers de neurones qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de neurones servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

III.3. Architectures de réseaux de neurones profonds

Il existe un grand nombre de variables d'architectures profondes. La plupart d'entre eux sont dérivés de certaines architectures originales. Nous allons choisir les réseaux de neurones convolutifs (CNNs).

III.3.1. Principe d'architecture d'un CNN

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images (Figure III.2). Une des grandes différences entre le Deep Learning et les algorithmes de ML classique c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleurs. Contrairement à plusieurs algorithmes de ML classiques qui possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", il faut faire des méthodes de sélection des échantillons [27], par contre dans les modèles de Deep Learning n'ont pas de telles limitations.

Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML classique l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme [28].

L'architecture générale d'un apprentissage profond est représentée dans la Figure III.3. Une image est passée à travers d'une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la

résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé CNN.

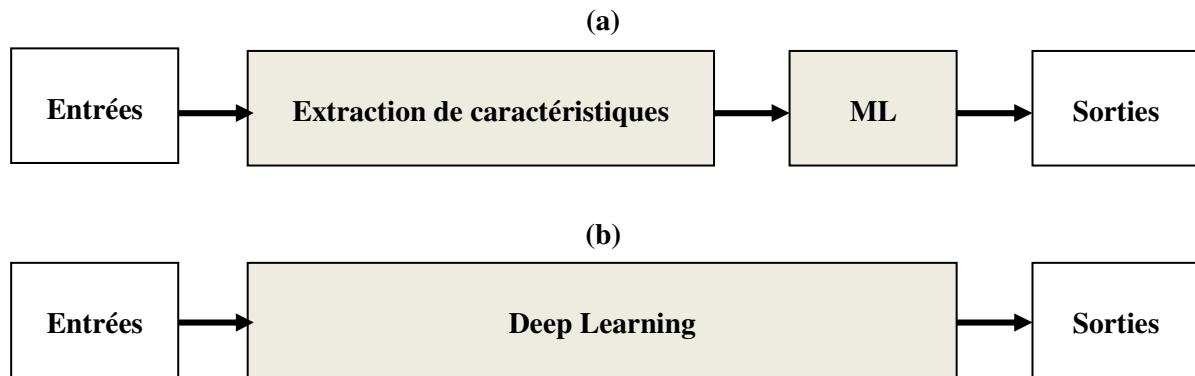


Figure III. 2. Comparaison entre (a) ML classique et (b) deep learning.

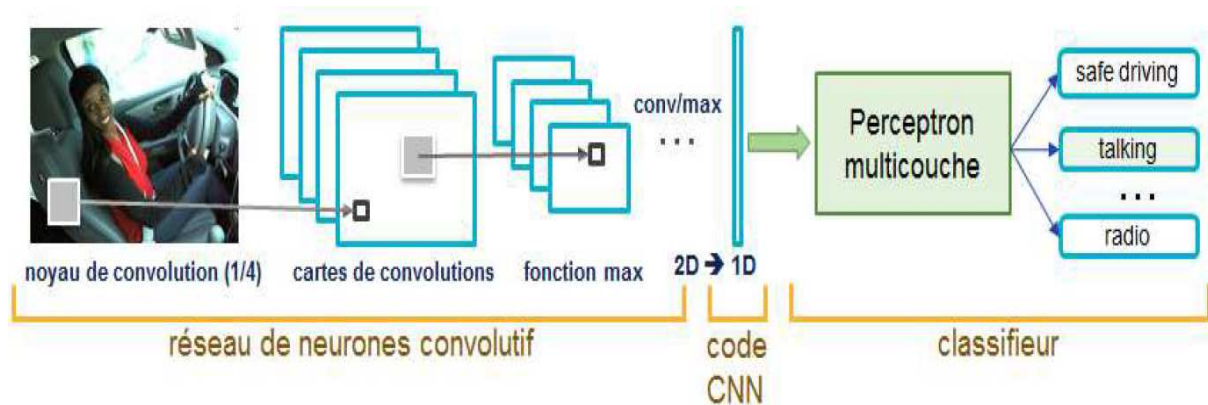


Figure III. 3. Les réseaux de neurones convolutifs (CNN).

III.3.2. Les blocs de construction

Une architecture CNN est formée par un empilement de couches de traitement indépendantes:

III.3.2.1. La couche de convolution (CONV)

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : la 'profondeur', le 'pas' et la 'marge'.

✚ **Profondeur de la couche** : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).

✚ **Le pas** contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

✚ **La marge (à 0) ou 'zeropadding'**: Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

III.3.2.2. Couche de pooling (POOL)

Un autre concept important des CNNs est le pooling « mise en commun », ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN pour contrôler l'overfitting (sur-apprentissage). L'opération de pooling crée aussi une forme d'invariance par translation.

La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne uniquement au niveau de la surface. La forme la plus courante est une couche de mise en commun avec des tuiles de taille 2×2 (largeur/hauteur) et comme valeur de sortie la valeur maximale en entrée. On parle dans ce cas de «Max-Pool 2×2 » (compression d'un facteur 4).

Il est possible d'utiliser d'autres fonctions de pooling que le maximum. On peut utiliser un «average pooling» (la sortie est la moyenne des valeurs du patch d'entrée), du «L2-norm pooling». Dans les faits, même si initialement l'average pooling était souvent utilisé il s'est avéré que le max-pooling était plus efficace car celui-ci augmente plus significativement l'importance des activations fortes. En d'autres circonstances, on pourra utiliser un pooling stochastique.

Le pooling permet de gros gains en puissance de calcul. Cependant, en raison de la réduction agressive de la taille de la représentation (et donc de la perte d'information associée), la tendance actuelle est d'utiliser de petits filtres (type 2×2) (figure III.4). Il est aussi possible d'éviter la couche de pooling, mais cela implique un risque sur-apprentissage plus important.

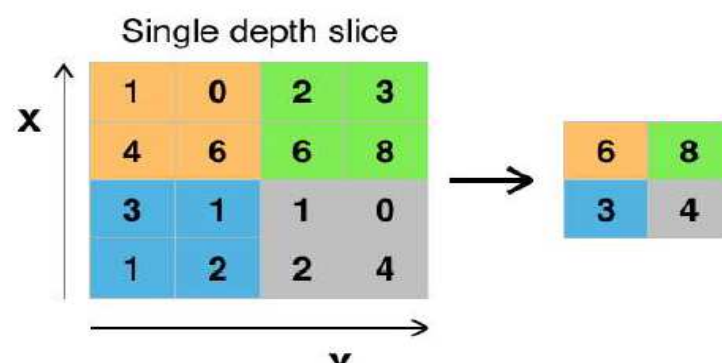


Figure III. 4. Pooling avec un filtre 2×2 et un pas de 2.

III.3.2.3. Couche de correction (ReLU) :

Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. On a notamment:

✚ La correction ReLU (abréviation de Unités Rectifié linéaires):

$$f(x) = \max(0, x) \quad (\text{III.1})$$

L'équation III.1, appelée aussi «fonction d'activation non saturante», augmente les propriétés non linéaires de la fonction de décision et de l'ensemble du réseau sans affecter les champs récepteurs de la couche de convolution.

✚ La correction par tangente hyperbolique :

$$f(x) = \tanh(x) \quad (\text{III.2})$$

✚ La correction par la tangente hyperbolique saturante donnée par l'équation suivante :

$$f(x) = |\tanh(x)| \quad (\text{III.3})$$

✚ La correction par la fonction sigmoïde.

Souvent, la correction Relu est préférable, car il en résulte la formation de réseau neuronal plusieurs fois plus rapide, sans faire une différence significative à la généralisation de précision.

III.3.2.4. Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux réguliers de neurones). Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.

III.3.2.5. Couche de perte (LOSS)

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte «Soft max» est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans $[0,1]$. La perte euclidienne est utilisée pour régresser vers des valeurs réelles [28, 29].

III.3.3. Choix des hyperparamètres

Les CNNs utilisent plus d'hyperparamètres qu'un MLP (multilayer perceptron ou réseaux de neurones multi-couche) standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling.

III.3.3.1. Nombre de filtres

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre.

Le nombre d'images intermédiaires contrôle directement la puissance du système, qui est dépend du nombre d'exemples disponibles et la complexité du traitement.

III.3.3.2. Forme de filtres

Les formes de filtre sont généralement choisies en fonction de l'ensemble de données. Par exemple en choisi une dimension de 28×28 15×15 pixels dans la première couche.

III.3.3.3. Forme de Max pooling

Les valeurs typiques sont 2×2 (Figure 3.4). De très grands volumes d'entrée peuvent justifier un pooling 4×4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.

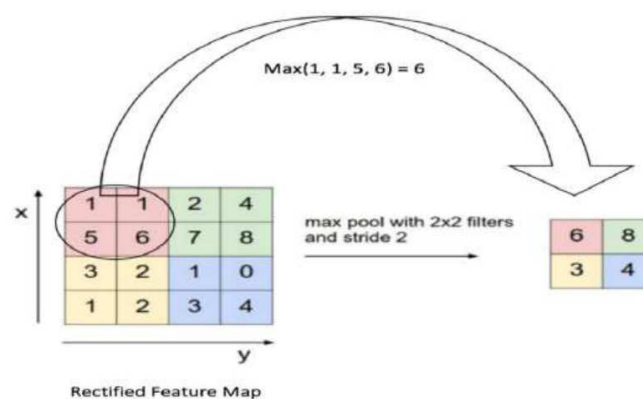


Figure III. 5. Exemple de max pooling (2×2).

III.3.4. Avantages de CNNs

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au MLP, qui lui considère chaque neurone indépendant et donc affecte un poids différent à chaque signal entrant.

Comparés à d'autres algorithmes de classification de l'image, les réseaux de neurones convolutifs utilisent relativement peu de prétraitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

III.3.5. La classification

On a généralement le choix entre deux approches : la classification supervisée et non supervisée.

III.3.5.1. L'objectif de la classification

L'objectif de la classification d'images est d'élaborer un système capable d'affecter un classement automatique d'images. Ainsi, ce système permet d'effectuer une tâche d'expertise qui peut s'avérer coûteuse à acquérir pour un être humain en raison notamment de contraintes physiques comme la concentration, la fatigue et le temps nécessaire pour un volume important de données images.

III.3.5.2. Domaines d'application de la classification

Parmi les champs d'application de cette technologie on trouve : le traitement d'image, vidéo, la voix, et plus spécifiquement de la détection d'objets dans le domaine automobile (marqueur, piétons, panneaux de signalisation, voitures, bus...).

De nombreux secteurs d'activité exploitent des programmes de reconnaissance d'image, les sites marchands pour classer, identifier et trouver des similitudes entre des produits ou des familles de produits. Aussi, ces applications permettent la détection des marqueurs pour la localisation dans les environnements intérieurs, la robotique pour permettre par exemple à un robot-aspirateur de se déplacer en toute autonomie, les réseaux sociaux et autres sites de partage de photos pour la reconnaissance faciale, les moteurs de recherche pour détecter les similitudes entre des images, l'industrie automobile pour le développement des voitures autonomes, la médecine pour la recherche de cellules cancéreuses [30].

III.3.5.3. Les différentes méthodes de la classification et l'apprentissage

III.3.5.3.1. Méthodes supervisées

Dans le cas de l'apprentissage supervisé, on dispose d'un ensemble de données étiquetées, ou d'exemples qui se sont vus associés une classe par un expert. Cet ensemble d'échantillons constitue la base d'apprentissage. Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de construire à partir de la base d'apprentissage, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier, la classe (voir Figure III.6)

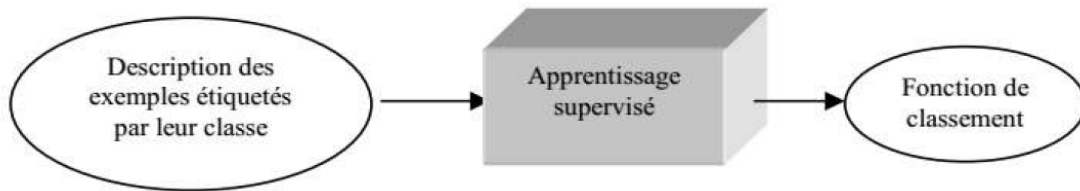


Figure III. 6. L'apprentissage supervisé.

L'inférence inductive est définie comme un processus qui à partir d'une connaissance spécifique observée sur certains objets et d'une hypothèse inductive initiale, permet d'obtenir une assertion inductive impliquant ou rendant compte fortement ou faiblement des observations. Dans le cas de l'apprentissage inductif supervisé, qui est un sous domaine de l'inférence inductive, la connaissance spécifique consiste en un ensemble d'objets appartenant à des classes connues. L'assertion inductive est exprimée par une règle de classification qui assigne une classe à chaque objet. L'implication forte est satisfaite si la règle classe correctement tous les objets connus.

III.3.5.3.2. Méthodes non-supervisées

L'apprentissage non-supervisé, encore appelé apprentissage à partir d'observations ou découverte, consiste à déterminer une classification à partir d'un ensemble d'objets ou de situations données (des exemples non étiquetés).

On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes. Ce type d'apprentissage, encore appelé Cluster.

L'automatisation de la construction de classification constitue aujourd'hui un véritable domaine de recherche. La notion clé utilisée pour créer des classes d'objets est une mesure de la similarité entre les objets. Les classes ou concepts sont construits de façon à maximiser la similarité intraclasse et à minimiser la similarité interclasses.

L'apprentissage non supervisé correspond également à la classification conceptuelle, où une collection d'objets forme une classe si cette classe peut être décrite par un concept, compte tenu d'un ensemble de concepts prédéfinis.

III.4. Les algorithmes d'apprentissage automatique

III.4.1. Le voisin K le plus proche (K-Nearest Neighbours : KNN)

L'algorithme KNN parmi les plus simples algorithmes d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation x , l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x (Figure III.7).

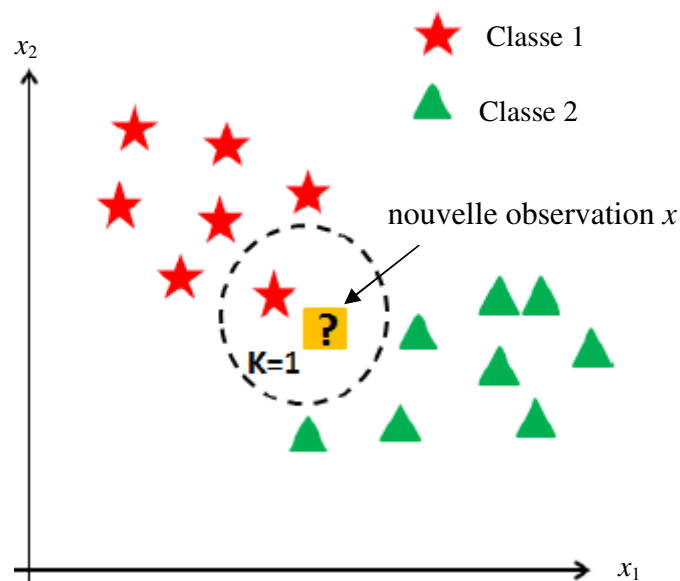


Figure III. 7. La classification avec KNN.

III.4.2. Machine à vecteurs de support

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification et de régression. Les SVM sont une généralisation des classifieurs linéaires.

Les SVM ont été développés dans les années 1990 à partir des considérations théoriques de Vladimir Vapnik [31] sur le développement d'une théorie statistique de l'apprentissage (la Théorie de Vapnik-Chervonenkis). Les SVM ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyper paramètres, leurs garanties théoriques, et leurs bons résultats en pratique.

Les SVM ont été appliqués à de très nombreux domaines (bio-informatique, recherche d'information, vision par ordinateur, finance...) [32]. Selon les données, la performance des

machines à vecteurs de support est de même ordre, ou même supérieure, à celle d'un réseau de neurones ou d'un modèle de mixture gaussienne.

III.4.2.1. Principe général

Cette technique est une méthode de classification à deux classes qui tente de séparer les exemples positifs des exemples négatifs dans l'ensemble des exemples, comme le représente dans la Figure III.8. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Cela garantit une généralisation du principe car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être situés d'un côté ou l'autre de la frontière. L'intérêt de cette méthode est la sélection de vecteurs supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas, ce qui peut être considéré comme un avantage pour cette méthode.

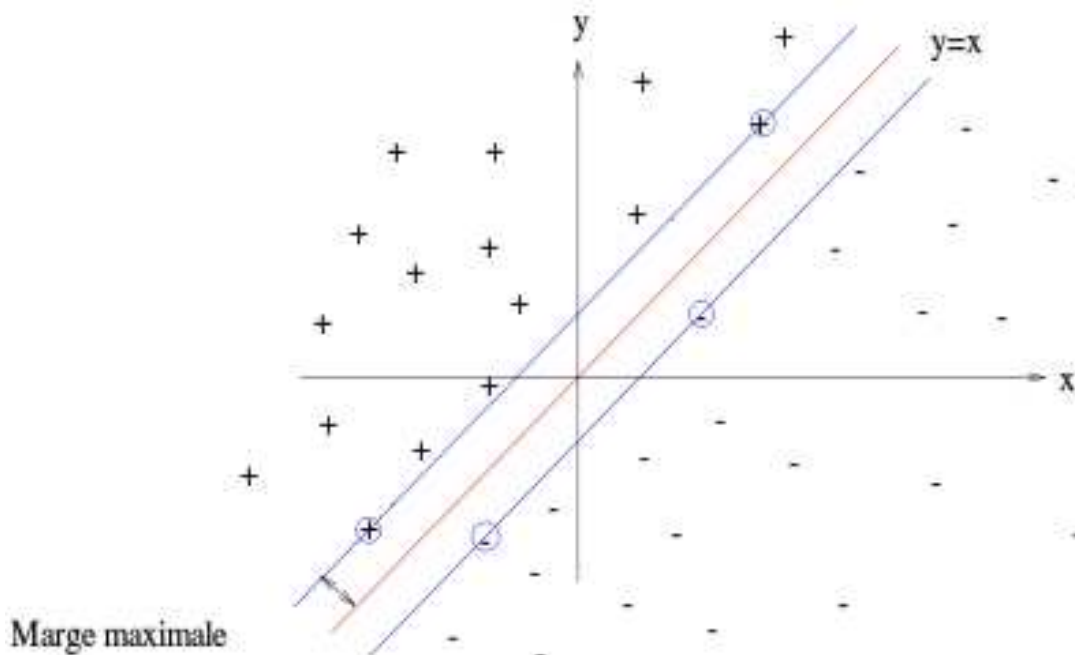


Figure III. 8. L'hyperplan optimal (en rouge) avec la marge maximale. Les échantillons entourés sont des vecteurs supports.

III.4.3. Les Réseaux de Neurones

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate: le neurone formel. Les réseaux de neurones sont constitués d'un

ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone, ces liens sont connectés par des poids.

Ces liens permettent à chaque cellule de disposer d'un canal pour envoyer et recevoir des signaux en provenance d'autres cellules du réseau. Chacune de ces connexions reçoit un poids (une pondération), qui détermine son impact sur les cellules qu'elle connecte. Chaque cellule dispose ainsi d'une entrée, qui lui permet de recevoir de l'information d'autres cellules, mais aussi de ce que l'on appelle une fonction d'activation [33] (figure III.9).

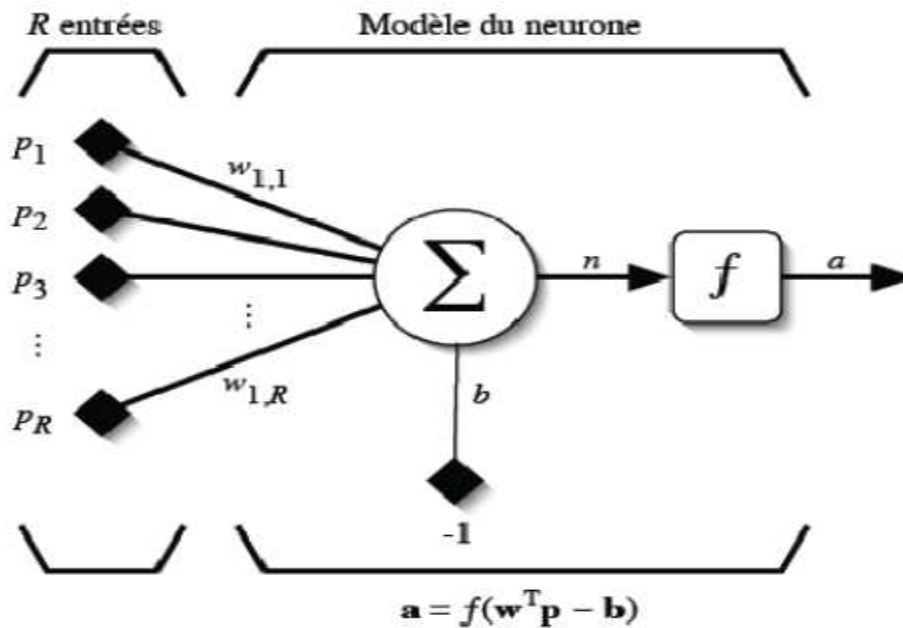


Figure III. 9. Modèle d'un neurone artificiel.

En suivant les notations présentées dans la figure précédente, les R entrées du neurone correspondent au vecteur $\mathbf{p}=[p_1, p_2, \dots, p_R]^T$, alors que $\mathbf{w}=[w_{1,1}, w_{1,2}, \dots, w_{1,R}]^T$ représente le vecteur des poids du neurone. La sortie n du modèle est donnée par l'équation suivante:

$$n = \sum_{j=1}^R \omega_{1,j} p_j - b = \omega_{1,1} p_1 + \omega_{1,2} p_2 + \dots + \omega_{1,R} p_R - b \quad (\text{III.4})$$

Le résultat n de la somme pondérée s'appelle le niveau d'activation du neurone. b est le biais et aussi s'appelle le seuil d'activation du neurone.

La sortie a du neurone est calculée par la fonction d'activation f donnée par l'équation suivante :

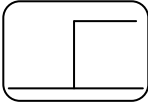
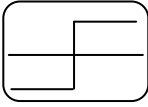
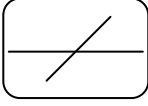
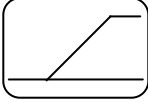
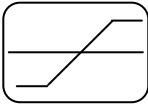
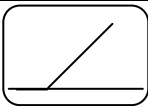
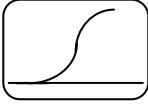
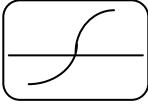
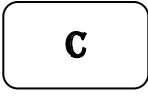
$$a = f(n) \quad (\text{III.5})$$

III.4.3.1. Fonctions de transfert

Jusqu'à présent, nous n'avons pas spécifié la nature de la fonction d'activation de notre modèle. Il se trouve que plusieurs possibilités existent. Différentes fonctions de transfert

pouvant être utilisées comme fonction d'activation du neurone sont énumérées au tableau III.1. Les trois les plus utilisées sont les fonctions «seuil», «linéaire», et «sigmoïde» [34].

Tableau III. 1. Fonctions de transfert $a=f(n)$.

Nom de la fonction	Relation d'entrée / sortie	Icône
Seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
Linéaire	$a = n$	
Linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
Linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
Linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
Sigmoïde	$a = \frac{1}{1 + e^{-n}}$	
Tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Compétitive	$a = 1$ si n maximum $a = 0$ autrement	

III.4.3.2. Architecture de réseau

Un réseau de neurones est un maillage de plusieurs neurones, généralement organisé en couches. Pour construire une couche de S neurones, il s'agit simplement de les assembler comme à la Figure III.10. Les S neurones d'une même couche sont tous branchés aux R entrées. On dit alors que la couche est totalement connectée.

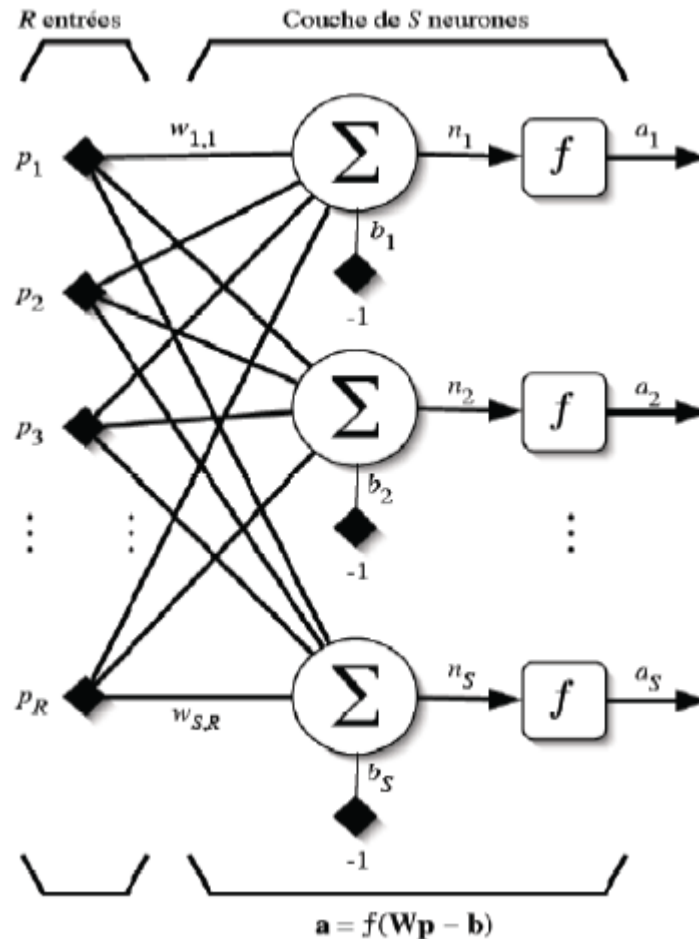


Figure III. 10. Couche de S neurones.

Un poids $w_{i,j}$ est associé à chacune des connexions. Le premier indice i désigne toujours le numéro de neurone sur la couche, alors que le deuxième indice j (colonne) spécifie le numéro de l'entrée. Ainsi, $w_{i,j}$ désigne le poids de la connexion qui relie le neurone i à son entrée j . L'ensemble des poids d'une couche forme donc une matrice \mathbf{W} de dimension $S \times R$.

III.4.3.3. Le perceptron multicouche (Multi Layer Perceptron MLP)

L'idée principale est de grouper des neurones dans une couche. La première couche est reliée aux entrées, puis ensuite chaque couche est reliée à la couche précédente. Les neurones de la première couche sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée. Ils calculent alors leurs sorties qui sont transmises aux neurones de la deuxième couche, etc. Ces

sorties ne sont pas visibles à l'extérieur du réseau, et elles sont appelées pour cette raison couches cachées. C'est la dernière couche qui produit les sorties du réseau [35].

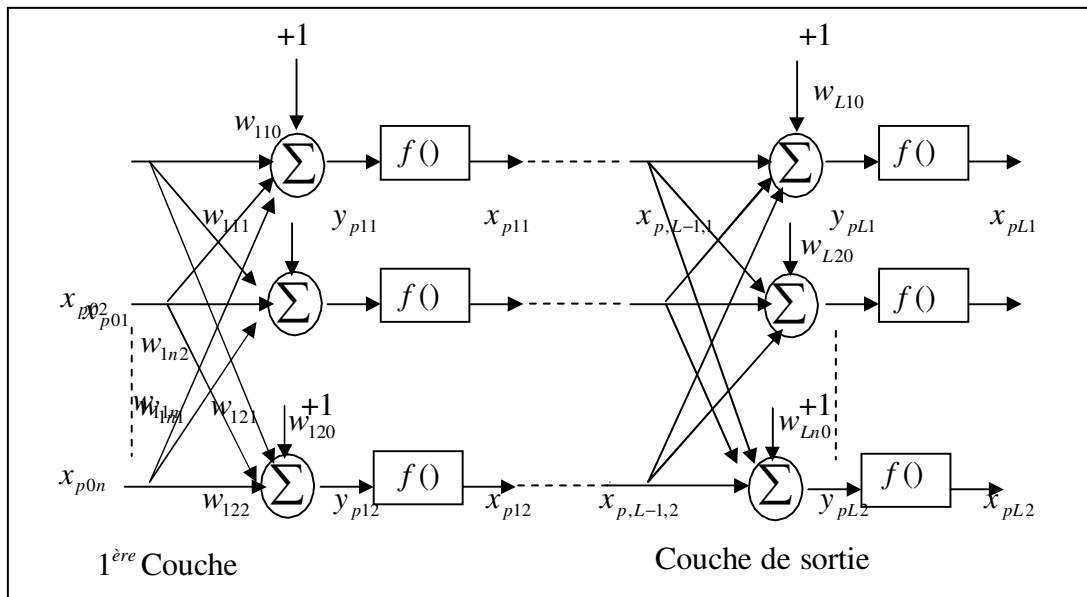


Figure III. 11. Structure d'un réseau de neurone multicouche.

La mise en œuvre des réseaux de neurones multi couche est composée de cinq étapes principales :

Étape 1 : fixer le nombre de couches cachées

Étape 2 : déterminer le nombre de neurones par couches cachées

Étape 3 : choisir la fonction d'activation

Étape 4 : choisir l'apprentissage, la méthode d'apprentissage la plus utilisée est rétro-propagation. L'apprentissage par rétro-propagation nécessite la détermination du paramètre d'ajustement des poids synaptiques à chaque itération.

Étape 5 : La détermination du critère d'arrêt.

III.4.4. Réseaux Neurones Convolutionnels

En apprentissage automatique, un réseau de neurone convolutifs (ou réseau de neurones à convolution, ou CNN ou ConvNet) est un type de réseau de neurones artificiels acycliques dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

III.5. Conclusion

Dans ce chapitre, nous avons présenté l'apprentissage profond et synthétisé les différentes architectures de réseaux de neurones convolutionnels. Nous avons ensuite présenté la classification et les algorithmes d'apprentissage automatique. Ce qui fait montrer l'énorme potentiel des réseaux de neurones artificiels est loin d'être pleinement exploité et leur application n'en est qu'à ses débuts.

Après cette présentation théorique des différents domaines utilisés dans ce projet, le chapitre suivant contient les résultats de nos tests de notre système proposé.

Chapitre IV

Tests et résultats

IV. 1. Introduction

Pour tester le système développé, on doit d'abord préparer le matériels et l'environnement de test, telles que les bases de données et les étiquettes imprimées pour les fixer sur les murs dans les endroits souhaitable.

L'élément principale de notre travail est le Raspberry Pi, ce dernier est connecté à un modem, donc on peut accéder à la configuration du Raspberry Pi depuis un PC portable et recevoir le flux de données (images).

Dans les sections suivantes, nous allons essayer de présenter une description détaillée de la réalisation de notre système.

IV. 2. Description des bases de données

La tâche principale de la navigation intérieure est de trouver l'emplacement exacte des utilisateurs et de leur permettre la navigation en toute sécurité dans des environnements intérieurs tels que les bâtiments publics et les centres commerciaux. Ces dernières années, l'intelligence artificielle (IA) est devenue utile en raison de sa capacité particulière, plusieurs chercheurs ont utilisé les techniques de l'IA pour améliorer la qualité de ces applications en améliorant la précision et en leur permettant d'être adaptées à une utilisation en temps réel. Dans le traitement basé sur des marqueurs, le système se compose d'un Raspberry Pi et une caméra, des marqueurs et un serveur. La caméra est utilisée pour scanner les marqueurs, tandis que le serveur est utilisé pour stocker des informations telles que la carte de bâtiments ou centre commercial. On trouve plusieurs types des marqueurs : codes QR et marqueurs Aruco.

Dans ce mémoire, nous nous sommes appuyés sur les marqueurs Aruco que sur les codes QR parce que les marqueurs Aruco peuvent être détectés à des distances allant jusqu'à 4 m alors que les codes QR étaient limités à 2 m seulement. Détecter les marqueurs Aruco de longues et courtes distances, il n'est pas nécessaire que la caméra soit dans LOS (Line Of Sight). Pour les codes QR, ils ne peuvent pas être détectés à plus de 2 m et si la caméra était dans LOS n'était pas pertinente.

Pour détecter les marqueurs dans les conditions difficiles, les cadres d'image sont affectés par diverses conditions bruyantes telles que le flou et la distorsion des marqueurs, etc. De tels bruits affectent la précision de l'identification des marqueurs. On a créé un ensemble de données qui couvre la plupart des défis. Une base de données est créée correspondante à chaque classe de conditions difficiles et différentes orientations. Cet ensemble de données contient 5 classes de 5 marqueurs (Figure IV.1).

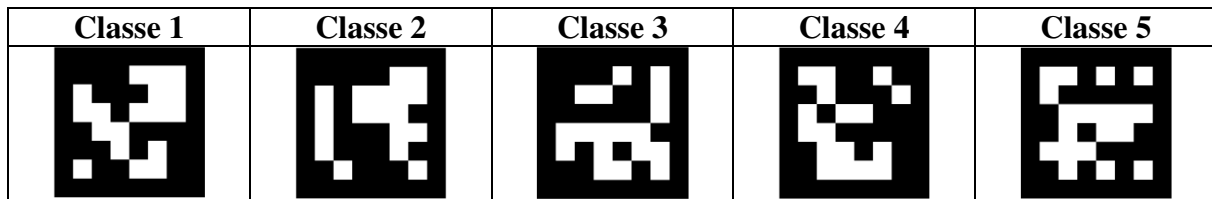


Figure IV. 1. Les 5 classes des marqueurs

Comme la montre la figure ci-dessous, on a créé différents effets en appliquant au hasard diverses transformations aux images originales. Pour chaque classe, on a généré environ 30 échantillons, et en augmente ces derniers par les différents filtres. Ainsi, un total de 7334 images a été créé pour les 5 classes (Figure IV.2).

Pour trouvé les meilleurs paramètres des méthodes de l'intelligence artificielle, on divise l'ensemble de données (7334 images) en deux sous-ensembles (*i*) ensemble d'apprentissage et (*ii*) ensemble de validation, sans qu'une donnée ne soit commune. Un nombre non négligeable de données est bien entendu nécessaire dans l'ensemble de validation pour estimer correctement le critère de généralisation, donc pour optimiser les paramètres de notre méthode de classification utilisé. Souvent on garde 2/3 des données dans l'ensemble d'apprentissage et en réserver 1/3 pour la validation. Les ensembles d'apprentissage et de validation sont tirés aléatoirement parmi les 7334 images disponibles.

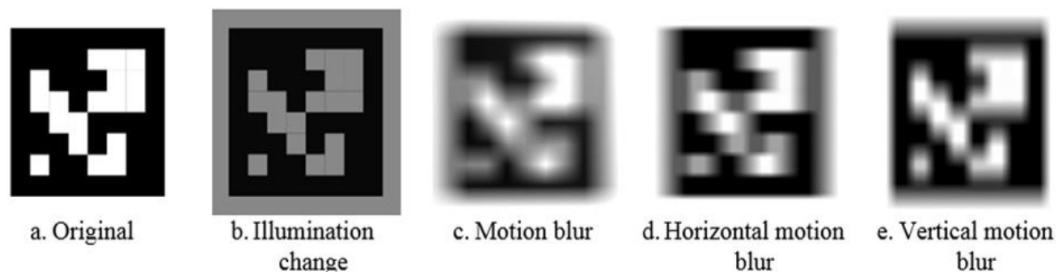


Figure IV. 2. Exemples sur les effet appliqués sur les marqueurs originaux.

Des marqueurs Aruco d'une dimension de 10×10 *cm* sont imprimés et installés dans l'environnement à intervalles réguliers. Pour les essais de détection (segmentation, seuillage) et classification (deep learning).

IV. 3. Evaluation des algorithmes de segmentation implémentés

Dans cette section, nous présentons tout d'abord notre collection de test sur laquelle nous avons appliqué les méthodes de segmentation présentés dans le chapitre II, ensuite les résultats obtenus.



Figure IV. 3. Images de test.

Les figures IV.4 à IV.9 montrent la segmentation des deux images de tests obtenus par les méthodes de segmentation basées sur l'entropie, la similarité, la forme de l'histogramme.

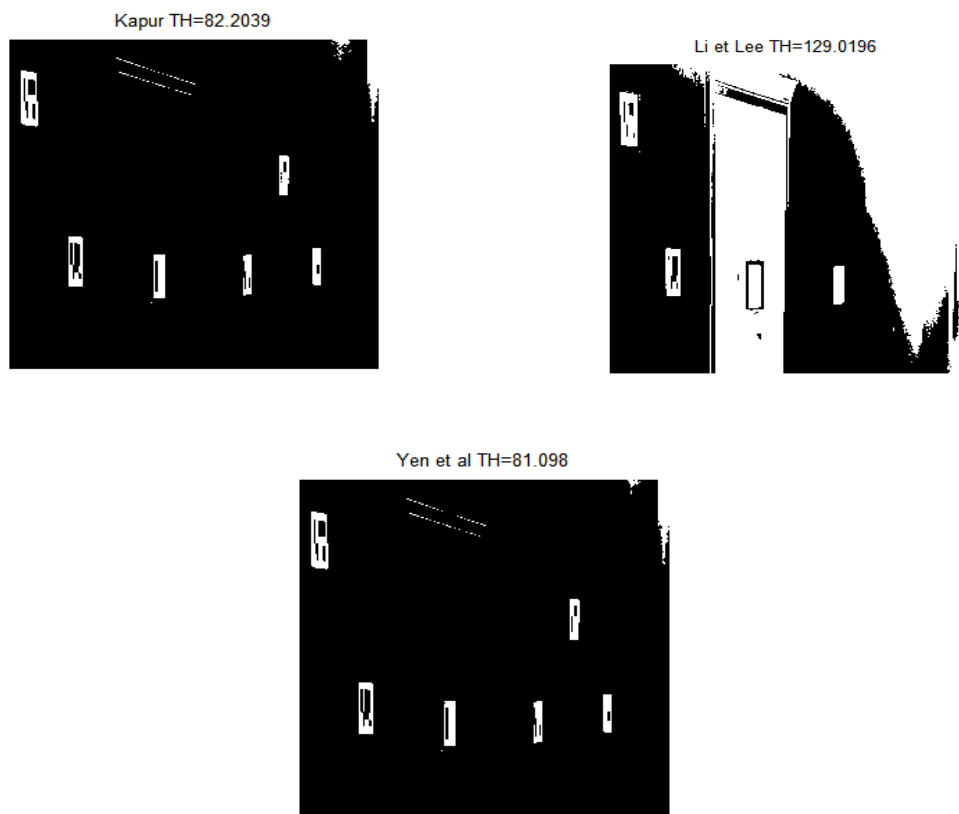


Figure IV. 4. Image de Test 1 segmentée par les méthodes basées sur l'entropie.

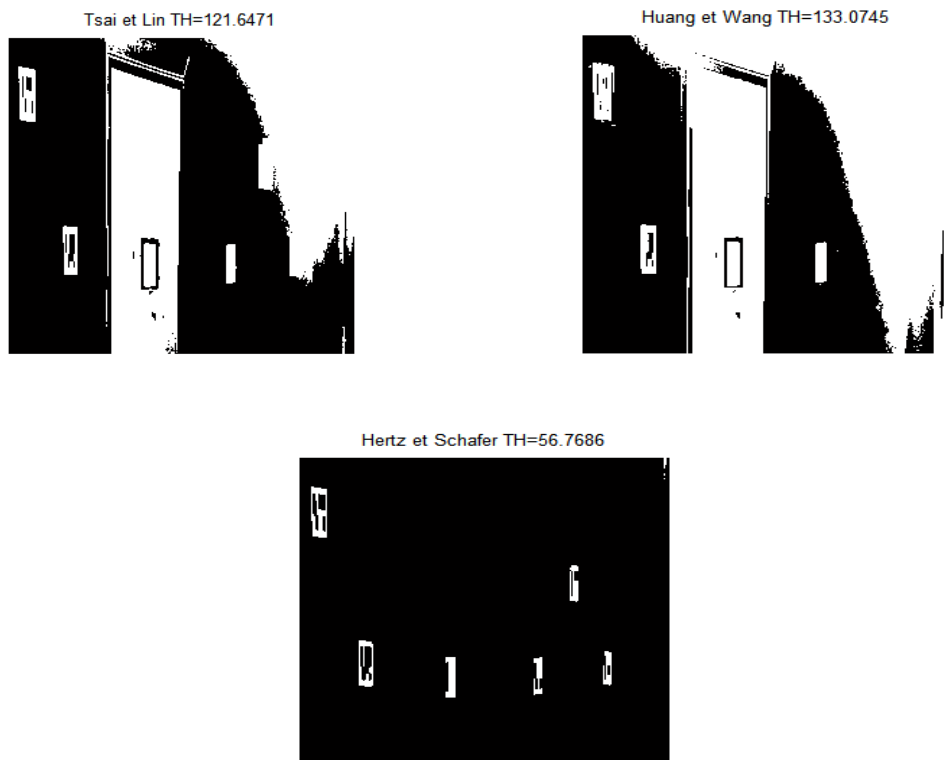


Figure IV. 5. Image de Test 1 segmentée par les méthodes basées sur la similarité.

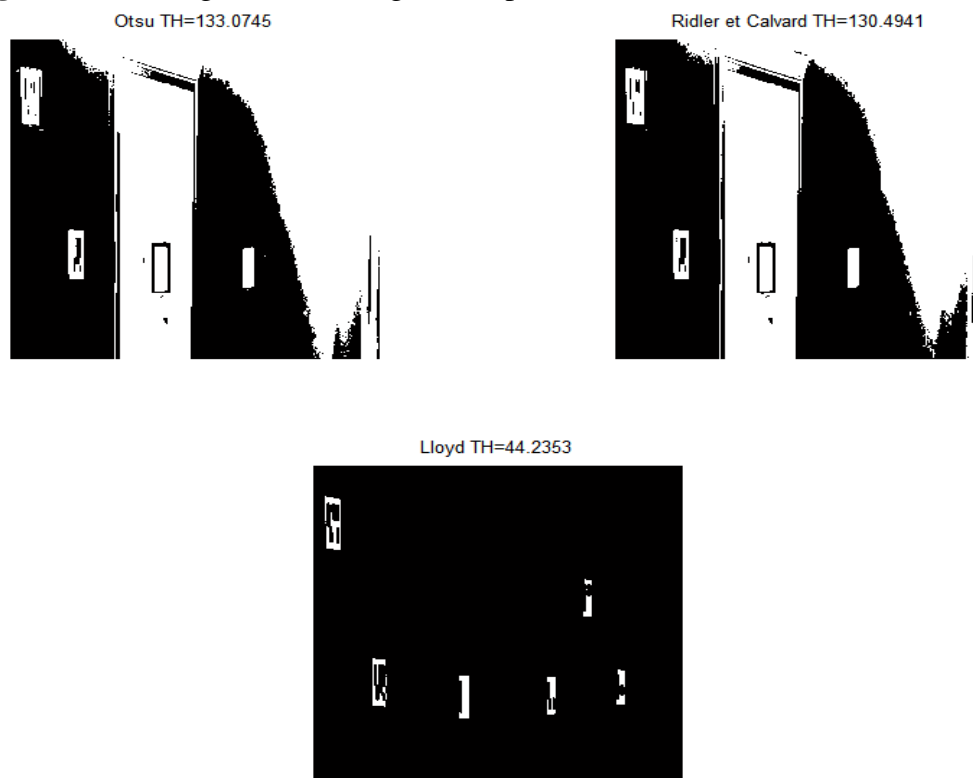


Figure IV. 6. Image de Test 1 segmentée par les méthodes basées sur la forme de l'histogramme.

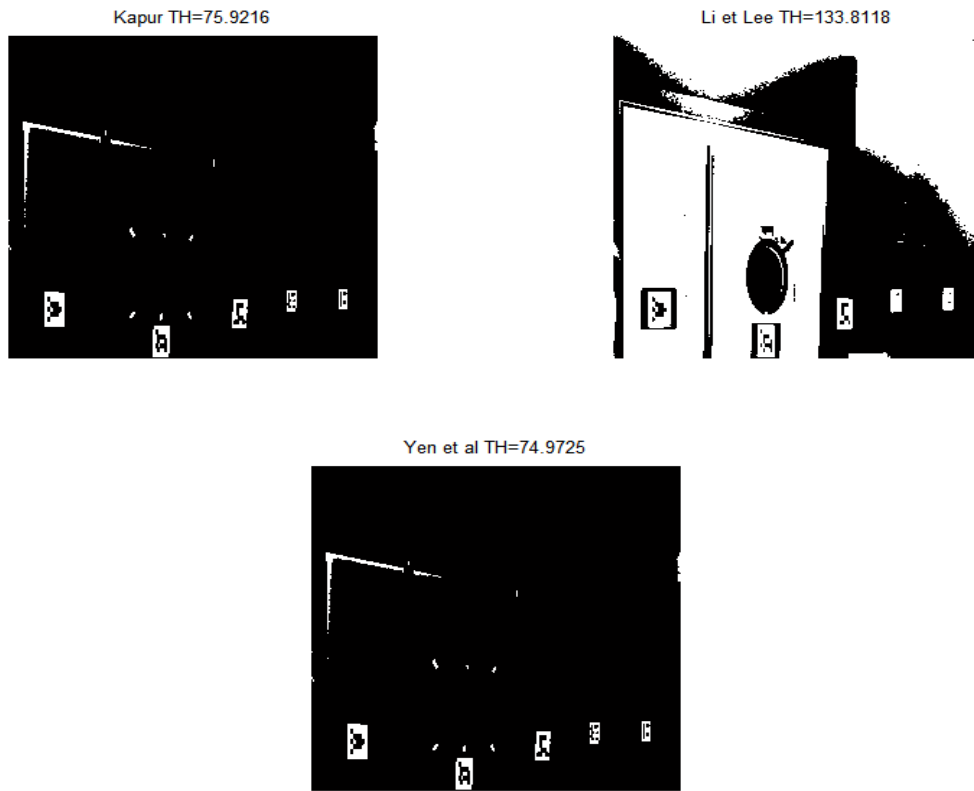


Figure IV. 7. Image de Test 2 segmentée par les méthodes basées sur l'entropie.

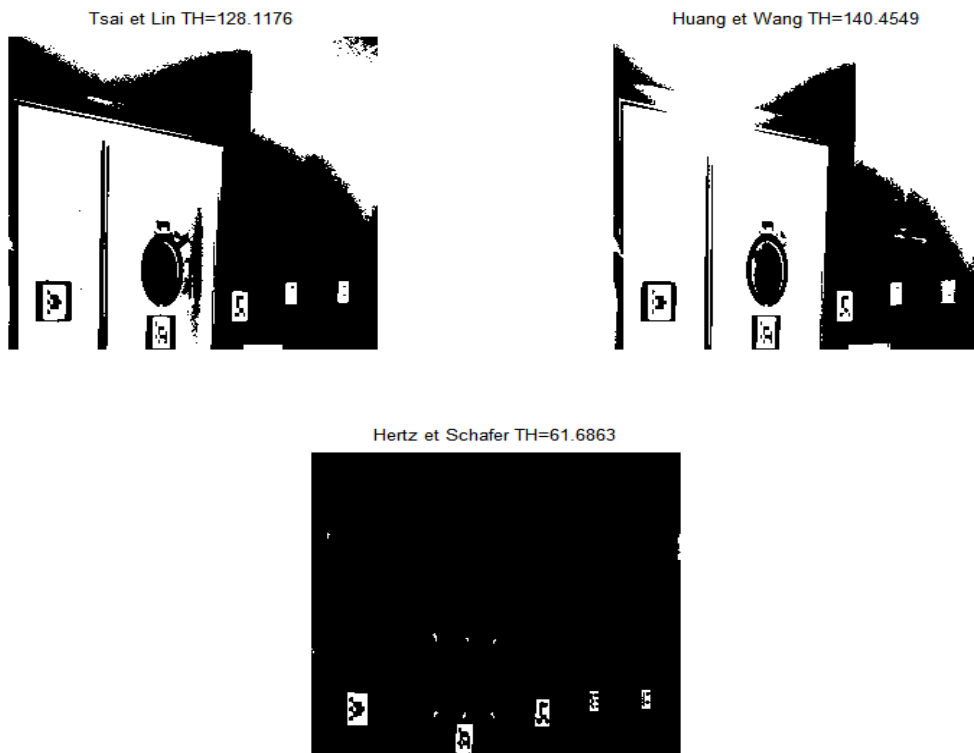


Figure IV. 8. Image de Test 2 segmentée par les méthodes basées sur la similarité.

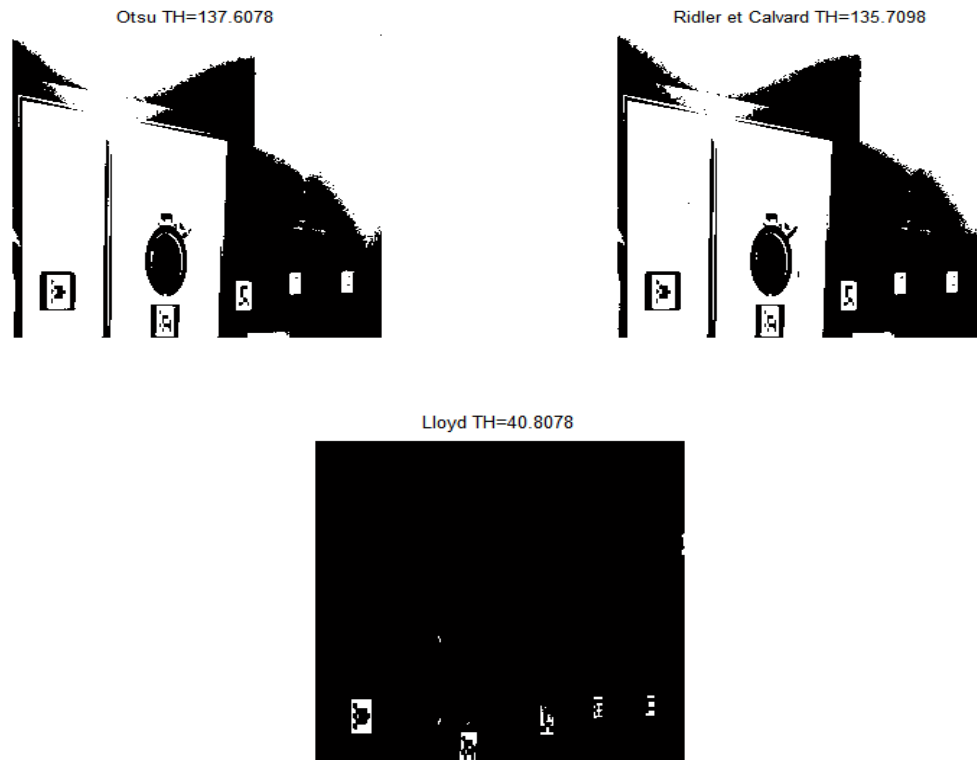


Figure IV. 9. Image de Test 2 segmentée par les méthodes basées sur la forme de l’histogramme.

Dans le chapitre II, nous avons divisés les méthodes de segmentation sur trois catégories : les méthodes basées sur l’entropie, les méthodes basées sur la similarité et les méthodes basées sur la forme de l’histogramme. Dans cette section nous avons implémentés les différentes catégories de segmentation sur deux images de tests.

Les résultats obtenues sur les deux images de test avec la méthode de segmentation basée sur l’entropie ‘‘Hertz et Schafer’’ sont meilleurs par rapport à ceux obtenus avec les méthodes ‘‘Kapur, Li et Lee, Yen et al, Tsai et Lin, Huang et Wang, Hertz et Schafer, Otsu, Ridler et Calvard, et Lloyd’’. Dans la suite de notre travail en utilise la méthode ‘‘Hertz et Schafer’’, comme une méthode de segmentation.

Pour la séparation des régions détectées par la segmentation d’Hertz et Schafer en utilise la méthode de l’étiquetage en composantes connexes (ECC). Cette technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe.

Un exemple de traitement ECC sur deux images de test est donné sur la Figures IV.10 et IV.11. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé. Chaque région étiqueter est un marquer (Figures IV.12 et IV.13) et par la suite identifie en utilisant les méthodes d’apprentissage profond (deep learning).

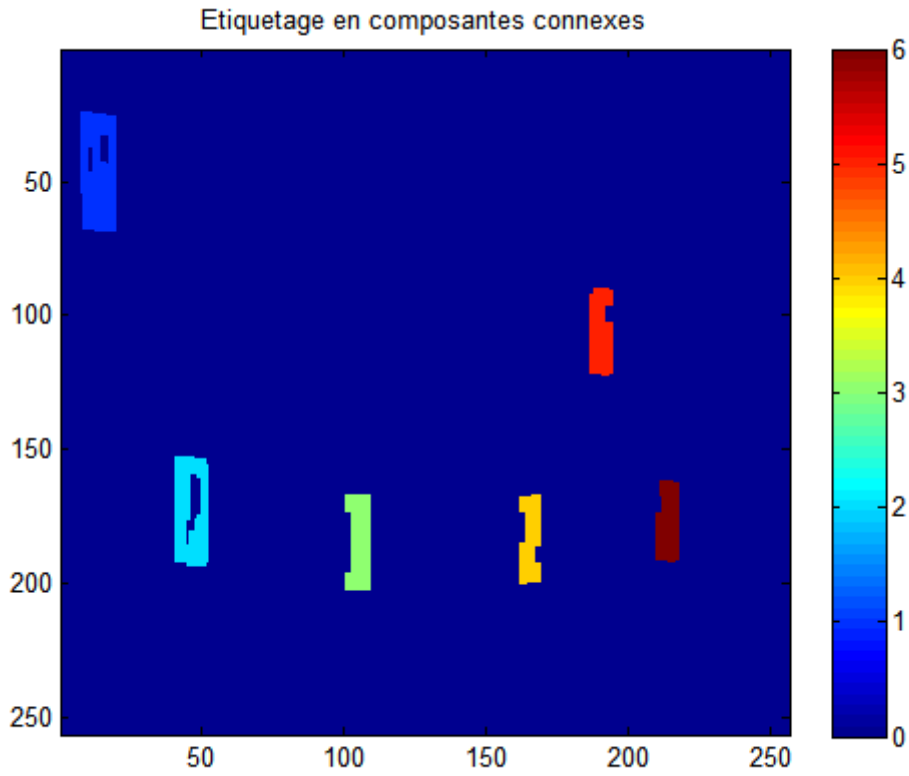


Figure IV. 10. Etiquetage en composantes connexes de l'image de test 1.

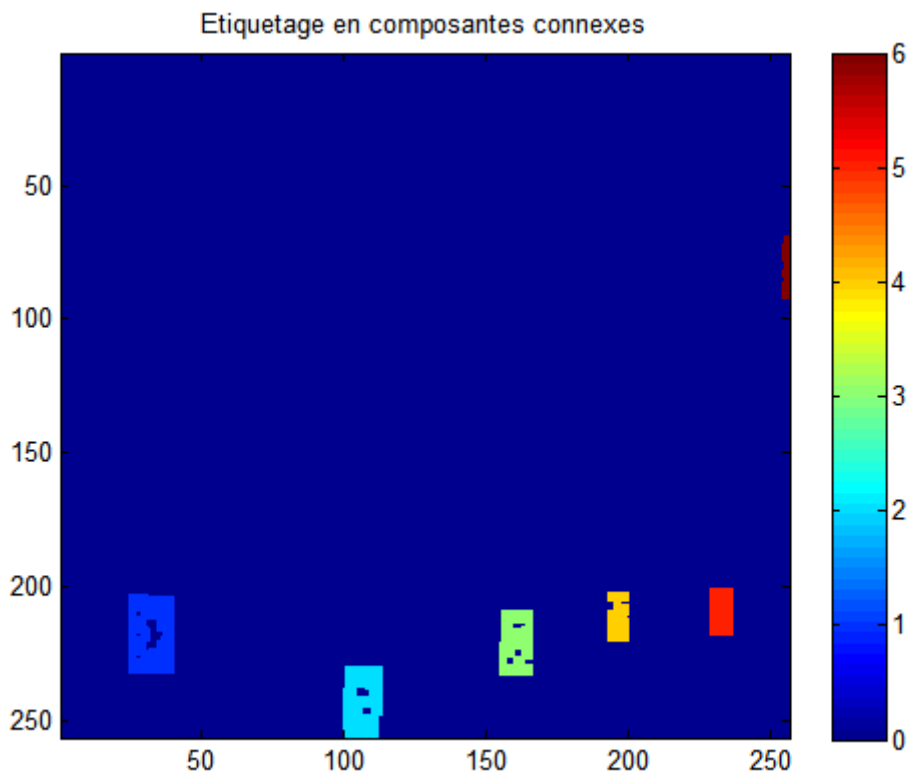


Figure IV. 11. Etiquetage en composantes connexes de l'image de test 2.



Figure IV. 12. Détection de marqueurs Aruco de l'image de test 1.



Figure IV. 13. Détection de marqueurs Aruco de l'image de test 2.

IV. 4. Evaluation des algorithmes d'apprentissage profond implémentés

Nous avons utilisé deux architectures pour classifier les images provenant du flux de la caméra de la carte Raspberry *Pi*. Ce qui nous permet de définir l'emplacement désiré en détectant le marqueur.

On présente dans cette partie l'approche proposée pour l'identification des marqueurs (voir la Figure IV.14). Notre méthode basée sur deux étapes : l'étape hors ligne, dans ce cas on détermine les meilleurs paramètres de notre modèle basée sur l'apprentissage profond, et la deuxième étape en temps réel qui permet d'identifier les marqueurs et déterminer la localisation de l'utilisateur, avec l'utilisation des paramètres trouvés dans l'étape hors ligne.

Pour la première étape de notre méthode proposée, nous avons besoin des critères d'évaluation. Dans notre cas nous avons choisi de calculer deux critères : la précision de classification (*Accuracy*) et l'erreur de classification (*Loss*):

$$Accuracy(\%) = \frac{P}{N} \times 100 \quad (IV.1)$$

P : Echantillons bien classés.

N : Nombre d'échantillons des classes cibles.

$$Loss(\%) = 100 - Accuracy(\%) \quad (IV.2)$$

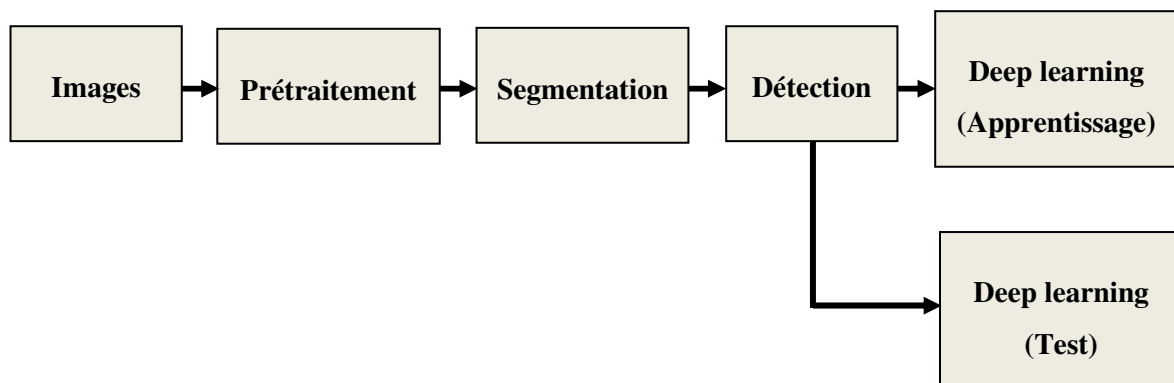


Figure IV. 14. Diagramme block d'identification des marqueurs.

➤ Le prétraitement de l'image

Le prétraitement de l'image est une étape importante, dans la mesure où les images de l'environnement obtenues à partir d'une caméra contiennent les différentes classes ou positions qui peuvent fausser l'étape de segmentation et d'apprentissage profond. Dans notre cas, nous avons utilisé un filtre moyenneur.

➤ La segmentation

L'objectif principal de l'étape de segmentation est de séparer la région de l'image contenant les marqueurs du reste. Dans la section précédente nous avons trouvé que la méthode basée sur la segmentation "*Hertz et Schafer*" donne des meilleurs résultats par rapport à ceux des autres méthodes.

➤ La Détection

Pour la séparation des régions détectées par la segmentation de "*Hertz et Schafer*" on utilise la méthode de l'étiquetage en composantes connexes (ECC).

➤ La Classification

La classification est la dernière étape de notre système de localisation indoor. Son objectif est de reconnaître et d'interpréter les informations à propos des marqueurs basant sur les fonctionnalités extraites des images. La littérature a rapporté l'application de plusieurs méthodes de classification des images. Les méthodes les plus fréquemment utilisées peuvent être classées en termes d'approches. Nous aurons pour cela l'approche par apprentissage profond. Les approches par apprentissage profond semblent, selon [5, 36-38], être les plus prometteuses.

La classification par l'apprentissage profond opère en deux phases consécutives : une phase d'apprentissage et une phase de test. La première phase consiste à choisir l'architecture optimale du modèle (nombre de filtres, les paramètres des filtres, nombres des couches) pour pouvoir s'adapter à différentes conditions d'entrée, l'application de cette phase est en mode hors ligne. Une fois l'entraînement du modèle est terminé, il passe en phase de test pour réaliser la classification en temps réel.

IV. 5. Environnement de développement

Sachant que le Raspberry *Pi* a une faible performance par rapport à un PC, nous avons utilisé un ordinateur pour effectuer la phase de l'apprentissage. Pour former des réseaux de neurones convolutifs (CNN), nous avons besoin d'une machine avec une unité de traitement graphique puissante. Ces unités de traitement graphique sont excellentes pour exécuter les algorithmes de Deep Learning à cause de leur architecture de traitement parallèle. Après la préparation des données, on a procédé à l'apprentissage en utilisant deux architectures CNN simples.

Du fait de la caractéristique limitée de Raspberry *Pi* (CPU) pour la classification en temps réel, nous avons choisi de développer notre algorithme en basant sur des modèles profonds avec des architectures simples. Durant la phase d'apprentissage, seuls les poids des couches sont mis à jour en utilisant l'algorithme d'optimisation RMSprop ($lr=0.0001$, $\rho=0.9$).

IV.5.1. Résultat de l'apprentissage

La première architecture proposée est constituée de deux couches de convolution (avec une entrée de dimension $28 \times 28 \times 3$), dont chaque couche on utilise un filtre de taille 3×3 , et nous avons utiliser 16 filtres dans la 1^{er} couche, et 32 filtres dans la 2^{ème} couche. Les deux couches convolutives conv1-2 avec une rectification linéaire Relu. Cette fonction force les neurones à retourner des valeurs positive, une couche de Maxpooling est appliquée après chaque couche de convolution, afin de réduire la taille de l'image et la quantité des paramètres, et une couche de normalisation a été appliqué aussi pour chaque couche de convolution, le raison pour laquelle nous utilisons, cette couche est d'accélérer l'apprentissage et de réduire la sensibilité

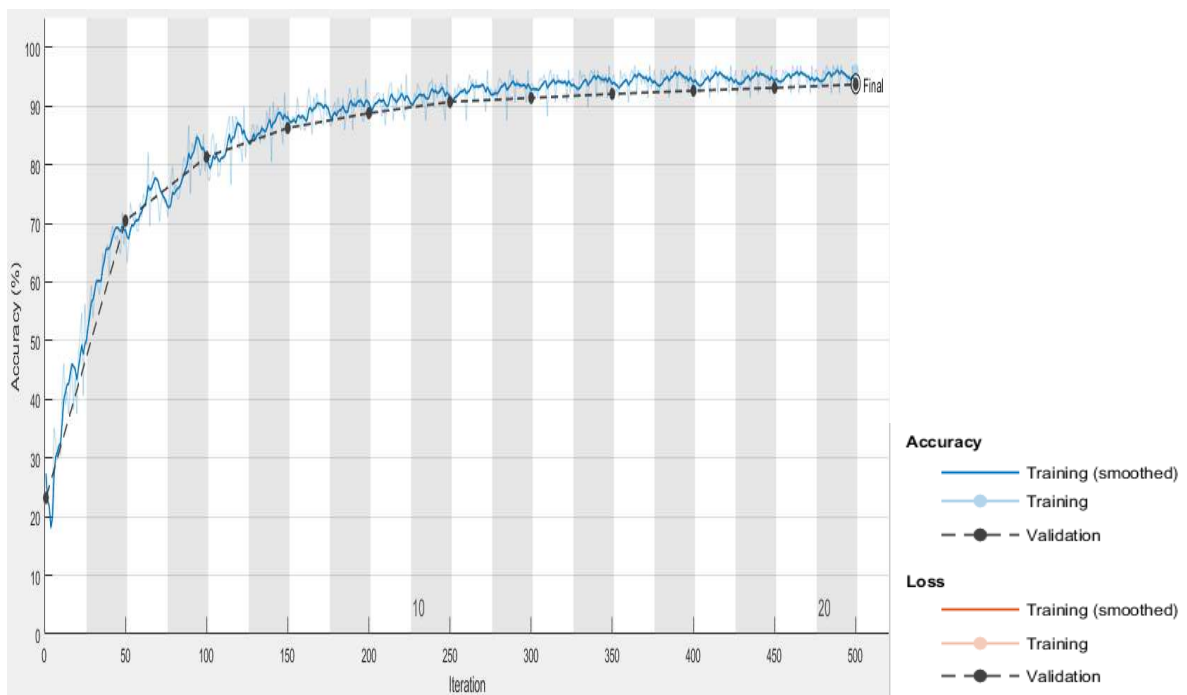
de l'initialisation du réseau. La dernière couche est une couche entièrement connectée, où elle a été ajoutée avec l'utilisation d'un classifieur complètement connectés (le même principe de réseaux de neurones multicouche).

Les résultats de l'apprentissage sont figurés ci-dessous :

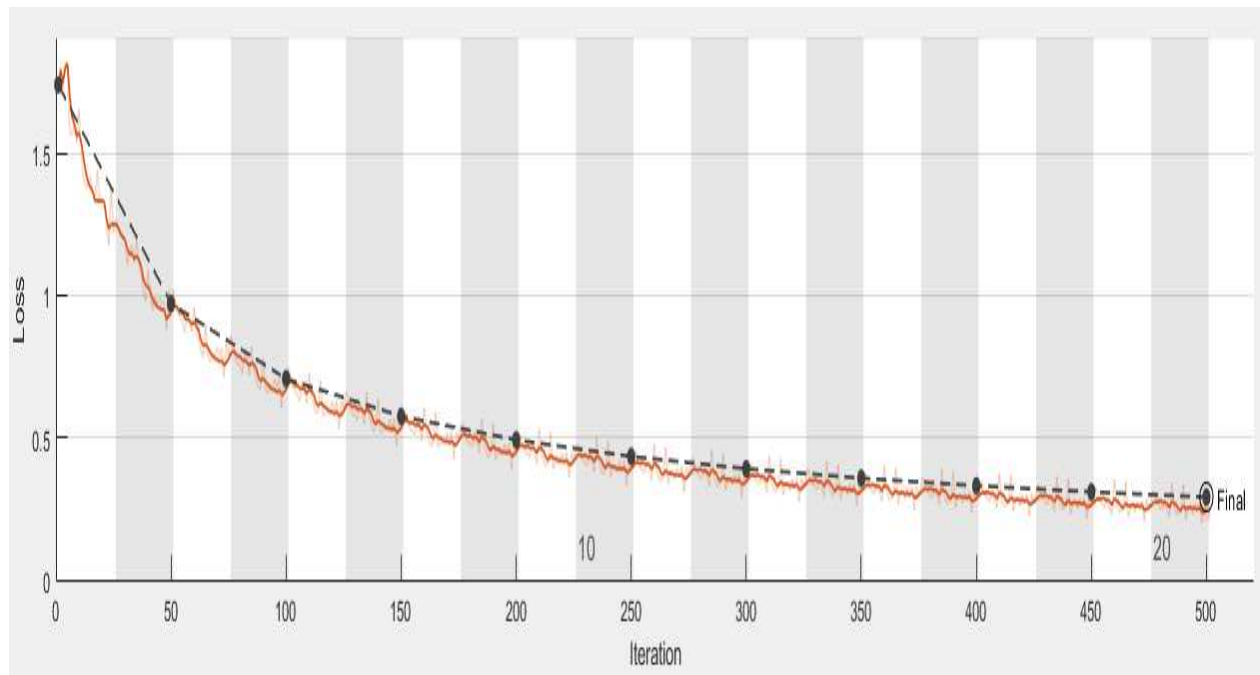
Tableau IV. 1. Résultats de la première architecture de notre modèle d'approche par apprentissage profond sur le jeu de test.

```

Training on single GPU.
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Accuracy   | Loss       | Loss       | Rate          |
|=====|=====|=====|=====|=====|=====|=====|=====|
| 1     | 1       | 00:00:24    | 27.34%    | 23.20%    | 1.7100    | 1.7400    | 1.0000e-04   |
| 2     | 50      | 00:00:28    | 69.53%    | 70.45%    | 0.9604    | 0.9712    | 1.0000e-04   |
| 4     | 100     | 00:00:30    | 81.25%    | 81.31%    | 0.6822    | 0.7076    | 1.0000e-04   |
| 6     | 150     | 00:00:33    | 89.06%    | 86.24%    | 0.5443    | 0.5757    | 1.0000e-04   |
| 8     | 200     | 00:00:36    | 90.63%    | 88.77%    | 0.4564    | 0.4925    | 1.0000e-04   |
| 10    | 250     | 00:00:39    | 92.19%    | 90.68%    | 0.3942    | 0.4344    | 1.0000e-04   |
| 12    | 300     | 00:00:42    | 93.75%    | 91.36%    | 0.3476    | 0.3913    | 1.0000e-04   |
| 14    | 350     | 00:00:44    | 94.53%    | 92.04%    | 0.3115    | 0.3581    | 1.0000e-04   |
| 16    | 400     | 00:00:47    | 94.53%    | 92.60%    | 0.2825    | 0.3316    | 1.0000e-04   |
| 18    | 450     | 00:00:50    | 95.31%    | 93.09%    | 0.2585    | 0.3098    | 1.0000e-04   |
| 20    | 500     | 00:00:53    | 96.09%    | 93.65%    | 0.2384    | 0.2917    | 1.0000e-04   |
|=====|=====|=====|=====|=====|=====|=====|=====|
    
```



(a)



(b)

Figure IV. 15. Résultats d'apprentissage de l'architecture 1 , en termes de (a) précision (Accuracy) et (b) d'erreur (Loss).

La deuxième architecture est constituée de trois couches de convolution avec la configuration suivante :

Tableau IV. 2. Configuration de l'architecture 2.

Couche	Taille de filtre	Nombre de filtre	ReLu	MaxPooling
Conv-1	7	100	Oui	Oui
Conv-2	4	150	Non	Oui
Conv-3	4	250	Non	Oui

L'image en entrée est de taille $48 \times 48 \times 3$, l'image passe d'abord à la première couche de convolution. Cette couche est composée de 100 filtres de taille 7×7 . Après cette convolution 150 filtres de taille 4×4 seront créés. Les trois couches de convolution est suivie d'une fonction d'activation ReLU cette fonction force les neurones à retourner des valeurs positives. La dernière couche est constituée de 250 filtres de taille 4×4 . À la sortie de cette couche, nous aurons 5 classes de marqueurs.

Les résultats de l'apprentissage profond sont figurés ci-dessous :

Tableau IV. 3. Résultats de la deuxième architecture de notre modèle d’approche par l’apprentissage profond sur le jeu de test.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:11	28.13%	34.86%	10.6316	9.2461	1.0000e-04
2	50	00:00:30	67.19%	91.36%	1.0197	0.3058	1.0000e-04
4	100	00:00:49	85.16%	95.43%	0.4529	0.1703	1.0000e-04
6	150	00:01:09	86.72%	96.24%	0.2969	0.1276	1.0000e-04
8	200	00:01:28	89.06%	97.35%	0.2550	0.1001	1.0000e-04
10	250	00:01:48	94.53%	97.47%	0.1312	0.0902	1.0000e-04
12	300	00:02:07	93.75%	98.70%	0.1580	0.0649	1.0000e-04
14	350	00:02:26	91.41%	98.83%	0.2250	0.0514	1.0000e-04
16	400	00:02:45	94.53%	98.83%	0.1654	0.0519	1.0000e-04
18	450	00:03:04	97.66%	98.95%	0.0860	0.0468	1.0000e-04
20	500	00:03:25	96.88%	98.89%	0.1025	0.0417	1.0000e-04

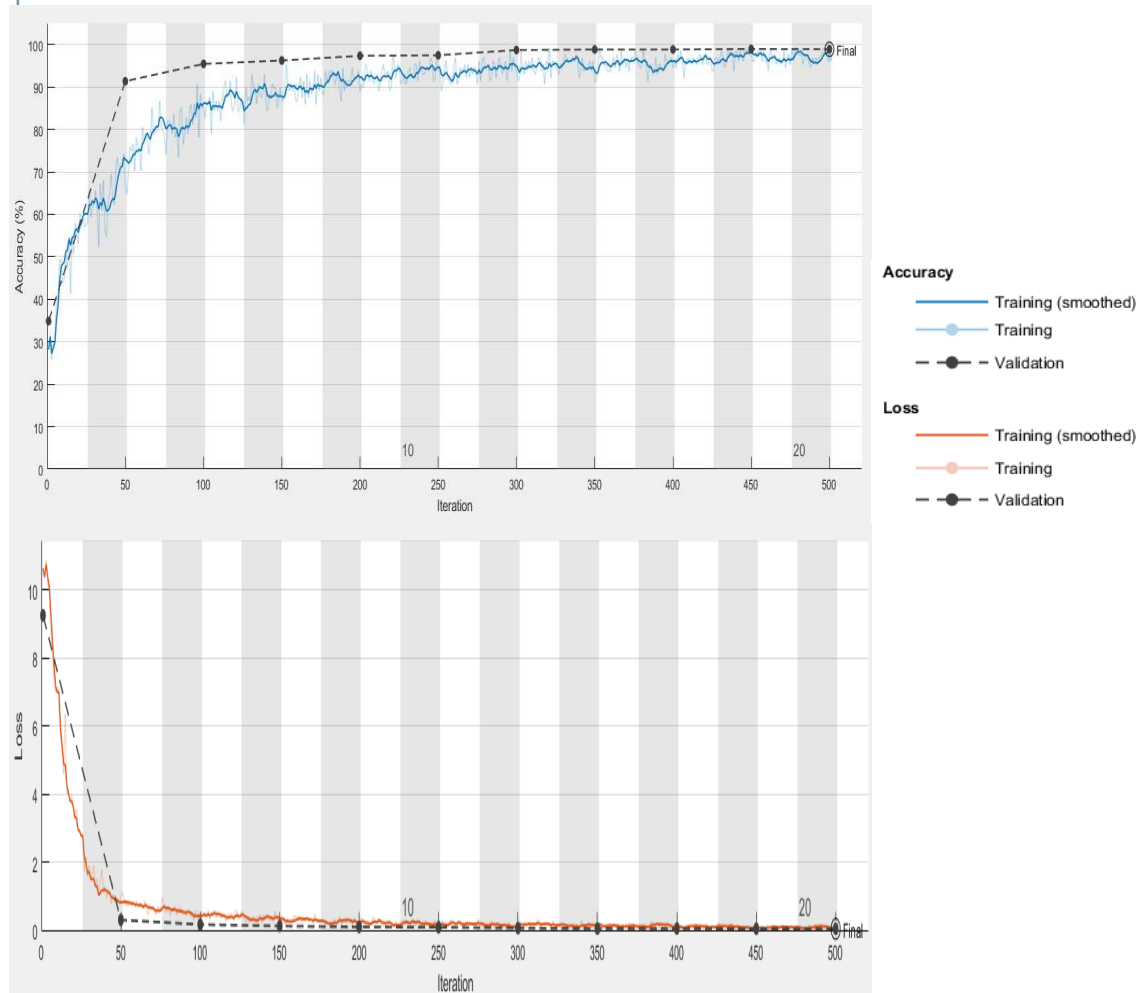


Figure IV. 16. Résultats d’apprentissage de l’architecture 2 en termes de précision (Accuracy) et d’erreur (Loss).

Le Tableau IV.4 présente les performances sur la base de données de validation des deux types des classifieurs CNN.

Tableau IV. 4. Validation sur le jeu de validation des classifieurs 28×28 et 48×48.

Méthodes	Accuracy (%)	Loss
Architecture 1	93.65	0.2917
Architecture 2	98.89	0.0417

Ce tableau illustre une comparaison entre les meilleures performances obtenues par les deux architectures proposée sur l'ensemble de validation. Nous remarquons que le classificateur de la deuxième architecture donne des résultats significativement meilleurs que la première architecture. Les meilleures combinaisons entre la segmentation, détection, et la classification sont examinés dans la section suivante.

IV. 6. Réalisation finale et tests

Nous avons développé une application basée sur une carte Raspberry Pi et une caméra avec l'utilisation de la technologie WiFi (Figure IV.17), pour l'objectif de localisé l'emplacement d'un utilisateur dans une Map, comme le montre dans la figure IV.18. Dans un premier temps, l'application ouvre la caméra pour obtenir un flux d'images en temps réel. Puis en implémente les méthodes segmentation, détection et classification pour identifier le marqueur et trouver l'emplacement exacte de l'utilisateur. Comme le montre la figure IV.18, le marqueur ArUco est un marqueur carré avec une zone binaire interne qui stocke l'identifiant du marqueur et une large bordure noire. Les bordures noires sont utilisées pour une détection rapide de l'image projetée. Dans le cas d'un marqueur visible par la caméra, notre système d'identification peut facilement identifier le marqueur avec l'utilisation de l'apprentissage profond.



Figure IV. 17. Matériel utilisé dans le test.

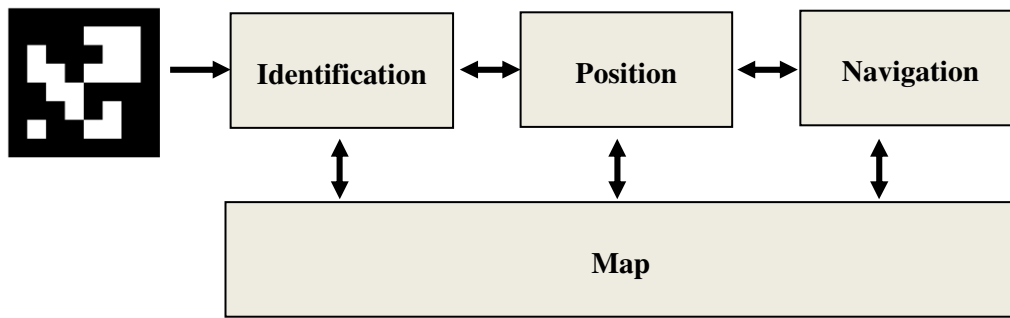
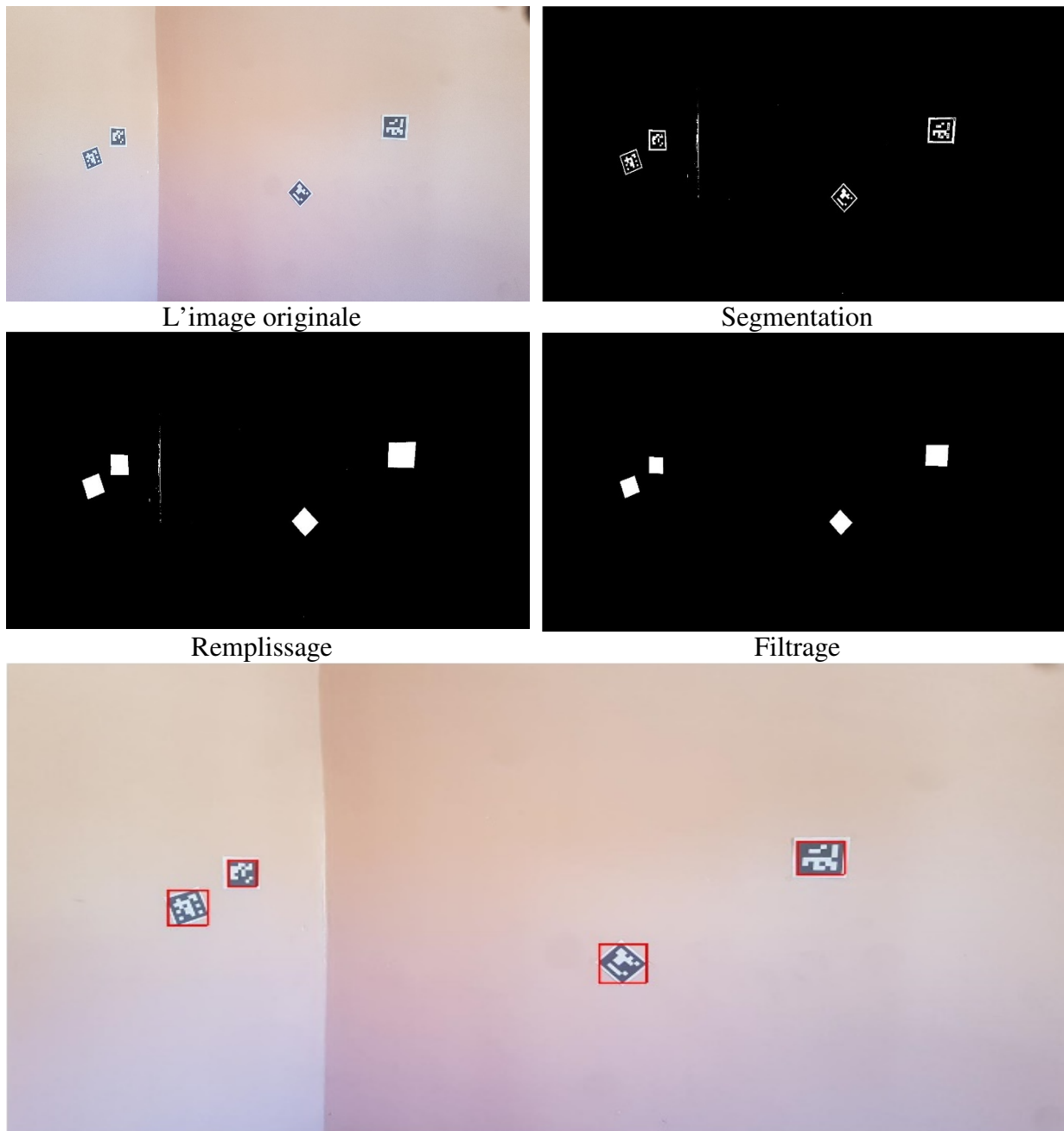


Figure IV. 18. Système de localisation.



Détection et classification
 Figure IV. 19. Les captures des étapes du test.

La figure IV.19 représente un exemple d'identification et de classification des marqueurs avec l'utilisation de la carte Raspberry Pi et une caméra. Les différentes étapes de notre méthode se résument en : Nous avons construit une carte interne, on utilise des nœuds pour représenter les points d'intérêt dans la carte par exemple comme l'emplacement des portes. Ensuite, on utilise la caméra de Raspberry pour capturer les images de notre environnement en temps réel. Les étapes qui suivent sont : Prétraitement, segmentation, et classification. Pour améliorer notre méthode de segmentation, nous proposons ensuite d'appliquer sur les images binaires une stratégie de remplissage des objets trouvés et ensuite des techniques de filtrage morphologique. Le filtrage consiste à réduire de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image. Dans notre cas, nous allons effectuer une érosion (Ouverture morphologique). Cela nous permet de supprimer les pixels isolés qui ne correspondent pas à notre marqueur. Ensuite nous effectuons une dilatation qui nous permet de renforcer les groupes denses de pixels.

Après la détection et l'identification, nous devrions obtenir la position de l'utilisateur à partir des marqueurs. La classification de marqueur correspond à la position de l'utilisateur dans la carte.

IV. 7. Conclusion

Dans ce chapitre nous avons présenté les différentes étapes qui nous ont permis de réaliser un système de localisation indoor en utilisant les techniques de l'intelligence artificielle, on a présenté aussi l'efficacité et les performances de notre algorithme de détection et d'identification des marqueurs implémenté sur la carte Raspberry Pi.

Conclusion générale

Conclusion générale

La localisation indoor basée sur l'IA est au cœur d'une dynamique de recherches nouvelles depuis plus d'une dizaine d'années. Nous avons évoqué les avantages potentiels qu'il nous a apportés l'IA a travers sa grande précision dans la classification utilisant la technique de l'apprentissage profond à partir du mécanisme de réseaux de neurone convolutif (CNN). En considérant dans un premier temps la localisation classique dont ses points faibles dans les environnements indoor ont limité les utilisateurs de bénéficier du service de localisation indoor. Dans un second temps, nous avons présenté les atouts de l'IA qui peuvent être renforcé plusieurs domaines dans notre vie quotidiennes, nous avons enfin montré que la combinaison des techniques de traitement d'image décrites et l'apprentissage profond (CNN) donne une forte précision et rapidité et ajoute une fiabilité d'informations dans les décisions de classification.

Le travail sur ce projet nous a permis de réutiliser des notions vues en cours de notre étude et de les appliquer au cas d'un système réel. Cela a été l'occasion de revisiter les notions de la localisation par GPS, déjà vu précédemment dans notre cursus, et de perfectionner notre connaissance de Matlab. Nous avons découvert également la réalisation pratique d'un système de localisation indoor, qui nous a été très utile afin de visualisé le système.

Enfin, ce projet va nous permettre de développer de nombreux domaines du quotidien, tel que système d'aide à la navigation destiné aux malvoyants à accomplir leurs tâches de vie quotidiennes telles que les achats et les déplacements au sein des différents services (administration, hôpitaux, ...).

Bibliographie

Bibliographie

- [1] N. Mekhfi, N. Lahbab, "Mise au point d'une application de Géolocalisation Indoor via WiFi," *Master en Informatique, Option: Réseaux et Systèmes Distribués (R.S.D), Université Abou Bakr Belkaid– Tlemcen*, 2018.
- [2] R. Tapu, B. Mocanu., T. Zaharia, "Wearable assistive devices for visually impaired: A state of the art survey," *Pattern Recognition Letters*, vol. 137, pp. 37-52, 2020.
- [3] E. Valero, A. Adán., C. Cerrada, "Evolution of RFID Applications in Construction: A Literature Review," *Sensors* vol. 15, pp. 15988–16008, 2015.
- [4] B. Ozdenizci, V. Coskun, K. Ok, "NFC internal: An indoor navigation system," *Sensors* vol. 15, pp. 7571-7595, 2015.
- [5] M. Elgendy, T. Guzsvinecz, C. Sik-Lanyi, "Identification of Markers in Challenging Conditions for People with Visual Impairment Using Convolutional Neural Network," *Applied Sciences*, vol. 9, pp. 1-24, 2019.
- [6] H. Zhang, C. Zhang, W. Yang, C.-Y. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," *Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China*, pp. 2501-2506, 6–9 December 2015.
- [7] A. Idrees, Z. Iqbal, M. Ishfaq, "An efficient indoor navigation technique to find optimal route for blinds using QR codes," *Proceedings of the 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), Auckland, New Zealand*, pp. 690–695, 15–17 June 2015.
- [8] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, pp. 2280–2292, 2014.
- [9] E. Marchand, H. Uchiyama, F. Spindler, "Pose Estimation for Augmented Reality: A Hands-On Survey," *IEEE Trans. Vis. Comput. Graph*, vol. 22, pp. 2633–2651, 2016.
- [10] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, R. Medina-Carnicer, "Generation of fiducial marker dictionaries using Mixed Integer Linear Programming," *Pattern Recognition*, vol. 51, pp. 481-491, 2016.
- [11] M. Kara, "Réseau de capteurs sans fil: étude en vue de la réalisation d'un récepteur GPS différentiel à faible coût " *Thèse de doctorat Université Blaise Pascal-Clermont-Ferrand II*, 2009.
- [12] S. Malys, J. Slater, "Maintenance and enhancement of the World Geodetic System 1984," *In Proceedings of the 7th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1994)*, pp. 17–24, 1994.
- [13] H. T. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, pp. 254-256, 1946.
- [14] P. Bahl, V. N. Padmanabhan, "Radar : an in-building rf-based user location and tracking system," *In The 19th Conference of the IEEE Computer and Communications Societies (Infocom'00)*, 2000.
- [15] A. Theljeoui, "Système hybride de localisation des personnes âgées dans un habitat intelligent," *Thèse de doctorat, Université Toulouse, Spécialité: Réseaux, Télécoms, Systèmes et Architecture*, 2017.
- [16] F. Evennou, "Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor," *Thèse de doctorat, Université Joseph Fourier, Spécialité: Optique et Radio Fréquences*, 2007.
- [17] Yu-Chee Tseng, "Active Bat: A 3D Location Device," 1998.

- [18] J.N. Kapur, P.K. Sahoo, A. C. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 273-285, 1985.
- [19] C. H. Li, C. K. Lee, "Minimum Cross Entropy Thresholding," *Pattern Recognition*, vol. 26, pp. 617-625, 1993.
- [20] J. C. Yen, F. J. Chang, S. Chang, "A New Criterion for Automatic Multilevel Thresholding," *IEEE Transactions on Image Processing*, vol. 4, pp. 370-378, 1995.
- [21] W. H. Tsai, "Moment-preserving thresholding: A new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 377-393, 1985.
- [22] L. K. Huang, M. J. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, pp. 41-51, 1995.
- [23] L. Hertz, R. W. Schafer, "Multilevel thresholding using edge matching," *Comput Vision Graphics Image Process*, vol. 44, pp. 279-295, 1988.
- [24] N. Otsu, "A threshold selection method for grey level histograms," *IEEE Transactions on System, Man and Cybernetics SMCvol*, vol. 9, pp. 62-66, 1979.
- [25] T. W. Ridler, S. Calvard, "Picture thresholding using an iterative selection method," *IEEE transactions on Systems, Man and Cybernetics*, vol. 8, pp. 630-632, 1978.
- [26] D. E. Lloyd, "Automatic Target Classification Using Moment Invariant of Image Shapes," *Technical Report RAE IDN AW126, Farnborough-UK*, 1985.
- [27] B. Marir, M. Kalla, F. Douak, A. Daamouche, "A modular support vector machine for active learning of urban remote sensing images classification in Algeria," *Journal of the Indian Society of Remote Sensing*, vol. 46, pp. 515-529, 2018.
- [28] A. Habba, O. Ishak, "La classification des images satellitaires par l'apprentissage profonde (deep learning)," *Mémoire Master, Option: Systèmes Intelligents, Université Ahmed Draïa-Adrar*, 2019.
- [29] M. M. Zakaria, "Classification des images avec les réseaux de neurones convolutionnels," *Mémoire de Master en informatique. Option : Modèle Intelligent et Décision, Université Abou Bakr Belkaid Tlemcen*, 2017.
- [30] B. Daha, M. Medileh, "Un miroir intelligent interactif basé sur Raspberry Pi," *Mémoire de Master en informatique, Spécialité: Systèmes Distribués et Intelligence Artificielle, Université Echahid Hamma Lakhdar El Oued*, 2018.
- [31] V. Vapnik, "The nature of statistical learning theory," *Springer science & business media*, 2013.
- [32] B. Schölkopf, A. J. Smola, "Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond," *MIT Press*, 2002.
- [33] C. Touzet, "Les réseaux de neurones artificiels introduction au connexionnisme," *Préface de Jeanny Hérault, EC2 éd. France*, 1992.
- [34] M. Parizeau, "Réseaux de Neurones GIF-21140 et GIF-64326," *université de Laval*, 2006.
- [35] F. Moutarde, "Introduction aux réseaux de neurones," *Centre de Robotique (CAOR), Ecole des Mines de Paris*, 2007.
- [36] M. L. Mekhalfi, F. Melgani, Y. Bazi, N. Alajlan, "Toward an assisted indoor scene perception for blind people with image multilabeling strategies," *Expert Systems with Applications*, vol. 42, pp. 2907-2918, 2015.
- [37] M. L. Mekhalfi, F. Melgani, Y. Bazi, N. Alajlan, "A Compressive Sensing Approach to Describe Indoor Scenes for Blind People," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, pp. 1246-1257, 2015.
- [38] P. Theodorou, A. Meliones, "Towards a Training Framework for Improved Assistive Mobile App Acceptance and Use Rates by Blind and Visually Impaired People," *Education Sciences*, vol. 10, p. 58, 2020.