

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene



Faculté de Génie Electrique

Domaine Sciences et Technologie

Filière Electronique

Mémoire de Licence

Spécialité Electronique

Thème

**Parking intelligent automatique basé IoT :
La conception et la réalisation d'un système
d'enregistrement de matricule automatique**

Présenté par :

CHERAITIA Islam

191931042293

S. Amina

Encadreurs :

Dr. CHEKIR Amira

Année Universitaire 2021-2022

REMERCIEMENTS

Nous remercions Dieu « ALLAH » tout puissant qui nous a donné la patience et le Courage durant ces longues années d'étude et qui nous a Donné le courage pour réaliser le défi de ce projet, et la force de finaliser ce travail.

On veut Nous remercier D'avoir cru en Nous... On Veux Nous remercier de N'avoir Jamais abandonné... On Veux Nous remercier D'être La et D'être ce Que Nous sommes Maintenant

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons à remercier sincèrement notre encadreur Madame CHEKIR Amira pour la confiance qu'elle nous a accordé, son encouragement, et son précieux conseils et notre équipe de jury

Ainsi à toute l'équipe d'atelier Qui nous as rassemblé et former Une merveilleuse Equipe Qui a produit Un merveilleux Smart Parking

Pour conclure, ont effectué Ce travail au Notre faculté d'électronique et d'informatique (F.E.I) et on remercie toutes les personnes et tout le département d'électronique qui a participé d'une manière ou autre de près ou de loin à l'exécution de ce travail.

Résumé :

La dernière décennie fut marquée par une augmentation significative du nombre de véhicules, il est devenu nécessaire de profiter de l'énorme développement dans le domaine des télécommunications, réseaux, et l'intelligence artificielle afin de créer un parking adapté programmé pour faire fonctionner et gérer cette énorme quantité de véhicules et ainsi réduire le phénomène de stationnement chaotique et éviter l'exposition au vol.

De nos jours il existe de nombreux systèmes de reconnaissance des plaques d'immatriculation dans le monde, ces systèmes ont deux axes majeurs, qui sont la détection de la plaque d'immatriculation et la reconnaissance de ses caractères. Par conséquent ce présent travail propose de concevoir et implémenter un système de reconnaissance des plaques d'immatriculation algérienne basé sur la reconnaissance de caractères. Le système implémenté, sur une « Raspberry Pi 2 », passe par trois étapes majeures ; prétraitements et amélioration, localisation de la plaque, et enfin la reconnaissance des caractères. Les résultats des expérimentations obtenus montrent que la méthode proposée achève des taux de reconnaissance acceptable malgré le non-respect des normes de construction des plaques d'immatriculation en algerie, ce qui rend difficile de les identifier automatiquement.

Mots clés : Reconnaissance, Localisation, Segmentation, Plaques d'immatriculation, Contours, Raspberry Pi, OpenCV

ملخص:

تميز العقد الماضي بزيادة كبيرة في عدد المركبات ومع تطور تصميمها والزيادة المقابلة في عدد السرقات المستهدفة خاصة لمكونات السيارات باهظة الثمن ، أصبح من الضروري الاستفادة من التطور الهائل في مجال الاتصالات والشبكات والذكاء الاصطناعي من أجل إنشاء موقف مكيف مبرمج لتشغيل وإدارة هذه الكمية الهائلة من المركبات وبالتالي الحد من ظاهرة الفوضى في وقوف السيارات وتجنب التعرض للسرقة.

يوجد حاليا العديد من أنظمة التعرف على لوحات الترخيم في العالم، وهذه الأنظمة تركز على محورين أساسيين، هما اكتشاف موضع لوحة الترخيم والتعرف بعد ذلك على أرقامها. للقيام بهذه العملية يتم عموما استعمال عدة خصائص من طرف هذه الأنظمة، الخلفيات الثابتة التي تثبت عليها اللوحات ... إلخ ، وتختلف من حيث الطرق المعتمدة كالاعتماد على لون اللوحات المعروف مسبق لاستخراج الأرقام والتعرف عليها. لذلك، نقترح من خلال هذا العمل تصميم وتنفيذ نظام للتعرف على لوحات الترخيم الجزائرية معتمدين فيها على مبدأ استكشاف ودراسة حواف الأشياء وكذا محركات التعرف على الأحرف. النظام المنجز، والذي يشغل على جهاز " راسبيري يمر بثلاث خطوات رئيسية ؛ المعالجة والتحسين، توطين اللوحة، وأخيرا التعرف على الأرقام التي تحويها. تظهر نتائج " باي " التجارب التي تم الحصول عليها أن الطريقة المقترحة تحقق معدلات تعرف مقبولة على الرغم من عدم الامتثال لمعايير إنشاء لوحات الترخيم ما يصعب عملية التعرف عليها أوتوماتيكيا.

الكلمات المفتاحية : التعرف، تحديد الموقع، التجزئة ، لوحات الترخيم، الحواف، راسبيري باي، أوبن سي في

Abstract :

Nowadays, there are many license plate recognition systems in the world, these systems have two major axes, which are the detection of the license plate and the recognition of its characters.

Therefore, this present work proposes to design and implement an Algerian license plate recognition system based on character recognition. The system implemented, on a "Raspberry Pi 2", goes through three major stages; pre-processing and enhancement, plaque localization, and finally character recognition. The results of the experiments showed that the proposed method achieves acceptable recognition rates despite the non-compliance with the construction standards of license plates which makes it difficult to identify them automatically.

The last decade has been marked by a significant increase in the number of vehicles and with the development of their design and the corresponding increase in the number of targeted thefts especially for expensive automotive components, it has become necessary to take advantage of the huge development in the field of telecommunications, networks, and artificial intelligence in order to create a suitable parking program programmed to operate and manage this enormous amount of vehicles and in consequence reduce the phenomenon of disorder parking and avoid exposure to theft.

Liste des abréviations

OCR : optical character recognition
RAPI : reconnaissance automatique de plaques d'immatriculation
IOT: Internet of Things
CCD: charge-coupled device
SVM: Support Vector Machine
ARM: Advanced RISC Machines
GNU: stand for GNU's Not Unix
RISC: Reduced Instruction Set Computer
OS: One side
GPIO: General Purpose Input/Output
LED : Light Emitting Diode
GPU : unité de traitement graphique
USB: The Universal Serial Bus
SD: Secure Digital
MIPI: Mobile Industry Processor Interface
CSI: Camera Serial Interface
HDMI: High-Definition Multimedia Interface
PIL: Python Imaging Library
MPSI:
IDE: Integrated development environment
PNG: Portable Network Graphics
TIFF: Tag Image File Format
JPEG: Joint Photographic Experts Group
PDF: portable document format
WIFI: Wireless fidelity

SOMMAIRE

Remerciement

Introduction Général

Chapitre 1 : Technique de détection de matricules Dans les parking intelligent.....1

Introduction	1
I. Un aperçu sur le smart parking	
1. Définition du smart parking	2
2. Plaques d'immatriculations	2
II. La détection et la reconnaissance automatique des plaques de matriculations	
1. La vision intelligente par ordinateurs	3
III. Les algorithmes de traitements d'images	
1. SVM (Support Vector Machine)	5
2. OCR (reconnaissance optique des caractères)	6
3. Python-tesseract.....	8
IV. Etat de l'art	
1. Les dernières technologies utilisent pour détecte les matricules.....	9
Conclusion	12

Chapitre 2 : Les composants et logiciel et outils utilisés.....13

Introduction	13
I. Matérielles utilisés	
1. Raspberry pi 2	14
2. Raspberry pi camera.....	15
3. Capteur ultra son	16
4. Les actionneurs	17
II. Logiciels utilisés	
1. Le module PIL de Python La bibliothèque PIL (Python Imaging Library).....	17
2. Open-cv	17
3. Python.....	18
4. Firebase	18
5. Thonny IDE sur Raspberry pi OS.....	19
6. Les commandes (les bibliothèques)	19
Conclusion	21

Chapitre 3 : conception et réalisation détaillées 22 |

Introduction	22
I. Conception détaillée	
1. Téléchargée et installée Raspberry Pi Imager.....	22
2. Gravure l'image du système d'exploitation Raspberry Pi sur la carte SD.....	23
3. Configuration de notre Raspberry Pi.....	24
4. Connecte Raspberry pi à un ordinateur.....	24
5. Installation des librairies nécessaires	25
II. Réalisation	
1. Montage.....	26
2. Programation.....	28
Conclusion.....	31

Conclusion Général

Liste des figures

Figure 1.1 : voiture avec matricule algérien.....	2
Figure 1.2 : les composants nécessaires dans les dernières technologies utilisées pour Détecter les matricules	9
Figure 2.1 : les fonctions des pins du Raspberry pi 2.....	14
Figure 2.2: l'aspect extérieur du Raspberry pi	14
Figure 2.3: Raspberry Pi Camera Module, Adjustable-Focus.....	15
Figure 2.4: Comment connecter le module caméra Pi au Raspberry Pi.....	16
Figure 2.5 : Dimensions de pi camera (B) rev 0.2.....	16
Figure 2.6 : Fonctionnement de capteur ultrason.....	17
Figure 2.7 : capteur ultrasons HC-SR04.....	17
Figure 2.8: l'interface et l'emplacement de Thonny Python IDE.....	19
Figure 3.1 : le site officiel pour télécharger Raspberry pi imager	23
Figure 3.2: alimentation de Raspberry pi 2.....	24
Figure 3.3 : Associer Raspberry Pi à un ordinateur portable à l'aide de la VNC viewer.....	25
Figure 3.4 : simulation	26
Figure 3.5 : Montage réel	26
Figure 3.6 : le prise du matricule par la Raspberry pi camera.....	27
Figure 3.7 : La reconnaissance du matricules.....	27
Figure 3.8 : les matricules enregistrés dans la base de données	28
Figure 3.5 : organigramme	30

Introduction générale :

Jusqu'à présent l'accès à internet était limité à des appareils tel que l'ordinateur ou le smart phone mais maintenant avec l'internet des objets (IoT) pratiquement tous les appareils pourrons être connectés à internet et surveillés à distance.

L'IoT est un système d'appareils interconnectés qui transfère et échange des données sur un réseau sans fil sans aucune intervention humaine, ces appareils sont connectés au réseau à l'aide des passerelles. Ces passerelles traitent les informations collectées à partir des capteurs et les transfèrent vers le cloud qui agit à la fois comme unité de stockage et de traitement. Toutes les actions effectuées sur les données collectées sont utilisées pour un apprentissage avec des capacités améliorées de surveillances des réponses et d'analyses. L'IoT est adoptée dans presque toutes les industries et les domaines ouvrant des portes à la fabrication d'innombrables applications comme le manufacturing, le transport et logistiques, le secteur public, les soins et la santé. L'IoT avec un si large éventail d'applications, son avenir semble plus prometteur que jamais. L'intégration de L'IoT avec d'autres technologies comme le cloud computing le machine learning, l'intelligence artificielle ouvre la voie à de nombreuses innovations nouvelles et passionnantes.

Dans les dernières années, l'identification des véhicules est devenue nécessaire pour plusieurs raisons tel que la sécurité, l'organisation du stationnement ... Chaque voiture est caractérisée par son numéro de plaque d'immatriculation. Pour cela l'identification des véhicules doit se faire de manière très efficace en utilisant des ressources limitées. L'un des meilleurs moyens est la reconnaissance automatique des plaques d'immatriculations en utilisant L'IOT.

Le numéro de plaque d'immatriculation est un code numérique utilisé pour identifier de manière unique les véhicules. La reconnaissance des plaques d'immatriculation basée sur L'IOT, utilise la technologie de traitement des images et de reconnaissance des caractères (OCR) pour identifier les véhicules en lisant automatiquement les numéros de plaque d'immatriculation.

Le principe de ce système est l'identification automatique des voitures à partir de captures d'images prises sur le devant ou à l'arrière des véhicules. Passer par des étapes ; la première consiste à extraire à partir d'une image du véhicule, obtenu d'une caméra haute résolution, la plaque de la voiture, en se basant essentiellement sur une méthode de détection de contours. Cette plaque subit par la suite une segmentation en caractères afin d'être analysée par un système de reconnaissance optique de caractère (OCR) (optical character recognition).

Le processus de reconnaissance automatique des plaques d'immatriculation (RAPI) peut avoir de nombreuses applications. Il est peut-être utilisé dans les parcs de stationnement, les postes de frontières et de péage, la surveillance des autoroutes, ... etc. C'est une technologie qui trouve son essence dans ces 20 dernières années dans le développement des techniques de traitement d'image ainsi que dans les OCR.

De ce fait, une problématique peut être posée de la façon suivante :

- Quels sont les systèmes RAPI existants dans le marché, et quel sont les avantages et les inconvénients de chaque système ? Quel est le système le plus approprié à notre cas d'étude ?
- 2 - Après avoir choisi un système à implémenter, quelles sont les étapes à suivre pour qu'on puisse concevoir un système RAPI efficace et performant ? - Quels sont les outils matériels et logiciels à exploiter pour réaliser un tel système ?

Pour arriver à répondre aux questions composant notre problématique, nous allons organiser notre travail, en quatre chapitres de la façon suivante :

- ✓ Le premier chapitre donnera un aperçu sur le smart parking et le traitement d'images avec le logiciel de traitements d'images et les dernières technologies utilisée pour détecter les matricules.
- ✓ Le deuxième chapitre sera consacré aux différents outils utilisés dans notre projet comme les matériels électroniques et les logiciels utilisés.
- ✓ Le troisième chapitre est consacré à la conception et la réalisation détaillées de notre projet.

CHAPITRE 1 : Technique de détection de matricules dans les parkings intelligents

Introduction :

L'un des problèmes auxquels nous sommes confrontés dans les parkings est celui de la réglementation et de la sécurité. Pour éliminer ces obstacles, nous sommes amenés à identifier chaque voiture par une particularité spécifique telle que : la plaque de numérotation ou d'immatriculation.

Le plus grand problème aux quels nous confortes en Algérie est les matricules algériens qui ne sont pas biométrique, ce qui cause la difficulté de connaître le matricule de la voiture algérien, la seule solution pour remédier à ce problème, est d'utiliser la technologie de reconnaissance d'images et le traitement d'image pour détecte le matricule.

Dans ce chapitre, nous étudions les technologies utilisées pour détecte, connaître et enregistres les plaques d'immatriculations en abordant les méthodes de traitements d'images et les différents logiciels de traitements d'images.

I. Un aperçu sur le smart parking

1. Définition du smart parking :

Le Smart Parking, ou parking intelligent fait notamment appel aux technologies IoT (Internet of Things). En pratique, plusieurs possibilités s'offrent aux gestionnaires de parking. La plus répandue reste la pose de capteurs sur les places de stationnement qui permettent d'informer les usagers de la disponibilité des places. Certaines technologies vont plus loin en apportant des informations sur le temps de stationnement, s'il y a eu un dépassement (lien avec l'horodateur), le paiement du stationnement via une application mobile, l'analyse des plaques d'immatriculation en cas d'infraction, etc. Le smart parking s'intègre parfaitement dans un projet plus global de smart city. [1]

2. Plaques d'immatriculations :

Plaque d'immatriculation Une plaque de matricule est une plaque portant une combinaison unique de chiffres ou de lettres (pour une zone géographique donnée), destinée à identifier facilement un véhicule terrestre (automobile, moto, véhicule agricole, etc.)

Pour l'Algérie Le numéro d'immatriculation est composé de 3 groupes de chiffres séparés par un espace come illustre la Figure 1.1.

- 1 er groupe : 5 chiffres, dont le 0 initial, qui correspond au numéro de dossier du véhicule
- 2 e groupe, composé de 3 chiffres : le premier permet d'identifier le type de véhicule (ex : 1 un véhicule familial, 2 pour un poids lourd, 3 voiture commerciale, 4 bus, 5 tracteur d semi-remorque, 6 tracteur, 7 engins, 8 remorque, 9 moto.) ; les deux suivants renvoient à l'année de mise en circulation du véhicule
- 3 e groupe : 2 chiffres qui identifient la wilaya d'immatriculation.



Figure 1.1 : voiture avec matricule algérien

II. La détection et la reconnaissance automatique des plaques de matriculations :

1. La vision intelligente par ordinateur :

La vision par ordinateur est constituée de l'ensemble des technologies de traitement de l'information numérique ayant pour origine des capteurs photométriques. Un système de vision artificielle prend en entrée une ou plusieurs images numériques et effectue des traitements sur ces images afin d'en extraire des informations relatives à la scène observée. Contrairement à la perception humaine qui est basée sur deux images (œil droit et œil gauche), l'ordinateur peut utiliser des sources d'informations plus nombreuses et plus variées, éventuellement dans d'autres bandes spectrales que le visible. [2]

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées. [3]

1.1. Le concept de l'image [4]

Une image est une représentation planaire d'un objet quelconque. Mathématiquement, c'est une fonction bidimensionnelle de la forme $f(x, y)$, où $f(x_0, y_0)$ est la valeur de f aux coordonnées (x_0, y_0) . Donc, c'est un processus continu 2D résultant d'une mesure 6 physique. L'amplitude de f est appelée intensité (ou niveau de gris) de l'image au point de coordonnées (x, y) . Si une image est générée par un processus physique, $f(x, y)$ correspond à l'énergie irradiée par ce processus. Exemples de processus physiques : Emission et réflexion d'ondes lumineuses, Rayonnement infrarouge, rayons X, Echo ultrason, etc...

Quand les valeurs de l'amplitude de f et des coordonnées (x, y) sont des quantités discrètes, l'image est appelée image numérique. Une image numérique est donc composée d'un nombre fini d'éléments, ayant chacun un emplacement et une valeur donnés. Ces éléments sont appelés pixels (contraction de PICture Element).

1.2. Le point fort des images numériques par rapport aux images analogiques :

La grande différence entre images numériques et analogiques est que les images numériques peuvent se décrire par un ensemble fini de valeurs entières. Si on connaît cette suite de valeurs, on peut recréer une copie exacte de l'image d'origine. On peut assimiler cette suite de valeurs entières à un "code génétique" de l'image. Au contraire, une image analogique est liée à un support matériel : plaque photo, pigments de peinture et toile, par exemple. Il n'est pas possible de reproduire l'image originale à l'identique.

Les copies sont nécessairement Dégradées par rapport à l'original. Ceci est un point fort très important des images numériques : on peut créer de nombreuses répliques de l'image sans dégradation et on peut facilement transmettre, stocker, dupliquer ce fameux "code génétique" d'une image numérique comme on transmettrait un texte

1.3. De l'analogiques vers le numériques :

En mathématiques, une image est une fonction. Cette fonction quantifie l'intensité lumineuse de n'importe quel point dans l'image. Comme une image est une surface délimitée du plan, on peut se munir d'un repère cartésien et repérer la position de ses points par les coordonnées (x, y) (x, y) , avec xx son abscisse et yy son ordonnée.

Une image numérique est une discrétisation de notre modèle continu, Cette discrétisation se fait à la fois sur l'ensemble de définition de la fonction image (échantillonnage, et sur son ensemble d'arrivée (quantification).et enfin l'étape de codage pour coder et stocker dans la mémoire de l'ordinateur la valeur et les coordonnées de chaque point

1.4.Le traitement d'image :

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelles). Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. [5]

Le traitement d'images est l'ensemble des opérations effectuées sur l'image, afin d'en améliorer la lisibilité et d'en faciliter l'interprétation. C'est, par exemple, le cas des opérations de rehaussement de contraste, élimination du bruit et correction d'un flou. C'est aussi l'ensemble d'opérations effectuées pour extraire des "informations" de l'image comme la segmentation et l'extraction de contours.

Avant le traitement d'images, on peut aussi effectuer des opérations de prétraitement qui sont toutes les techniques visant à améliorer la qualité d'une image. De ce fait, la donnée de départ est l'image initiale et le résultat est également une image. Les opérations de traitement peuvent être divisées en deux niveaux : Le traitement bas-niveau qui se construit autour des méthodes d'analyses d'image ayant pour but d'extraire des caractéristiques des images et d'analyser sans les interpréter (contours, texture, par exemple). C'est des données de nature numérique Le traitement haut-niveau intègre l'ensemble des méthodes permettant d'interpréter les caractéristiques issues du bas-niveau (prise de décision, classification, IA) C'est des entités de nature symbolique associées à une représentation de la réalité extraite de l'image. [4]

1.4.1. Les procédures utilisées dans le système de traitement d'image : [6]

a) Les aspects des traitements d'images

Il est commode de subdiviser les différents algorithmes de traitements d'images en des sous classes.il existe différents algorithmes pour différentes tâches et problèmes et souvent nous tenons à distinguer la nature de la tâche à accomplir :

- Amélioration d'images
- Restauration d'images
- Segmentation d'images

Amélioration d'images : Il s'agit du traitement d'une image de sorte que le résultat est plus approprié pour une application particulière exemple : Sharpening, mettre en évidence les contours, amélioration du contraste, ou de l'éclaircissement d'une image et suppression de bruit

Restauration d'images : Cela peut être considéré comme inverser les dommages causés à une image par une cause connue. Exemple : Suppression de flou causé par le mouvement linéaire, Suppression des distorsions optiques et la Suppression des interférences périodique.

Segmentation d'images : Il s'agit de diviser une image en des éléments distinctifs, ou d'isoler certains aspects d'une image. Exemple : trouver des lignes, des cercles ou des formes particulières dans une image. Sur une photographie aérienne, identifier des voitures, arbres, bâtiments, ou des routes.

Ces classes ne sont pas disjointes, un algorithme donné peut être utilisé à la fois pour l'amélioration d'images ou pour la restauration d'images. Toutefois essayons de faire avec notre image simplement lui donner une meilleure apparence (amélioration), ou la suppression des dommages (restauration)

b) Les étapes du traitement d'images

- L'acquisition de l'image : Caméra CCD ou scanner.
- Prétraitement : Améliorer le contraste, la suppression du bruit et identifier les régions susceptibles que nous voulons prélever
- Segmentation : Extraction de la section déterminée.
- Représentation et description : Chercher des courbes, les trous et les coins qui nous permettrons de Distinguer les différentes formes de cette région extraie.
- Reconnaissance et interprétation : Attribuer des étiquettes à des objets en fonction de leurs Descripteurs (de l'étape précédente), et assigner des significations à ces labels

III. Les algorithmes de traitements d'images :

1. SVM (Support Vector Machine) : [7]

SVM (Support Vector Machine ou Machine à vecteurs de support) : Les SVMs sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie. Ils sont connus pour leurs solides garanties théoriques, leur grande flexibilité ainsi que leur simplicité d'utilisation même sans grande connaissance de data mining.

Les SVMs ont été développés dans les années 1990. Comme le montre la figure ci-dessous, leur principe est simple : ils ont pour but de séparer les données en classes à l'aide d'une frontière aussi « simple » que possible, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de la frontière.

2. OCR (La reconnaissance optique de caractères)

2.1.Définition du OCR : [8]

La reconnaissance optique des caractères, ou *Optical Character Recognition* – OCR en anglais, est une conversion électronique d'images textuelles dactylographiées, manuscrites ou imprimées. Ce texte est encodé par une machine dans un fichier de format texte.

Avec l'OCR, un grand nombre de documents papier peuvent être numérisés en texte lisible à la machine, peu importe la langue et le format dans lesquels ils sont rédigés. Cette technique facilite non seulement le stockage, mais rend disponibles des données qui auparavant étaient difficilement accessibles.

Il suffit simplement de penser à la quantité de données qui dorment dans des boîtes d'archives papier d'une ville ou d'un gouvernement par exemple.

Ces images et documents peuvent être numérisés sous forme de document texte, de photo de document ou de photo de scène (par exemple pour décoder le texte sur un panneau d'affichage).

2.2.Comment fonctionne l'OCR : [9]

Acquisition d'image : Un scanner lit les documents et les convertit en données binaires. Le logiciel OCR analyse l'image numérisée et classe les zones claires comme fond et les zones sombres comme texte.

Prétraitement : Le logiciel OCR nettoie d'abord l'image et prélève les erreurs pour la préparer à la lecture. Voici quelques-unes de ses techniques de nettoyage :

Désalignement ou inclinaison légère du document numérisé pour résoudre les problèmes d'alignement pendant la numérisation.

Déchatoisement ou prélèvement de toutes les taches de l'image numérique ou lissage des bords des images de texte.

Nettoyage des cases et des lignes dans l'image.

Reconnaissance d'écriture pour la technologie OCR multilingue

Reconnaissance de texte : Les deux principaux types d'algorithmes ou de processus logiciels qu'un logiciel OCR utilise pour la reconnaissance de texte sont appelés la correspondance de motifs et l'extraction de caractéristiques.

Correspondance de motifs : La correspondance de motifs fonctionne en isolant une image de caractère, appelée glyphe, et en la comparant à un glyphe stocké de manière similaire. La reconnaissance des formes ne fonctionne que si le glyphe stocké a une police et une échelle similaires au glyphe d'entrée. Cette méthode fonctionne bien avec les images numérisées de documents qui ont été tapés dans une police connue.

Extraction de caractéristiques : L'extraction de caractéristiques décompose les glyphes en caractéristiques telles que les lignes, les boucles fermées, la direction des lignes et les intersections de lignes. Il utilise ensuite ces caractéristiques pour trouver la meilleure correspondance ou le plus proche voisin parmi ses différents glyphes stockés.

Post-traitement : Après analyse, le système convertit les données textuelles extraites en un fichier informatisé. Certains systèmes OCR peuvent créer des fichiers PDF annotés qui comprennent les versions avant et après du document numérisé.

a) Quels sont les types d'OCR ?

Les scientifiques des données classent les différents types de technologies OCR en fonction de leur utilisation et de leur application. En voici quelques exemples :

- Logiciel de reconnaissance optique simple de caractères : Un moteur OCR simple
Cette solution présente des limites, car il existe un nombre pratiquement illimité de polices et de styles d'écriture, et chaque type ne peut être capturé et stocké dans la base de données.
- Logiciel de reconnaissance intelligente des caractères : Les systèmes OCR modernes utilisent la technologie de reconnaissance intelligente des caractères (ICR) pour lire le texte de la même manière que les humains. Ils utilisent des méthodes avancées qui entraînent les machines à se comporter comme des humains en utilisant des logiciels de machine Learning. Un système de machine Learning, appelé réseau neuronal, analyse le texte sur plusieurs niveaux en traitant l'image de manière répétée. Il recherche différents attributs de l'image, tels que les courbes, les lignes, les intersections et les boucles, et combine les résultats de tous ces différents niveaux d'analyse pour obtenir le résultat final. Même si l'ICR traite généralement les images un caractère à la fois, le processus est rapide, les résultats étant obtenus en quelques secondes.
- Reconnaissance intelligente des mots : Les systèmes de reconnaissance intelligente des mots fonctionnent sur les mêmes principes que l'ICR, mais traitent des images de mots entiers au lieu de prétraiter les images en caractères.
- Reconnaissance optique des marques : La reconnaissance optique des marques identifie les logos, les filigranes et autres symboles textuels dans un document.

2.3. Les avantages et les inconvénients du système OCR : [10]

a) Avantages :

- Les informations de l'OCR peuvent être lues avec un haut degré de précision. Les scanners à plat sont très précis et peuvent produire des images de qualité raisonnablement supérieure.
- Le traitement des informations OCR est rapide. De grandes quantités de texte sont souvent saisies rapidement.
- Un formulaire papier est souvent devenu un formulaire électronique facile à stocker ou à envoyer par courrier.

- C'est moins cher que de payer quelqu'un pour saisir manuellement une grande quantité de données textuelles. De plus, la conversion dans le formulaire électronique prend moins de temps.
- Le dernier logiciel peut recréer des array également sous forme de mise en page originale.
- Ce processus est beaucoup plus rapide que la saisie manuelle des informations dans le système
- La version avancée peut même recréer des tables, des colonnes et même produire des sites.

b) Inconvénients :

- Le texte OCR fonctionne efficacement avec le texte imprimé uniquement et non avec le texte manuscrit. L'écriture doit être apprise par le PC.
- Les systèmes OCR sont chers.
- Il y a le besoin de beaucoup d'espace requis par l'image produite.
- La qualité de l'image peut être perdue au cours de ce processus.
- La qualité de l'image finale dépend de la qualité de la première image.
- Tous les documents devaient être soigneusement vérifiés puis corrigés manuellement.
- Pas précis à 100%, il est probable que des erreurs soient commises au cours de la méthode.
- Cela ne vaut pas la peine de le faire pour de petites quantités de texte.

3. Python-tesseract :

Py-tesseract est un outil de reconnaissance optique de caractères (OCR) pour python. Autrement dit, il reconnaîtra et « lira » le texte intégré dans les images. Python-tesseract est un wrapper pour le moteur Tesseract-OCR de Google. Il est également utilisé comme un script individuel, car il peut lire tous les types d'images comme jpeg, png, gif, bmp, tiff, etc. De plus, s'il est utilisé comme script, Python-tesseract imprimera le texte reconnu plutôt que de l'écrire sur un fichier. Il a la capacité de reconnaître plus de 100 langues. [11]

3.1.Le fonctionnements interne du tesseract : [12]

Pour convertir une image en texte, Tesseract fonctionne en deux étapes principales. D'abord la segmentation de texte, puis la prédiction lettre à lettre.

La segmentation est un processus de traitement d'image. L'objectif étant de découper un document tout d'abord en paragraphes, puis en lignes, puis en lettres. Par un processus de transformation morphologique et de détection de contours, Tesseract obtient la position des blocs de texte.

Il s'agit ensuite d'obtenir la hauteur d'une ligne pour découper le paragraphe ligne par ligne, et, de la même manière, la largeur d'une lettre pour découper les lignes lettre à lettre.

A partir d'une collection d'images représentant des lettres individuelles et ordonnées, le moteur neuronal de Tesseract est activé. Il est composé d'un réseau de neurones effectuant l'analyse de l'image d'une lettre. Il permet de donner une estimation de quel caractère est représenté dans l'image.

Il contient également une série de cellules mémoires, qui lui permettent de se souvenir des caractères précédemment prédits. C'est comme cela que le réseau est capable d'estimer plus facilement des Caractères vraisemblables dans le langage utilisé dans le document.

Le fine-tuning de tesseract :

Ce système de prédiction de caractères avec mémoire est appelé LSTM (long short-term memory) et nécessite une phase d'entraînement sur des données réalistes et dont on connaît le contenu.

Pour générer de gros volumes de données pour l'apprentissage, Tesseract utilise une méthode astucieuse : pour une langue donnée, il possède un jeu de règles et un dictionnaire de mots. Le moteur est donc capable de créer des phrases similaires à du Français par exemple, et ceci en grande quantité. Les phrases sont ensuite transformées en images, avec différentes polices de caractères puis injectés dans le LSTM pour entraînement.

L'avantage de cette méthode est que l'on est certain du texte contenu dans les images, et on peut facilement guider le réseau de neurone lors de son entraînement.

Dans certains cas, nous souhaitons lire du texte qui ne ressemble pas forcément à des phrases complètes en Français, comme celles utilisées lors de l'entraînement du moteur. C'est le cas par exemple dans la lecture d'une bande MRZ de pièce d'identité, qui contient des séries de caractères encodés, ne ressemblant pas à du Français.

Tesseract nous permet de modifier son jeu de règles et la liste des mots utilisés pour la génération des données d'entraînement. Puis de relancer une phase appelée fine-tuning qui adapte la prédiction du LSTM à un problème plus précis. Cette approche permet d'intégrer les particularités d'un problème dans la phase d'apprentissage.

Tesseract-OCR a ouvert son code et a permis aux développeurs de l'améliorer en le modifiant, il est grâce à ça devenu un outil puissant et surtout adaptatif. Par exemple, via le processus de fine-tuning, il est possible d'adapter le moteur de reconnaissance de Tesseract pour réussir là où un outil fermé restera très généraliste et moins performant sur des cas d'application réelle.

IV. Etat de l'art :

1. Les dernières technologies utilisées pour détecter les matricules :

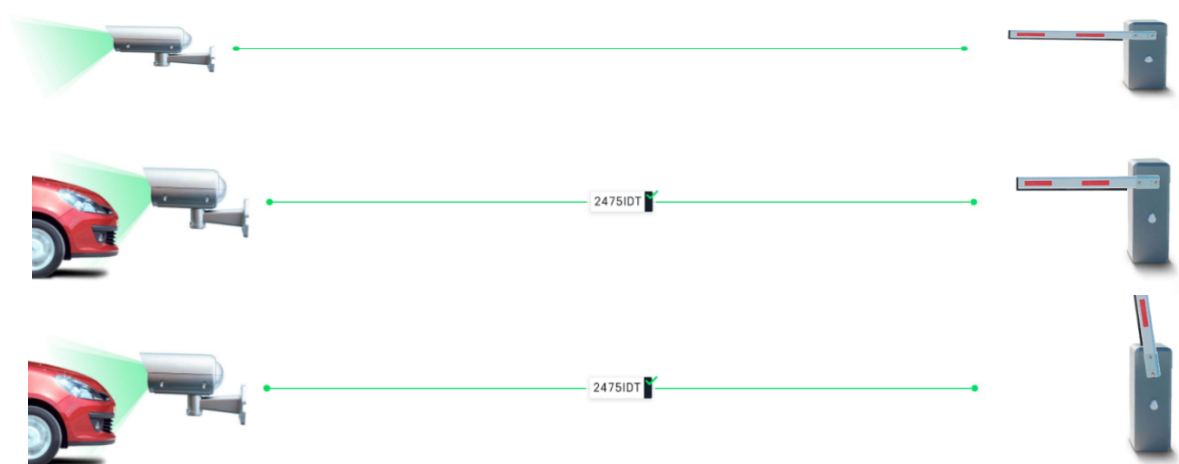


Figure 1.2 : les composants nécessaires dans les dernières technologies utilisées pour détecter les matricules [13]

3LPR : [13]

3LPR est un **système de reconnaissance de plaques d'immatriculation** (système LPR) utilisé pour identifier et enregistrer tous les véhicules qui accèdent ou sortent du parking, obtenant un grand contrôle des plaques d'immatriculation des véhicules et donc une sécurité accrue.

Le système de reconnaissance automatique de plaques d'immatriculation n'est pas seulement axé sur les parkings, mais peut être **utilisé dans toutes ces installations nécessaires pour contrôler**, surveiller et avoir un enregistrement de tous les véhicules qui passent par un certain accès. Exemple : les garages privés d'entreprises, de centres commerciaux, de péages, d'hôpitaux, etc.

Son objectif est de recueillir, de reconnaître et de stocker les photographies capturées par les lecteurs de plaques d'immatriculation (camera ANPR) placés sur chaque voie, dans le but de créer un registre d'accès et d'augmenter la sécurité du parking avec le contrôle de la plaque d'immatriculation.

Son fonctionnement est simple. Le logiciel LPR effectue une reconnaissance OCR de l'inscription, identifiant les caractères alphanumériques et stocke ces données dans la base de données.

LES AVANTAGES :

- **Fiabilité maximale** de la reconnaissance des plaques d'immatriculation.
- Système LPR adapté aux voitures, motos et camions.
- Augmente la sécurité et le contrôle des véhicules.
- Accès automatiques par enregistrement.
- Compatible avec les accès par téléphone mobile.
- Le logiciel LPR est adapté à plusieurs types de plaques d'immatriculation par pays.
- Fonctionnalités multiples : multi-prises, images sans mouvement, correction de la perspective...

LAPI :

LAPI La reconnaissance de plaques minéralogiques ou LAPI repose sur le procédé informatique de Reconnaissance Optique de Caractères (ROC). Ce procédé permet d'isoler et de capturer l'image d'une plaque minéralogique, d'en extraire les caractères alphanumériques et de les convertir en données numériques afin de les analyser et de les comparer avec une base de données.

Les algorithmes des logiciels LAPI sont en mesure de gérer la position du véhicule (orientation, distance, vitesse...), les problèmes d'éclairage et de contraste ainsi que les éventuelles barrières physiques (immatriculation couverte de poussière, police d'écriture fantaisie...). Leur vitesse d'exécution permet d'analyser un nombre élevé d'images prises en temps réel sur des axes de circulation à grande vitesse. [14]

Caractéristiques :

Le système LAPI est en mesure de posséder les caractéristiques suivantes : [15]

Chapitre 1 Technique de détection de matricules Dans les Parking intelligents

- ✓ Lecture et reconnaissance de plaques d'immatriculation des pays de l'Union Européenne et d'Afrique.
- ✓ Système automatique 24h/24 h et indépendant du matériel. La solution fonctionne avec tout type de caméras IP répondant à nos préconisations techniques.
- ✓ Interface simple à configurer disponible sur PC Windows et PC linux en 2 langues (Français/Anglais)
- ✓ Gestion des autorisations d'entrées et sorties

Possibilité de chiffrer les données (numéros de plaques et photos)

Les produits qui appliquent système LAPI : [15]

LAPI ACCES
LAPI-ENGINE
LAPI-SMARTPHONE

OpenALPR : [16]

OpenALPR est une bibliothèque open source de reconnaissance automatique de plaques d'immatriculation écrite en C ++ avec des liaisons en C #, Java et Python. Ce logiciel est distribué à la fois dans une version commerciale basée sur le cloud et dans une version open source.

OpenALPR utilise les bibliothèques OCR OpenCV et Tesseract. Il peut être exécuté comme un utilitaire de ligne de commande, une bibliothèque autonome ou un processus d'arrière-plan. Le logiciel s'intègre également aux systèmes de gestion vidéo (VMS) tels que Milestone XProtect. La bibliothèque analyse les images et les flux vidéo pour identifier les plaques d'immatriculation. La sortie est la représentation textuelle de tout caractère de plaque d'immatriculation.

Avantages : Le logiciel peut être utilisé de différentes manières. Par exemple, avec OpenALPR, vous pouvez :

- Reconnaître les plaques d'immatriculation des flux de caméras. Les résultats sont consultables et peuvent déclencher des alertes. Le référentiel de données peut être dans le cloud ou être entièrement stocké dans votre réseau sur site.
- Reconnaissez les plaques d'immatriculation des flux de caméras et envoyez les résultats à votre propre application .
- Traitez un fichier vidéo et stockez les résultats de la plaque d'immatriculation dans une base de données CSV et SQLite.
- Analyser des images fixes à partir de la ligne de commande
- Intégrez la reconnaissance de plaque d'immatriculation dans votre application directement dans le code (C / C ++, C #, VB.NET, Java, Python, Node.js)

Nous utiliserons le Tesseract OCR An Optical Character Recognition Engine (OCR Engine) pour reconnaître automatiquement le texte des plaques d'immatriculation des véhicules.

- ❖ **Nous nous avons utilisé dans notre projet les algorithmes de traitement d'image tesseract-OCR avec open cv intégrée dans notre Raspberry selon la programmation sur THONY**

Conclusion :

Enfin, nous avons pu voir un aperçu sur le smart parking basé sur l'IoT et la vision intelligente ainsi que les différents algorithmes de traitement d'image et les différents systèmes d'identification des plaques d'immatriculation disponible sur le marché.

Dans le chapitre suivant on va parler sur les différents outils nécessaires pour notre projet comme les matériels électroniques et les logiciels utilisées.

Chapitre 2 : Matériels, outils et logiciels utilisés

Introduction :

Il est toujours important d'avoir ce qu'il faut pour appréhender l'univers de l'électricité et l'électronique, pour assurer notre montage de notre réalisation on a besoin des **composants électroniques (tel que le Raspberry pi la caméra un capteur ultrason ...)** qui sont des pièces indispensables assemblés dans un circuit électrique (notre montage) afin d'assurer une fonction spécifique (le but). Dans ce chapitre, nous présentons l'environnement matériel et logiciel du projet.

I. Matérielles utilisée :

1) La carte raspberry pi 2

Raspberry est une carte mère d'un mini-ordinateur monocarte à processeur ARM qui peut être branchée à n'importe quel périphérique (souris, clavier...) comme illustre la Figure 2.1 . 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.

raspberry pi est couramment utilisé pour le traitement d'images/vidéos en temps réel, les applications basées sur l'IoT et les applications robotiques.

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 gratuite (GNU/Linux, Ubuntu, Archlinux, RISC OS, Windows 10 IOT Core, Raspbian Debian ...) et des autres logiciels compatibles [17]

Raspbian a une interface graphique qui comprend des outils pour la navigation, la programmation Python, le bureau, les jeux, etc.

Raspberry Pi est plus qu'un ordinateur car il permet d'accéder au matériel sur puce, c'est-à-dire aux GPIO, pour développer une application. En accédant au GPIO, nous pouvons connecter des appareils tels que des LED, des moteurs, des capteurs, etc. et nous pouvons également les contrôler.

Il dispose d'un SoC de processeur Broadcom basé sur ARM ainsi que d'un GPU (unité de traitement graphique) sur puce. [18]

Dans notre réalisation on a travaillé avec Raspberry pi 2 modelé B (Figure 2.2)

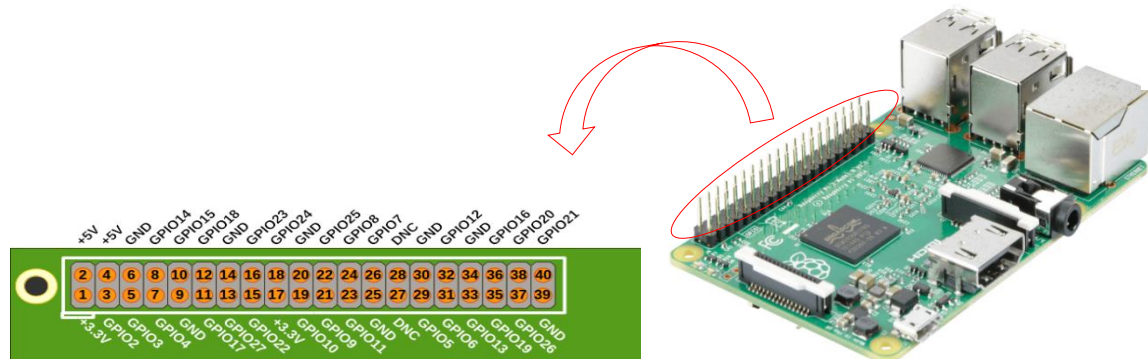


Figure 2.1 : les fonctions des pins du Raspberry pi 2 [19]

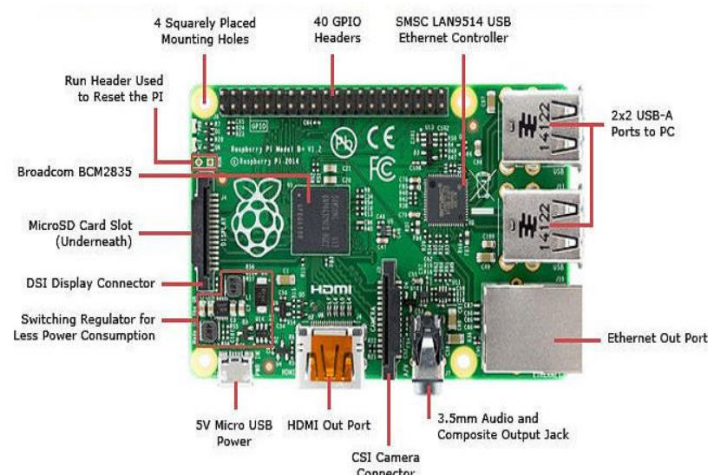


Figure 2.2: l'aspect extérieur du Raspberry pi [20]

Accessoire de Raspberry pi

Alimentation

Alimentation le Raspberry Pi s'alimente sous tension unique de 5 volts, tension sur laquelle il peut consommer jusqu'à 1.8A selon les tâches qu'il exécute. Cette alimentation doit être normalement fournie via le connecteur micro USB placé dans un angle de la carte. En effet, l'utilisation d'un chargeur pour Smartphone équipé d'un micro USB qui délivre une tension de 5 volts avec au minimum 1.8A est suffisante pour alimenter notre Raspberry Pi.

La carte SD

Pour héberger notre système d'exploitation (Raspbian) nécessaires au bon fonctionnement notre Raspberry Pi, et tous les fichiers nous auront besoin d'espace de stockage c'est pour cela qu'une microSD est requise. Au niveau de la taille, Raspbian étant très léger, il n'y a pas besoin d'une très grosse carte SD. 4Go suffisent, mais 16 Go sont préférables. Pour notre projet on a choisi une capacité de 16Go.

2) Raspberry Pi Camera (B) rev 2.0

Le module de caméra Pi est une caméra portable légère qui prend en charge Raspberry Pi (Figure 2.3). Il communique avec Pi en utilisant le protocole d'interface série de caméra MIPI. Il est normalement utilisé dans le traitement d'images, l'apprentissage automatique ou dans des projets de surveillance. Il est couramment utilisé dans les drones de surveillance car la charge utile de la caméra est très inférieure. En dehors de ces modules, Pi peut également utiliser des webcams USB normales qui sont utilisées avec un ordinateur. [21]

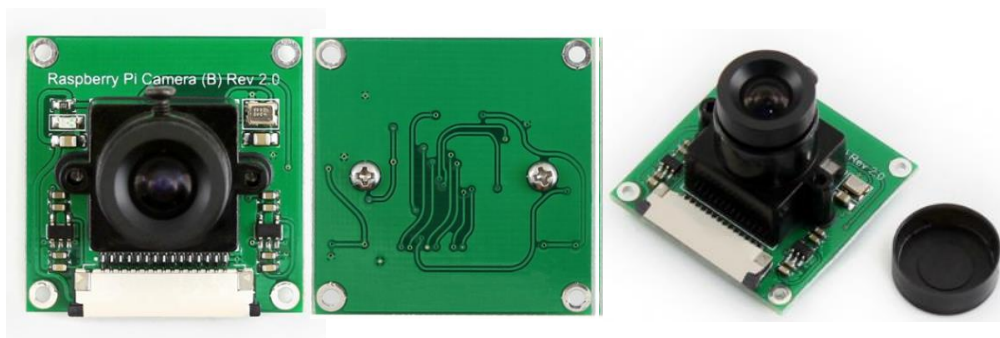


Figure 2.3: Raspberry Pi Camera Module, Adjustable-Focus [22]

Le module Pi Camera est un appareil photo qui peut être utilisé pour prendre des photos et des vidéos haute définition.

La carte Raspberry Pi possède une interface CSI (Camera Serial Interface) à laquelle nous pouvons attacher directement le module Pi Camera. Comme illustre la Figure 2.4

Ce module de caméra Pi peut se connecter au port CSI du Raspberry Pi à l'aide d'un câble ruban à 15 broches. [23]

Ce port se trouve près du port HDMI, connectez simplement le câble comme indiqué ci-dessous.

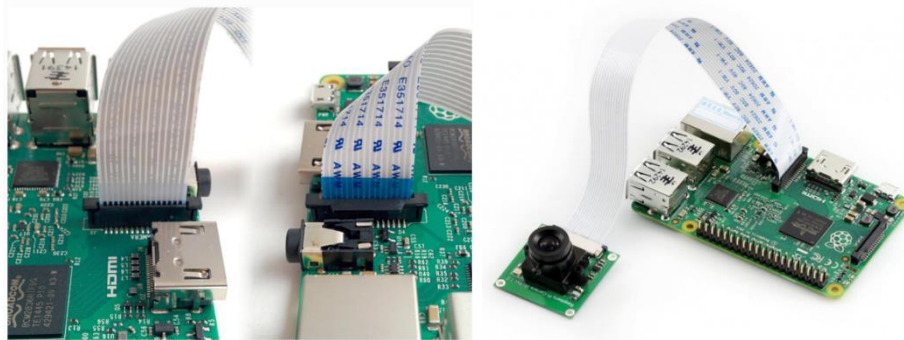


Figure 2.4 : Comment connecter le module caméra Pi au Raspberry Pi [24]

Caractéristique

- Caméra Raspberry Pi, prend en charge toutes les révisions du Pi
- Capteur OV5647 de 5 mégapixels
- Distance de mise au point réglable
- Spécifications de la caméra
 - Taille CCD : 1/4 pouce
 - Ouverture (F) : 2.0
 - Focale : 6mm
 - Angle de vue (diagonale) : 60,6 degrés
 - Meilleure résolution du capteur : 1080p
- Dimension : 32mm x32mm

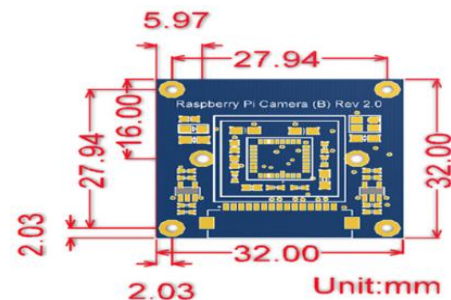


Figure 2.5: Dimensions de pi camera (B) rev 0.2[22]

3) Capteur ultrason HC-SR04

Le capteur de distance à ultrasons HC-SR04 est un capteur utilisé pour détecter la distance à un objet à l'aide d'un sonar. Il est idéal pour tous vos projets de robotique qui vous obligent à éviter les objets, en détectant à quelle distance ils se trouvent, vous pouvez vous en éloigner (Figure 2.5). Le HC-SR04 utilise un sonar à ultrasons sans contact pour mesurer la distance à un objet et se compose de deux émetteurs à ultrasons (essentiellement des haut-parleurs), d'un récepteur et d'un circuit de commande. Les émetteurs émettent un son ultrasonique à haute fréquence, qui rebondit sur tous les objets solides à proximité, et le récepteur écoute tout écho de retour. Cet écho est ensuite traité par le circuit de commande pour calculer la différence de temps entre le signal émis et reçu. Ce temps peut ensuite être utilisé, avec quelques calculs intelligents, pour calculer la distance entre le capteur et l'objet réfléchissant. [25]

Le ultrasons HC-SR04 peut être alimenté via la sortie 5V du Raspberry Pi

Caractéristiques du capteur de distance à ultrasons HC-SR04

Fréquence ultrasonique : 40 kHz

Tension de fonctionnement : 5 V (DC)

Courant statique : moins de 2 mA.

Signal de sortie : signal de fréquence électrique, haut niveau 5 V, bas niveau 0 V.

Angle du capteur : Pas plus de 15 degrés.
Distance de détection : 2 cm ~ 450 cm.
Haute précision : jusqu'à 3 mm
Mode de connexion : VCC / trig(T) / echo(R) / GND . [26]

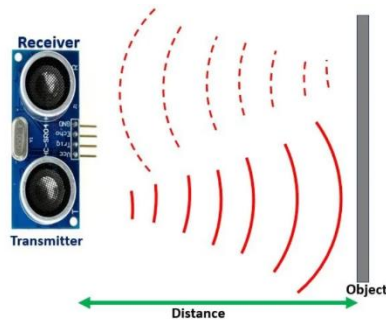


Figure 2.6: Fonctionnement de capteur ultrason [27]



Figure 2.7 : capteur ultrasons HC-SR04[28]

4) Les actionneurs

Les résistances
Les câbles mal-femelles
Les câbles mal-mal
Lab-d'essai
Cable ruban pour camera

II. logiciel utilisée :

1) Le module PIL de Python La bibliothèque PIL (Python Imaging Library)

Ajoute des possibilités de manipulation d'images de tout format avec la présence de quelques fonctions de traitement d'images. La documentation officielle de PIL est relativement complète. PIL vous a déjà été introduit en MPSI. Voici quelques fonctions très utiles à connaître : • En entrée : – Importation d'une image dans python, – Transformation en un tableau de nombres, – Copie « véritable » de la matrice avant sa modification. • En sortie : – Affichage de l'image, – Exportation au format matriciel .png pour sauvegarde de l'image modifiée. [29]

2) Opencv :

OpenCV est une bibliothèque de vision par ordinateur open source. La bibliothèque contient plus de 2500 algorithmes optimisés. Ces algorithmes sont souvent utilisés pour rechercher et reconnaître des visages, identifier des objets, reconnaître des décors et générer des marqueurs pour superposer des images en réalité augmentée, etc. [30]

3) Python :

Python est un langage de programmation multiparadigme, polyvalent, interprété et de haut niveau. Python permet aux programmeurs d'utiliser différents styles de programmation pour créer des programmes simples ou complexes, obtenir des résultats plus rapides et écrire du code presque comme s'ils parlaient dans un langage humain. [31]

- Les principales utilisations de Python par les développeurs sont :

La programmation d'applications

La création de services web

La génération de code

La métaprogrammation.

- On différencie deux versions : Python 2 et Python 3. Python 2, l'ancienne version propose des mises à jour jusqu'en 2020. Python 3 est la version actuelle. Son interpréteur est plus efficace, ainsi que son contrôle de concurrence. [32]

4) Fire base :

Firebase est le nom d'une plateforme mobile de Google qui facilite la création de back-end à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web.

L'objectif de la création de [Firebase.google.com](https://firebase.google.com) en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.

De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, Firebase SDKs a connu de nombreuses améliorations et n'a cessé de satisfaire ses utilisateurs. [33]

5) Thonny IDE sur Raspberry Pi OS :

Thonny IDE (Integrated Development Environment) est un excellent logiciel à utiliser lorsque vous écrivez des programmes Python sur Raspberry Pi OS.

En tant que débutant, cela vous aidera à vous concentrer sur la chose la plus importante : écrire du code.

Une fois que vous avez fait des progrès, cela peut vous rendre plus productif. Notez cependant qu'à un moment donné, il sera bon pour vous de comprendre chaque étape du processus, et de pouvoir les faire par vous-même : comment manipuler les fichiers et exécuter des scripts dans le terminal, comment utiliser différents éditeurs de texte (ex : Comment utiliser Vim sur votre Raspberry Pi), comment utiliser la commande pip pour gérer les modules Python, etc.

Et puis, au final, ce n'est qu'une question de préférences. Certaines personnes préfèrent utiliser un IDE, d'autres non. Il n'y a pas de bonne et de mauvaise solution. La meilleure chose que vous puissiez

Faire est d'essayer un peu de tout et de voir ce qui vous convient vraiment et vous rend le plus productif.

Démarrer Thonny IDE sur Raspberry Pi OS :

On Clique sur l'icône en haut à gauche représentant une framboise > Programmation > Thonny Python IDE. Comme illustre la Figure Figure 2.7

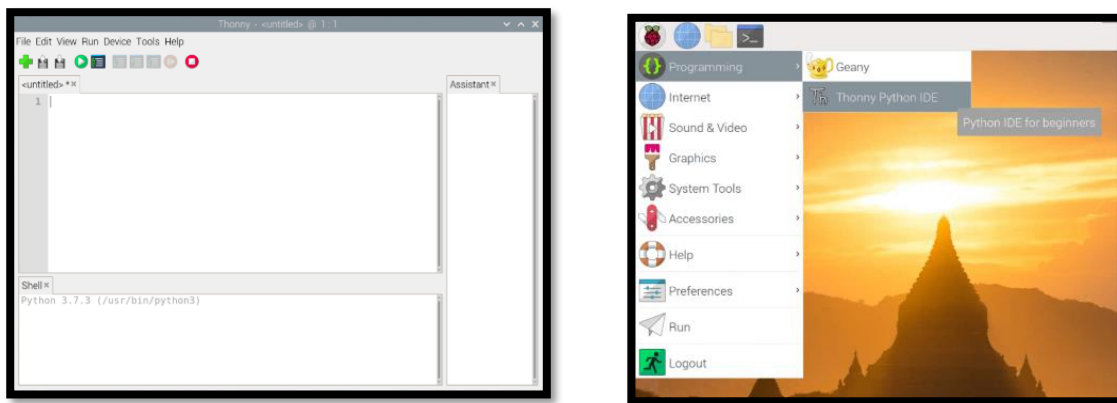


Figure 2.8 : Captur d'écran de l'interface et l'emplacement de Thonny Python IDE Sur notre Raspberry pi

6) Les commandes « les bibliothèques » :

- update & upgrade: `sudo apt-get update` and `sudo apt-get upgrade` are two commands you can use to keep all of your packages up to date in Debian or a Debian-based Linux distribution. They're common commands for Linux admins and people doing DevOps, but are handy to know even if you don't use the command line often. [34]
- `opencv-contrib-python==4.1.0.25`: OpenCV contrib est un module spécialisé présent dans le langage de programmation Python, qui est exclusivement nécessaire au système pour exécuter les descriptions de fonctionnalités SURF aux côtés du module OpenCV présent dans la bibliothèque open source. [35]
- `tesseract-ocr` : Tesseract is an open source optical character recognition (OCR) platform. OCR extracts text from images and documents without a text layer and outputs the document into a new searchable text file, PDF, or most other popular formats. Tesseract is highly customizable and can operate using most languages, including multilingual documents and vertical text. Although the software can be used on Windows or Linux, this guide will be based on Mac operating systems which is done through the terminal application. [36]
- `libtesseract-dev`: fichiers de développement de l'outil de ROC en ligne de commande de Tesseract. [37]
- `Pytesseract`: Pytesseract est un outil OCR pour python qui sert également de wrapper pour le moteur Tesseract-OCR. Il peut lire et reconnaître du texte dans des images et est couramment utilisé dans les cas d'utilisation d'image python ocr en texte. [38]

- libjpeg-dev : libjpeg est une bibliothèque gratuite avec des fonctions pour gérer le format de données d'image JPEG. Il implémente un codec JPEG (encodage et décodage) ainsi que divers utilitaires de gestion des données JPEG. [39]
- libtiff5-dev : libtiff est une bibliothèque prenant en charge le Tag Image File Format (TIFF), un format largement utilisé pour stocker des données d'image. Ce package comprend les fichiers de développement, la bibliothèque statique et les fichiers d'en-tête.[37]
- libjasper-dev : Fichiers de développement pour la bibliothèque JasPer JPEG-2000. [34]
- libpng-dev : Libpng est une bibliothèque qui implémente une interface pour lire et écrire des fichiers au format PNG (« Portable Network Graphics »). [37]
- libavcodec-dev : Cette bibliothèque fournit un cadriciel générique d'encodage/décodage et contient de nombreux décodeurs et encodeurs pour les flux audio, vidéo et de sous-titre, ainsi que plusieurs filtres de flux. [37]
- libavformat-dev : Cette bibliothèque fournit un cadriciel générique pour multiplexer et démultiplexer les flux audio, vidéo et de sous-titre. Elle englobe plusieurs multiplexeurs et démultiplexeurs pour les formats de conteneurs multimédia. Elle prend également en charge plusieurs protocoles d'entrée et de sortie pour accéder à une ressource multimédia. [37]
- libswscale-dev : Cette bibliothèque effectue des opérations de mise à l'échelle d'image et de conversion d'espace colorimétrique et de format de pixel hautement optimisées. Plus précisément, cette bibliothèque effectue les conversions suivantes :
 - ✓ Le redimensionnement est le processus de modification de la taille de la vidéo. Plusieurs remises à l'échelle. Options et algorithmes sont disponibles. Il s'agit généralement d'un processus avec perte.
 - ✓ La conversion du format pixel est le processus de conversion du format d'image et espace colorimétrique de l'image, par exemple du YUV420P planaire au RVB24 emballé. [37]
- libv4l-dev : Les bibliothèques libv4l sont un ensemble qui ajoute une mince couche d'abstraction au-dessus des périphériques video4linux2. Elle permet aux auteurs d'applications de gérer facilement des périphériques très divers sans avoir à écrire différents codes pour chacun des périphériques d'une même classe. libv4l consiste en trois bibliothèques différentes :
 - ✓ libv4lconvert fournit des fonctions permettant de convertir tous les formats de pixels (connus) en BGR24, RGB24, YUV420 et YVU420.
 - ✓ libv4l1 met en œuvre l'API v4l1 (obsolète) par-dessus des périphériques v4l2, indépendamment des pilotes de ces périphériques qui offrent une compatibilité v4l1 (ce que de nombreux pilotes v4l2 ne font pas).
 - ✓ libv4l2 fournit l'API v4l2 par-dessus des périphériques v4l2, tout en ajoutant pour l'application des conversions transparentes de libv4lconvert quand c'est nécessaire. [37]
- libxvidcore-dev : Xvid (anciennement XviD) est un codec vidéo MPEG-4 open source, implémentant les normes MPEG-4 Simple Profile, Advanced Simple Profile et Advanced Vidéo Coding. [37]
- libx264-dev : libx264 est une bibliothèque d'encodage avancée pour la création de flux vidéo H.264 (MPEG-4 AVC). [37]

- libfontconfig1-dev : Fontconfig est une bibliothèque de configuration et de personnalisation de police indépendante du système X Window. Elle a été conçue pour situer des polices dans le système et les choisir conformément aux besoins des applications. [37]
- libcairo2-dev : Cairo est une bibliothèque graphique multi-plateformes offrant un rendu vectoriel avec lissage des contours pour de multiples moteurs d'affichage. [37]
- libgdk-pixbuf2.0-dev : La bibliothèque GDK Pixbuf fournit des fonctions permettant de :
 - ✓ charger et enregistrer des images.
 - ✓ mettre à l'échelle et combiner les images en mémoire « pixbuf » .
 - ✓ charger des animations simples (p. ex. images GIF animées).
- libpango1.0-dev : Pango est une bibliothèque pour le placement et le rendu de texte, avec une attention particulière sur l'internationalisation. Pango peut être utilisé partout où il est nécessaire de disposer du texte. [37]
- libgtk2.0-dev & libgtk-3-dev : GTK+ est une boîte à outils multiplateforme pour créer des interfaces graphiques utilisateur. Elle offre un ensemble complet de composants graphiques et convient donc à tous les projets, depuis l'outil uni fonctionnel jusqu'aux suites logicielles complètes. [37]
- libatlas-base-dev : ATLAS est une approche de la génération automatique et de l'optimisation de logiciel de calcul numérique. [37]
- gfortran : Ce paquet fournit le compilateur GNU Fortran 95, qui permet de compiler du code Fortran 95 sur toutes les plate-formes gérées par le compilateur gcc. Il utilise le moteur de gcc pour créer du code optimisé. [37]
- libhdf5-dev & libhdf5-serial-dev & libhdf5-103 : HDF5 est un format de fichier et une bibliothèque pour stocker des données scientifiques. HDF5 a été conçu et implémenté pour combler les lacunes de HDF4.x. Il dispose d'un modèle de données plus puissant et flexible, prend en charge les fichiers de plus de 2 Go et prend en charge les E/S parallèles. [37]
- libqtgui4: Qt est un cadre applicatif C++ multiplate-forme. Sa principale caractéristique est son grand nombre de composants qui fournissent les fonctionnalités classiques d'une interface graphique. [37]
- libqtwebkit4 : QtWebKit fournit un moteur de navigateur Web qui facilite l'intégration de contenu du World Wide Web dans votre application Qt. [37]
- libqt4-test : Qt est un framework d'application C++ multiplateforme. La principale caractéristique de Qt est son riche ensemble de widgets qui fournissent des fonctionnalités d'interface graphique standard, Le module QtTest fournit des classes pour les tests unitaires des applications et bibliothèques Qt. [37]
- python3-pyqt5 : PyQt5 met à disponibilité l'API de Qt5 à Python 3. [37]
- libjasper1: bibliothèque d'exécution JasPer JPEG-2000. [40]
- pyrebase : Pyrebase est une interface Python pour l'API REST de Firebase. En termes simples, il vous permet d'utiliser Python pour manipuler votre base de données Firebase.[41]

Conclusion :

Dans ce chapitre, tout d'abord on a présenté les différents outils utilisés pour développer notre réalisation. Nous avons discuté sur les composants de détermination de notre réalisation. Enfin, nous avons conclu l'importance et l'efficacité de ses composants pour notre projet, ensuite comment nous avons utilisé ces composants pour produire notre réalisation sera abordé dans le prochain chapitre.

Chapitre 3 :

Conception et réalisation

Introduction :

Après l'étape de l'étude préalable et la spécification des besoins, nous réservons cette partie à la conception et la réalisation détaillée de notre système. En effet, au cours de ce chapitre nous allons décrire les différentes étapes de conception de la partie statique (des données), ainsi que celles de la partie réalisation du montage.

I. Partie conception :

1) Étape 1 : Téléchargement et installation de Raspberry pi imager :

Nous avons téléchargé Raspberry Pi Imager depuis [42]. Il existe des versions pour Windows, Mac, Ubuntu et Raspberry Pi OS lui-même.



Figure 3.1: le site officiel pour télécharger Raspberry pi imager [42]

2) Étape 2 : Gravure l'image du système d'exploitation Raspberry Pi sur la carte SD :

- Avec une carte SD fraîchement formatée, nous pouvons maintenant passer à l'installation du système d'exploitation.
- On a cliqué sur le bouton "Choisir un système d'exploitation" et on a sélectionné l'un des systèmes d'exploitation disponibles. Pour les besoins de ce didacticiel, nous utiliserons « 2021-05-07-raspbian-buster-armhf.zip »
- Maintenant on a sélectionné la carte SD que nous venons de formater à l'étape précédente.
- L'option "Ecrire" deviendra disponible. On a cliqué sur "Ecrire" pour commencer à graver l'image sur la carte SD.
- Cela peut prendre de quelques minutes à plus d'une demi-heure pour que le processus se termine en fonction de la qualité et de la vitesse de la carte SD, du lecteur de carte et de l'ordinateur.
- Une fois que Raspberry Pi Imager a fini d'écrire les fichiers sur la carte SD, il vérifie que l'image sur la carte SD est identique au fichier image utilisé pour graver l'image. Cela prend généralement moins d'une minute, mais peut prendre plus de temps.

3) Étape 3 : Configuration de notre Raspberry Pi

- On a inséré la carte microSD dans la fente pour carte sous le Raspberry Pi.
- On a branché le clavier USB et la souris USB dans l'un des ports USB.
- On a allumé notre moniteur ou notre téléviseur et on assure qu'il est réglé sur la bonne entrée (par exemple, HDMI 1 ou Composant).
- On a branché le câble HDMI ou composant vidéo sur le moniteur ou le téléviseur.
- Si on a prévoyez de nous connectée à Internet, on doit connectée l'adaptateur Wi-Fi au Raspberry Pi.
- On a connecté l'alimentation au Raspberry Pi.



Figure 3.2 : alimentation de Raspberry pi 2 [43]

Raspberry Pi peut également être connecté à un ordinateur :

- Étape 1 : Obtenir l'adresse IP de notre carte.

Nous avons cliqué sur l'icône framboise puis on a navigué dans > les préférences > configuration Raspberry pi puis on a cliqué sur l'onglet > interfaces et on activée SSH et VNC maintenant nous devrions pouvoir l'icône VNC en haut à droite on a cliqué dessus et on note l'adresse IP que nous voyons dans la fenêtre.

- Étape 2 : Installation de VNC viewer sur notre ordinateur portable.

Pour connecter la carte à notre ordinateur portable nous avons besoin du VNC serveur et de VNC viewer nous avons déjà activé le serveur VNC sur la carte dans la dernière étape, de notre ordinateur portable on allons à [44] et on fait défiler vers le bas et télécharger la VNC viewer et installer l'application.

- Étape 3 : Run VNC viewer. Depuis notre ordinateur portable après l'installation de VNC viewer on tape l'adresse IP de notre carte et on clique sur entrée et on entre notre mot de passe 'raspberrypi' pour nous connecter



Figure 3.3 : Captur d'écran d'Association de Raspberry Pi à notre ordinateur portable à l'aide de la VNC viewer

4) Étape 4 : Installation des librairies nécessaires pour notre projet :

Nous-avons ouvert le terminal, et on tape les commandes suivantes :

- `sudo apt-get update`
- `sudo apt-get upgrade`
- `sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev libgtk-3-dev libatlas-base-dev gfortran libhdf5-dev libhdf5-serial-dev libhdf5-103 libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5 libjasper1`
- `sudo pip3 install opencv-contrib-python==4.1.0.25`
- `sudo apt install tesseract-ocr`
- `sudo apt install libtesseract-dev`
- `sudo pip3 install pytesseract`
- `sudo pip install Pyrebase4`

II. Réalisation :

1) Etape 7 : Le montage :

a) Simulation

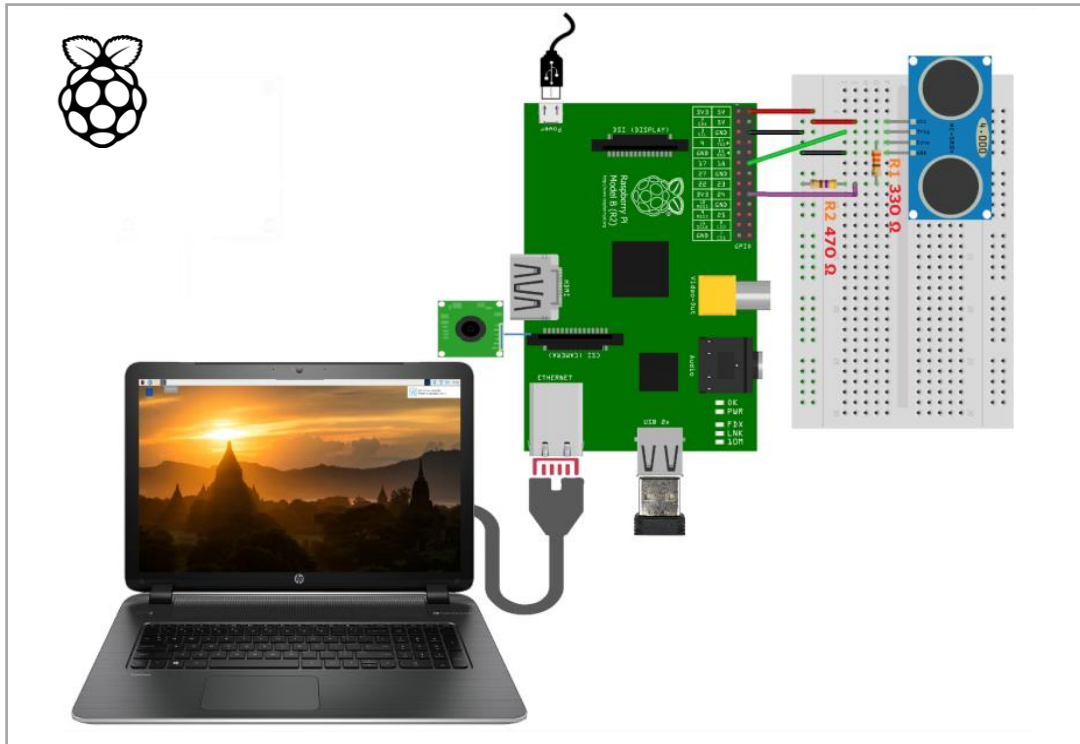


Figure 3.4 : simulation

b) Montage reel

- Cette photo représentée la photo réelle de notre montage Raspberry pi connecte a une caméra et un ultrason

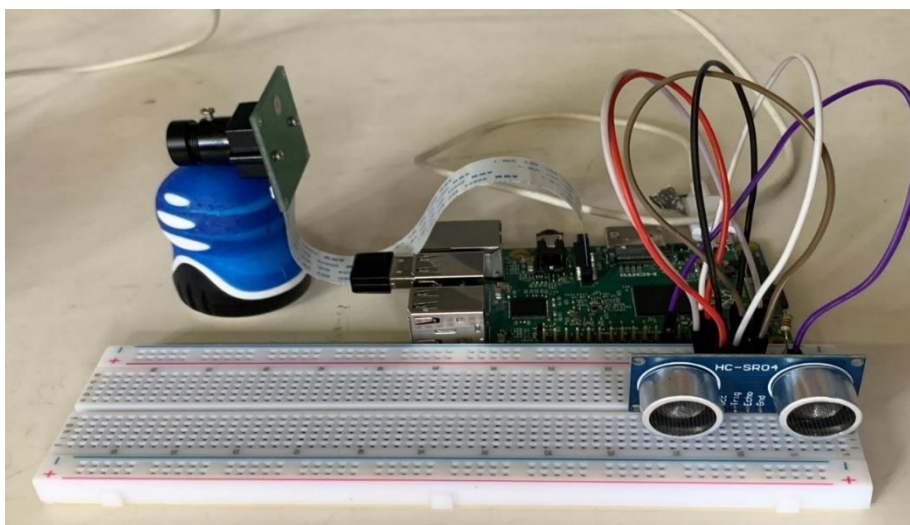


Figure 3.5 : Montage réel

- Photo réel montre la prises et la capture du matricule avec la caméra Raspberry pi

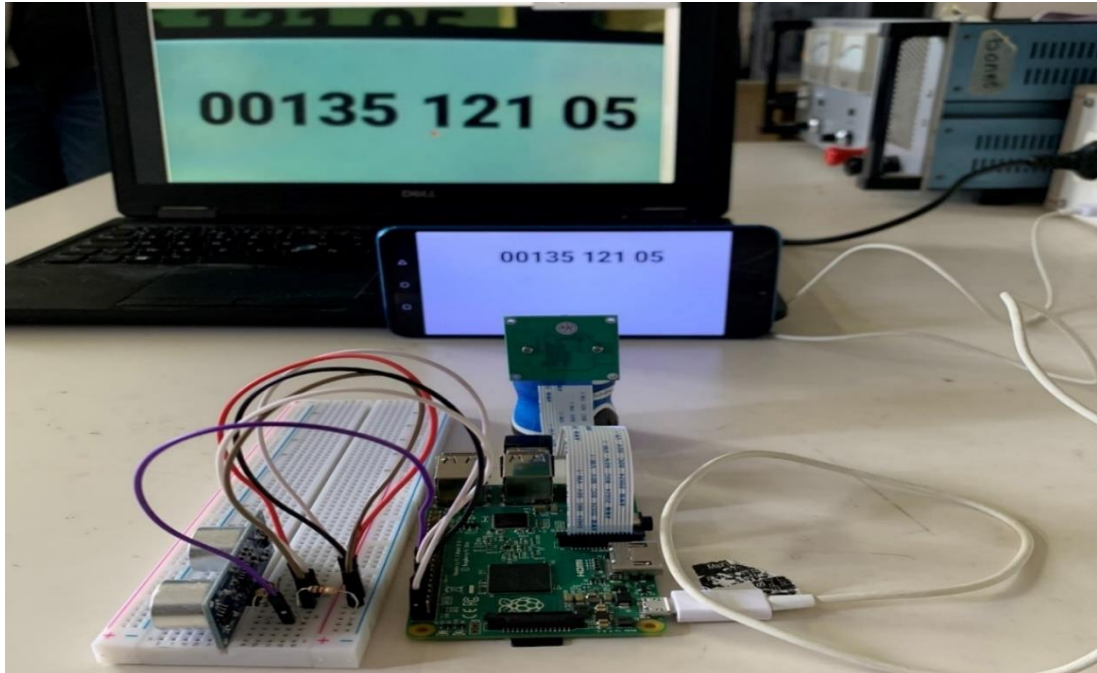


Figure 3.6 : le prise du matricule par le Raspberry pi camera

- Le traitement d'image et la reconnaissance du matricule capturée

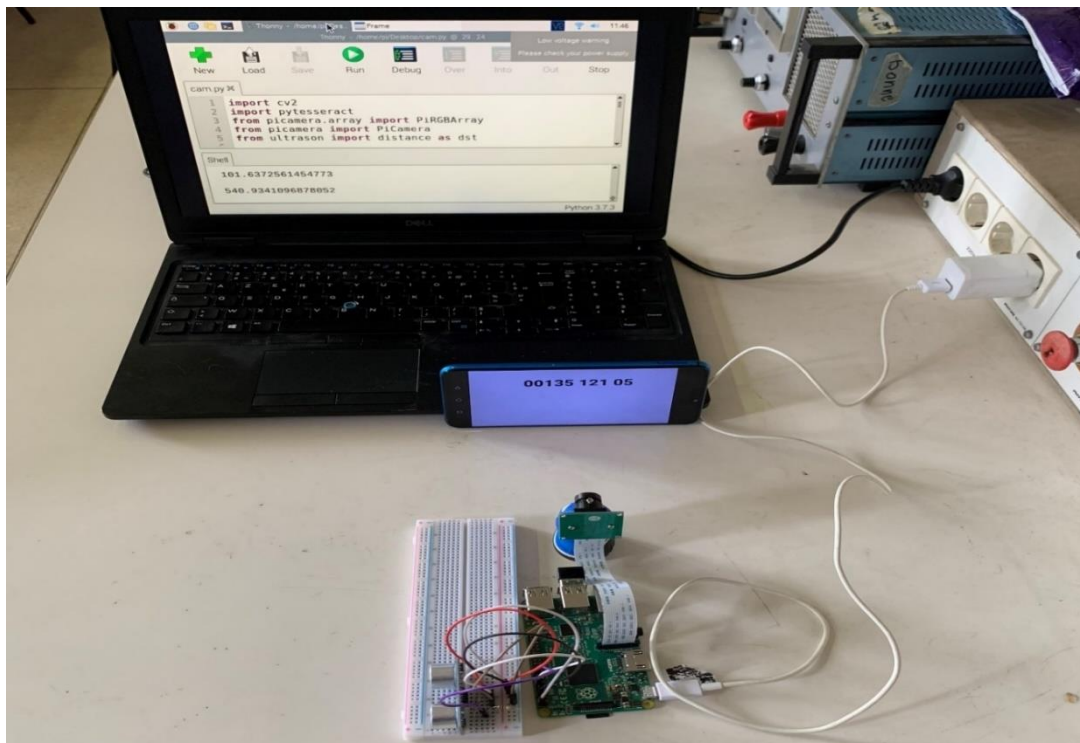


Figure 3.7 : La reconnaissance du matricule

- L'enregistrement du matricule dans la base de données (Firebase)

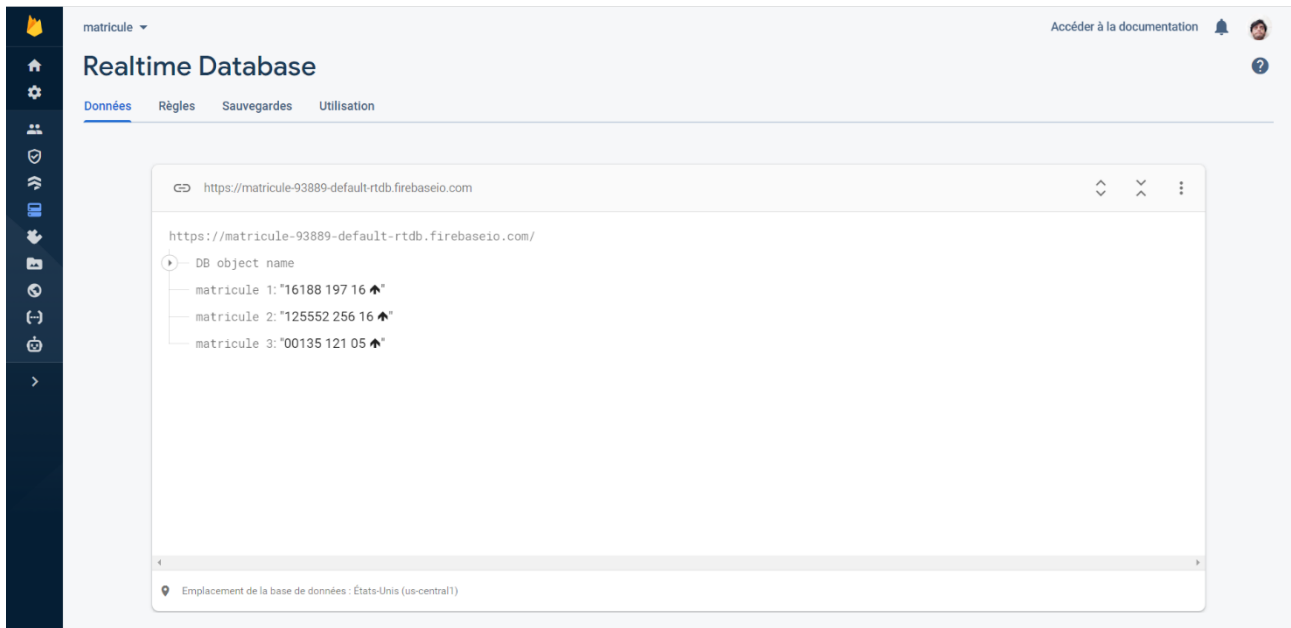


Figure 3.8 : les matricules enregistrées dans la base de données

2) Programation :

Etape 8: Programme

For the programme contact the web owner

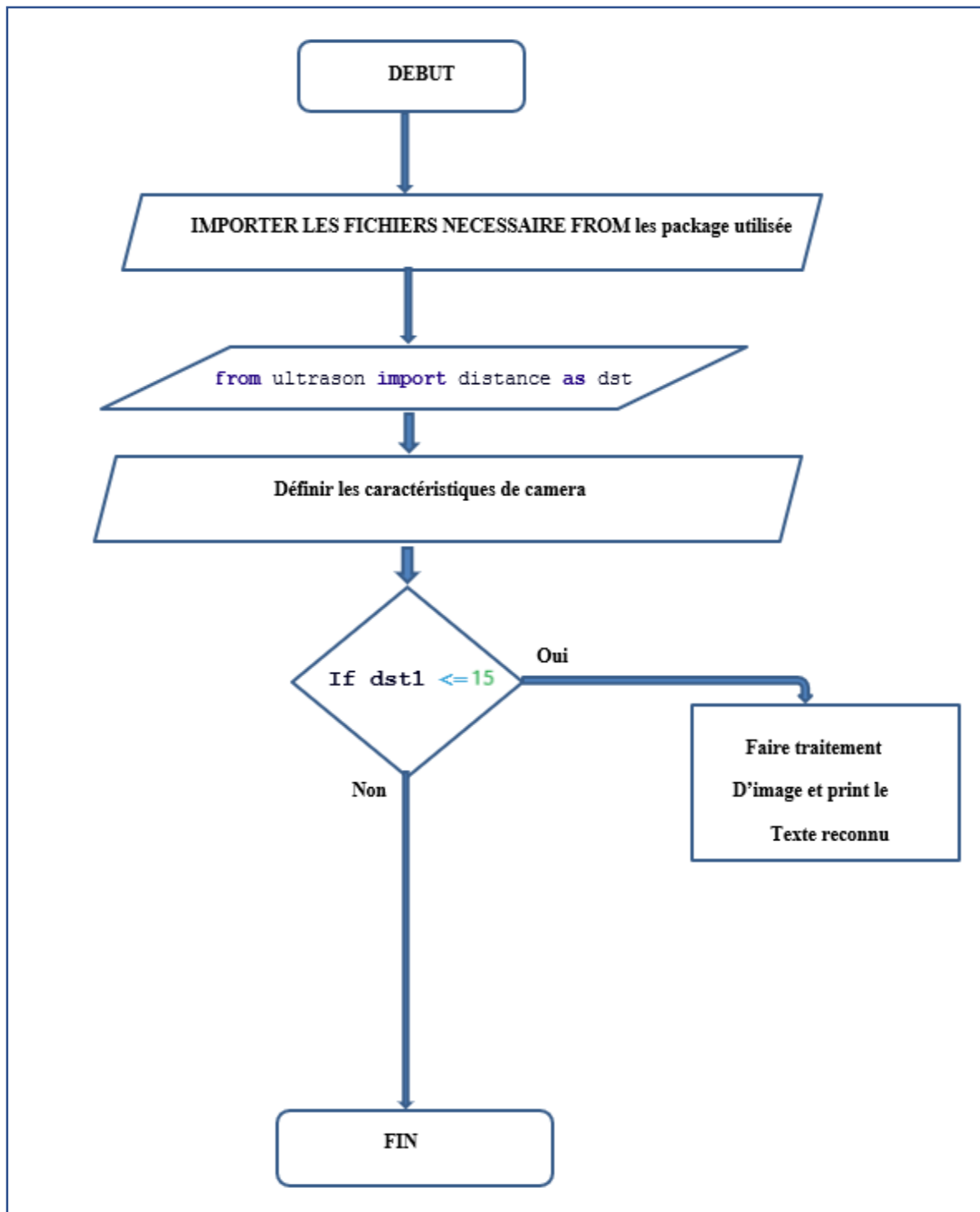


Figure 3.9: organigramme

Conclusion:

Au cours de ce chapitre, nous avons utilisé l'environnement du travail matériel et logiciel avec lequel nous avons pu implémenter cette application. Nous avons expliqué les étapes de réalisation de notre système en présentant quelques captures d'écrans et des photos réels de réalisation. En achevant ce chapitre, nous nous approchons de la fin de ce mémoire qui sera clôturée par une conclusion générale.

Conclusion Général :

Ce travail représente une étude sur les systèmes de reconnaissance automatique des plaques d'immatriculation suivi d'une conception et une implémentation de tel système sur un ordinateur à puce « Raspberry Pi » en tant qu'un système embarqué.

Premièrement nous avons exposé des généralités sur le domaine de traitement des images et les systèmes RAPI existants dans le marché afin de nous inspirer pour notre projet. Après nous avons choisi un système à implanter, avec des outils matériels et logiciels bien choisi pour un système RAPI efficace et performant.

Du côté implémentation matériel nous avons utilisé une carte Raspberry Pi 2 – B, du fait que cet ordinateur nous offre la possibilité de choisir les outils informatiques voulus pour la programmation, dont on a implémenté le système en utilisant le langage de programmation Python avec la bibliothèque de vision par ordinateur OpenCV ce qui a facilité le procédé d'implémentation.

Les résultats de reconnaissance obtenus montrent l'impact de plusieurs facteurs pour avoir une bonne localisation de la plaque et une bonne reconnaissance des chiffres de la plaque lui-même, dont on peut citer les suivants : Le bon éclairage au moment de la capture, les normes qui doivent être respectés pour la construction de la plaque, le choix des bons facteurs et des bons seuils pour les algorithmes implémentés, faire des captures de proche pour les véhicules.

Malgré les efforts déployés pour améliorer les résultats obtenus on n'arrive pas à une détection de 100% des plaques des immatriculations et pour arriver à améliorer ce travail nous pouvons citer qu'il est possible de faire plusieurs autres expériences pour arriver à fixer les bons facteurs et les bons seuils ou utiliser des méthodes plus performant de reconnaissance comme le deep learning dans un prochain Inchalah.

BIBLIOGRAPHIE :

- [1] - La gestion intelligente des parkings sur le site <https://hxpérience.com/la-gestion-intelligente-des-parkings/>
- [2] - S. AOuat « Vision par ordinateur », Cours de master2SII, Département informatique, USTHB, 2012.
- [3] - Traitement d'images, http://fr.wikipedia.org/wiki/Traitement_d'images consulté 12/10/2015
- [4] - PDF cours image processing1, Master : Systèmes des Télécommunications, Université des frères Mentouri (Constantine 1)
- [5] - chapitre 10 traitement d'image sur le site : <https://stephane-bonnaud.fr/wp-content/uploads/2021/03/Chapitre-10.pdf>
- [6] - Abdelouahed Abdou sabri , «Introduction au traitement d'images» , slideshare, 5 juin 2014
- [7] Data analyse post, SVM sur le site : <https://dataanalyticspost.com/Lexique/svm/>
- [8] comprendre la reconnaissance optique des caractères sur le site : <https://moov.ai/fr/blog/reconnaissance-optique-de-caracteres-ocr/>
- [9] Qu'Est se que OCR sur le site <https://aws.amazon.com/fr/what-is/ocr/>
- [10] les avantages et inconvénients du OCR sur le site <https://fr.acervolima.com/avantages-et-inconvenients-du-lecteur-optique-de-caracteres-ocr/>, Septembre 2021
- [11] L reconnaissance des plaques d'immatriculations sur le site : [https://fr.acervolima.com/reconnaissance-de-plaque-dimmatriculation-avec-opencv-et-tesseractocr/#:~:text=Nous%20utiliserons%20le%20Tesseract%20OCR,caract%C3%A8res%20\(OCR\)%20pour%20python.](https://fr.acervolima.com/reconnaissance-de-plaque-dimmatriculation-avec-opencv-et-tesseractocr/#:~:text=Nous%20utiliserons%20le%20Tesseract%20OCR,caract%C3%A8res%20(OCR)%20pour%20python.)
- [12] RENAUD. «5 min pour comprendre L'OCR», 7 Janvier 2020,Partie 2
- [13] © 2022 i+D3 Equipamientos Tecnológicos SL, « La reconnaissance automatique des matricles 3LPR »
- [14] 12 rue des Marquis - PA Les Marches de Bretagne
Saint-Hilaire-de-Loulay85600 MONTAIGU, « VS Sécurité » , Décembre 2019
- [15] 2013-2022 Alphanumerique vision, « LAPI ACCES Reconnaissance des plaques d'immatriculations » , December 2019.
- [16] - Site web : <http://doc.openalpr.com/>
- [17] Mémoire de fin d'études Master, Conception et Réalisation d'une Maison Intelligente via Raspberry Pi 3 BouaklineNassim_BenchabaAghiles.pdf ,01 octobre 2018

Bibliographie

- [18] - Documents sur présentation du Raspberry pi,2022 sur le site : <https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction>
- [19] DI/DSI solutions d'entreprise, 2000-2022 developpez, « raspberry pi python et le port GPIO»
- [20] Michel Vonlanthen HB9AFO, « Raspberry Pi 3 B+ », le 14 janvier 2022
- [21] components101, « Pi Camera Module – 5MP 9 », December 2018
- [22] RPi Camera (B), Adjustable-Focus, waveshare.com
- [23] ElectronicWings, « Pi Camera Module Interface with Raspberry Pi using Python »
- [24] Ben Nuttall (Alumni), « How to build projects using the Raspberry Pi camera », June 6, 2016
- [25] - PiBorg. « Capteur de distance à ultrasons (HC-SR04) » in : <https://www.piborg.org/>
- [26] - « Capteur à Ultrason HC-SR04 » sur le site : <https://www.twinschip.com/HC-SR04-Ultrasonic-Sensor>
- [27] HC-SR04 Ultrasonic Sensor Interfacing with Arduino – Distance Measurement Example, sur le site <https://microcontrollerslab.com/ultrasonic-hc-sr04-sensor-arduino-tutorial/>
- [28] 2015 Orbit. Tous droits réservés, Conception & hébergement Kdconcept. Adresse:43,Cite des Moudjahidines Chevalley BenAknoun ALGER, « HC-SR04 CAPTEUR ULTRASON MODULE »
- [29] chapitre 10 traitement d'image sur le site : <https://stephane-bonnaud.fr/wp-content/uploads/2021/03/Chapitre-10.pdf>
- [30] « reconnaissance-de-plaque-dimmatriculation-avec-opencv-et-tesseractocr », sur le site <https://fr.acervolima.com>
- [31] techopedia, «What Does Python mean » May 28, 2019
- [32] JDN, « Python : définition et utilisation de ce langage informatique »,le 31 MARS 2020
- [33] Junto, «A quoi sert Firebase, la plateforme mobile de Google? », 25 avril 2019
- [34] Freecodecamp , « sudo apt-get update vs upgrade – What is the Difference? », MAY 2, 2022
- [35] site web : <https://www.educba.com/opencvcontrib>
- [36] NYU Returns: What to know about using NYU Libraries for Spring 2022 ,« Tesseract OCR Software Tutorial »
- [37] <https://packages.debian.org/>

Bibliographie

- [38] Filip Zelic & Anuj Sable , « How to OCR with Tesseract, OpenCV and Python » Mai 2022
- [39]<https://en.wikipedia.org/>
- [40] UbuntuUpdates.org, « packages »,2022-04-21
- [41] Hackbright Academy, « How to Integrate Firebase Into Your App », Francisco in 2012.
- [42] site officiel du Raspberry pi : <https://www.raspberrypi.com/software/>
- [43] Raspberry.ma , « Alimentation officielle Raspberry Pi 4 5.1V 3A »
- [44] « Remote access software built into Raspberry Pi » sur le site : www.realvnc.com