

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان -

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Electronique

Spécialité : Instrumentation

Par: - Mlle. BENSAPHLA TANI Kheira & Mr. BEREKSI Mohammed Karim

Sujet

Réalisation d'un système autonome de contrôle d'accès de véhicules par reconnaissance optique des plaques d'immatriculation

Soutenu en Juin 2022, devant le jury composé de :

Mr NEMMICHE AHMED	MCB	Université de Tlemcen	Président
Mr BOUANATI SIDI MOHAMMED	MAA	Université de Tlemcen	Examinateur
Mr MOULAI KHATIR AHMED NASSIM	MCB	Université de Tlemcen	Encadrant

Année universitaire : 2021 / 2022

Remerciements

*Nous tenons tout d'abord à remercier dieu pour tout ce qu'il nous a offert, ainsi que notre Encadrant et cher Professeur qui nous a poursuivi et aider à réaliser notre projet de fin d'études et même pendant notre parcours d'études lors des années précédentes. Nous exprimons nos sincères sentiments de respect et gratitude. Un grand merci à vous **Mr MOULAI KHATIR Ahmed Nassim**.*

*Nous tenons à remercier, aussi, **Mr. NEMMICHE Ahmed**, Maître de Conférences Classe B à l'université de Tlemcen, pour l'honneur, de bien vouloir accepter de présider le jury de notre soutenance.*

*Nos remerciements sont, également, adressés à **Mr. BOUANATI Sidi Mohammed**, Maître Assistant Classe A à l'université de Tlemcen, d'avoir accepté d'examiner ce travail et de nous faire l'honneur de participer au jury*

Nous remercions tous nos professeurs de la promotion d'option Instrumentation.

Nous remercions nos familles ainsi que nos amis communs, avec qui nous avons passé de bons moments et pleins de souvenirs ensemble. On vous remercie pour vos encouragements, soutien et amour.

Dédicace

Je dédie ce travail, que j'ai fait de mon mieux pour le réaliser, à moi-même et je tiens à me remercier pour mes efforts fournis durant mon parcours universitaire, ainsi qu'à ma famille qui m'a encouragé et soutenu pendant toute ma vie pour arriver là où je suis.

*Je tiens à témoigner mes sincères remerciements à ma mère et mon père qui m'ont aidé depuis mon très jeune âge dans mes études et dans chaque décision dans ma vie, qui ont fait toujours de leurs mieux pour réaliser mes rêves. Merci Mama **Radia** pour tes sacrifices, Merci Papa*

Fethi.

Merci à toi ma sœur Nesrine et mes deux petits frères Ilyes et Mehdi. Et je remercie mon ami Adel RAHMOUNE qui m'a beaucoup aidé dans ce projet.

Et je tiens à remercier mon binôme et mon ami Karim et mes autres ami(e)s qui m'ont accompagné la durée de ces années universitaires.

Je vous aime tous.

BENSAHLA TANI Kheira

Dédicace

*Tout d'abord je tiens à me remercier moi même pour chaque effort
fournit durant ses cinq longues années et surtout ses deux dernières qui
étaient de loin les plus faciles.*

*je tiens aussi à remercier mes chères parents pour leurs aides que ce soit
du côté psychologique ou bien du côté financier et aussi mes deux chers
sœurs et à tout la famille bereksi et benayad*

*je tiens à remercier aussi chaque personne ayant été là durant ses
années que ce soit mes ami(e)s, mon binôme Bensahla Tani Kheira ou
bien des personnes de mon entourage.
mes vœux les plus sincères, cordialement.*

Karim Bereksi

Sommaire

Introduction Générale.....	1
Chapitre I :.....	3
Généralités sur la reconnaissance optique de caractères.....	3
I.1. Introduction.....	4
I.2. OCR (Optical Character Recognition) :	4
I.2.1. Historique :	4
I.3. Fonctionnement d'un système OCR :.....	5
I.3.3. Reconnaissance des caractères :.....	6
I.3.4. Post-traitement :	6
I.4. L'utilisation d'OCR :	7
I.5. La lecture des plaques d'immatriculation :	8
I.5.1. Les plaques d'immatriculation en Algérie :	8
I.5.2. Méthodes existantes pour la détection des plaques l'immatriculation :	9
I.5.3. Méthode d'extraction des régions d'intérêt :	9
I.5.4. Méthode basée sur la texture :	10
I.5.5. Méthode basée sur le contour	10
I.6. Domaines et exemples d'utilisation des systèmes de détection des plaques d'immatriculation :	11
I.6.1. Le péage urbain :	11
I.6.2. L'accès au parking.....	11
I.7. L'image :.....	12
I.7.1. Définition de l'image :	12
I.7.2. Représentation d'une image :	12
I.7.3. La différence entre une image matricielle ou vectorielle	13
I.8. Le traitement de l'image :	13
I.8.1. Le prétraitement d'une image :.....	14
I.8.2. L'analyse d'une image :	14
I.8.3. La localisation de la plaque d'immatriculation.....	15
I.8.4. La détection de contour :	15
I.8.5. Opération morphologique :.....	17
I.8.6. Méthodes basées sur les caractéristiques de l'image :.....	17
Conclusion du chapitre I :	17
Chapitre II :	18
Descriptions matérielle et logicielle du système.....	18

II.1. Introduction :	19
II.2. Le Raspberry PI :	19
II.2.1. Description du Raspberry pi :	19
II.2.2. Historique :	20
II.2.3. Raspberry PI 3 Modèle B :	21
II.2.3.1. Les composants du Raspberry PI3 :	22
Connecteurs :	22
4xport USB :	23
Port camera CSI :	23
Port d'écran DSI :	23
Port HDMI :	24
Micro Card slot :	24
Audio et vidéo composite :	24
Broadcom BCM2837 :	24
Circuit d'alimentation :	25
L'alimentation électrique :	25
Port RJ45 :	25
Les voyants lumineux vert et rouge :	25
II.2.4. Les étapes d'installation et de configuration du système d'exploitation :	26
II.2.5. Connexion SSH :	27
II.2.6. Connexion VNC :	29
II.3. Python :	30
II.3.1. Pourquoi on a choisi python :	30
II.3.2. Les bibliothèques python :	30
II.4. Mini servomoteur 9G SG90:	32
Conclusion du chapitre II :	32
Chapitre III :	33
Mise en œuvre du système :	33
III.1. Introduction :	34
III.2. Méthodes de détection des plaques d'immatriculation :	34
III.2.2. Technique basée sur les propriétés de la plaque :	34
III.2.2. Technique basée sur l'intelligence artificielle :	35
III.2.3. Technique basée sur la signature de la plaque :	35
III.2.4. Technique basée sur les contours de la plaque :	36
III.3. Matériels utilisés :	36
III.4. Composition du programme python :	38

III.5. Conversion csv to PHP:.....	47
III.6. La maquette réalisée :.....	51
Conclusion du Chapitre III :	52
Conclusion Générale.....	53
Conclusion générale :	54
Bibliographie :.....	55
Résumé :.....	58
Abstract:	58
:ملخص.....	58

Liste des figures :

Figure I.1 : Etapes d'une reconnaissance optique des caractères.....	6
Figure I.2 : Plaque d'immatriculation en Algérie.....	9
Figure I.3 : Péage urbain.....	11
Figure I.4 : Système d'accès automatique au parking	11
Figure I.5 : Différence entre une image matricielle ou vectorielle	13
Figure I.6 : Application du filtre Canny	16
Figure II.1: Classement des différents types du Raspberry.....	20
Figure II.2: Plaque électronique.....	21
pour les premiers tests du Raspberry.....	21
Figure II.3: Anatomie du Raspberry Pi 3B	21
Figure II.4: GPIO d'un Raspberry Pi B 3.....	22
Figure II.5: Port camera CSI.....	23
Figure II.6: Port d'écran DSI.....	24
Figure II.7: Broadcom BCM2837.....	25
Figure II.8: SDFormatter V4.0	26
Figure II.9: Raspberry Pi imager	26
Figure II.10: L'installation du Raspbian	27
Figure II.11: La configuration du Raspberry	27
Figure II.12: La configuration des connexions.....	28
Figure II.13 : L'activation du SSH.....	28
Figure II.14: Putty	29
Figure II.15: L'activation de VNC.....	29
Figure II.16: VNC.....	30
Figure II.17: Exemple d'une fonction de la bibliothèque OpenCV.....	31
Figure II.18: Principe du module Pytesseract	32
Figure II.19: Mini servomoteur 9G SG90.....	32
Figure III.1: Système RAPI.....	34
Figure III.2: Technique basée sur les propriétés de la plaque	35
Figure III.3: Technique basée sur les contours de la plaque	36
Figure III.4: Webcam utilisée	37
Figure III.5: Branchement du Raspberry.....	38
Figure III.6: Conversion l'image en échelle de gris.....	40
Figure III.7: Réduction du bruit et application du filtre Canny.....	40
Figure III.9: Exemple 2 d'une détection de plaque d'immatriculation.....	47
Figure III.10: Base de données base2.csv.....	50
Figure III.11: Page web affichée pour quelques exemples de plaques d'immatriculation	50
Figure III.12: VNC Viewer	51
Figure III.13: Maquette réalisée pour notre système.....	52

Liste des abréviations :

OCR : Optical Character Recognition

CSI: Camera Serial Bus

DSI: Display Serial Interface

SDFormatter: Secure Digital Formatter

GPIO: General Purpose Input/Output

SSH: Secure Shell

VNC: Virtual Network Computing

CSV: Comma Separated Values

USB: Universal Serial Bus

IMR: Intelligent Machines Research

ASCII: American Standard Code for Information Interchange

LED: Light Emitting Diode

SDRAM: Synchronous Dynamic Random Access Memory

ARM: Advanced RISC Machines

RAM: Random Access Memory

GPU: Graphics Processing Unit

IBM: International Business Machines

BBC: British Broadcasting Corporation

PC: Personal Computer

EEPROM: **E**lectrically-**E**rasable **P**rogrammable **R**ead-**O**nly **M**emory

HDMI: High-Definition Multimedia Interface

VGA: Video Graphics Array

ADSL: Asymmetric Digital Subscriber Line

RAPI: Reconnaissance Automatique des Plaques d'Immatriculation

CMOS: Complementary Metal Oxide Semi-conductor

BGR: Blue Green Red

PHP: Hypertext PreProcessor

HTML: HyperText Markup Language

API: Application Programming Interface

Introduction Générale

Introduction générale :

Le parc automobile en Algérie est apprécié à plus de 6.4 millions de véhicules en 2018, le numéro d'immatriculation est nécessaire pour l'identification de tous les véhicules de manière unique vu le grand nombre de voitures en circulation au niveau mondial.

Le but principal de notre projet est l'étude et la conception d'un système de contrôle d'accès aux parkings par reconnaissance automatique de plaques d'immatriculation à l'aide d'un Raspberry Pi, d'où trois objectifs ont été visés :

- a) Faire une étude générale sur la théorie d'OCR.
- b) Regrouper suffisamment d'informations sur l'une des cartes de prototypage: Raspberry Pi.
- c) Réaliser un système de contrôle d'accès par reconnaissance de plaques d'immatriculation à l'aide des composants disponibles.

Afin de réaliser ce projet, nous avons adopté deux parties principales : partie matérielles et partie logicielles. La partie matérielle est basée sur le mini-ordinateur incroyable qui a été lancé en 2012, ainsi ses accessoires. La deuxième partie est à propos des logiciels: Le système d'exploitation ainsi que le langage de programmation Python et ses bibliothèques.

Ce Manuscrit est réparti en trois chapitres :

Le premier chapitre est une brève présentation générale sur la reconnaissance optique de caractères, son utilisation dans la détection des plaques d'immatriculation ainsi que les domaines d'utilisation.

Le deuxième chapitre présente les différents outils matériels et logiciels nécessaires à la conception de notre système.

Le troisième chapitre est consacré à la conception et la réalisation pratique du système sous un prototype expérimentale pour montrer les différentes fonctionnalités. Enfin, nous allons terminer par une conclusion générale qui résume l'essentiel du travail réalisé.

Chapitre I :
**Généralités sur la reconnaissance
optique de caractères**

I.1. Introduction

Il existe près de 1,4 milliard de véhicules à moteur, et à quatre roues ou plus, en circulation dans le monde. Les estimations disent que ce nombre pourrait doubler d'ici 2040.

En Algérie, le parc automobile est apprécié à plus de 6.4 millions de véhicules en selon les statistiques de 2018. En raison du grand nombre de voitures en circulation au niveau mondial, le numéro d'immatriculation est indispensable pour identifier tous les véhicules de manière unique.

Le numéro d'immatriculation est constitué d'une combinaison unique de caractères alphanumériques. Il est destiné à identifier facilement et de façon unique n'importe quel véhicule

Les systèmes de Reconnaissance Automatique de Plaques d'Immatriculation (RAPI) sont apparus pour bien gérer le grand flux de voitures en circulation, la reconnaissance optique de caractère présente le principe de base de ces systèmes

I.2. OCR (Optical Character Recognition) :

Un système OCR (Optical Character Recognition en anglais) ou ROC (Optical Character Recognition) en français est un système qui permet de numériser un texte ou des documents, qui peuvent ensuite être édités sur un Smartphone ou un ordinateur.

Le système OCR correspond à la reconnaissance automatique du texte imprimé, ainsi qu'à la transcription en fichiers électroniques. En scannant le document, l'appareil est ainsi capable de "lire" le contenu.

Les systèmes OCR peuvent reconnaître différents types de polices et différents caractères des machines à écrire ainsi que des ordinateurs. Certains systèmes OCR peuvent même reconnaître l'écriture manuscrite.

Le texte lu par un Smartphone ou un ordinateur sur un document numérisé peut être utilisé pour remplir automatiquement des formulaires. [1]

I.2.1. Historique :

Gustav Tauschek, ingénieur allemand, en 1929 a créé la première machine d'OCR qui contenait un détecteur photosensible qui allumait une lumière lorsque le détecteur est sur un mot. En 1950, Frank Rowlett qui est un cryptanalyste de l'Agence de Sécurité des Forces Armées, prédécesseur de NSA (National Security Agency) a réglé le problème de la conversion de messages en langage de machine, alors David Shepard, qui est cryptanalyste aussi, a pensé qu'ils doivent construire une machine pour le faire, et grâce à l'aide de Harvey

COOK, le « Gismo » a été construit.

En 1951, Washington Daily News et en 1953, le New York Time ont publié le fait après le dépôt du brevet, donc shepard a créé le IMR (Intelligent Machines Research corporation) qui livra les systèmes OCR au monde.

Depuis 1965, la poste des Etats unis utilisait le système OCR pour trier le courrier.

Depuis 1971, la poste canadienne commence aussi l'utiliser.

Le système OCR lit le nom et l'adresse et fait le tri automatiquement et imprime le code postal et le code barre sur l'enveloppe.

Ray Kurzweil en 1974, un scientifique, avait créé un synthétiseur vocal qui permet aux aveugles de lire des documents enregistrés.

L'année 1976, Styvie Wonder finança le projet pour enfin finir le prototype. [2]

I.3. Fonctionnement d'un système OCR :

Les systèmes OCR utilisent les dernières technologies pour recueillir des informations à partir de vos documents numérisés (texte, photos) et les convertir en fichiers texte.

Pour ce faire, le système OCR compare les couleurs noir et blanc du document pour déterminer chaque code alphanumérique. Le système reconnaît ensuite chaque caractère et le convertit en texte ASCII (American Standard Code for Information Interchange).

Par exemple, cela vous permet de modifier, de rechercher et de copier du texte dans Word aussi rapidement que possible. [3]

Les étapes de traitement sont ainsi :

I.3.1. Pré-analyse de l'image :

Cela sert à améliorer la qualité de l'image, ainsi que le redressement de l'image et le passage en mode bicolore (noir et blanc), la détection de contours. [4]

I.3.2. Segmentation :

C'est l'analyse de page, il isole les lignes des caractères dans l'image, ainsi que la détection du texte. [4]

I.3.3. Reconnaissance des caractères :

Après l'étape de normalisation, la comparaison se fait entre l'instance de reconnaissance et la bibliothèque des formes connues, l'étape suivante sera trouvée la forme proche, selon les caractéristiques distance et vraisemblance. [4]

Voici les types des techniques de reconnaissance :

Classification par caractéristique :

Est une reconnaissance qui est représentée par un vecteur des valeurs numériques, les features sont classés, une classe de caractère c'est comme si classer une lettre qui sera présentée par un nuage, dans le but de déterminer à quelle classe de caractère. La forme appartient au plus vraisemblable.[4]

Méthode métrique :

Compare la forme à reconnaître au moyen d'algorithme de distance, c'est le type le moins efficace quelques autres méthodes.[4]

Méthode statique :

Pour la reconnaissance d'écriture manuscrite généralement utilisé pour les méthodes (probabilités/statistiques) comme les chaînes de Makrov. [4]

I.3.4. Post-traitement :

Utiliser des méthodes linguistiques et contextuelles pour réduire le nombre d'erreurs : systèmes à base de règles ou méthodes statiques. Et dans les systèmes industriels, des techniques spécialisées pour les zones du texte. [4]

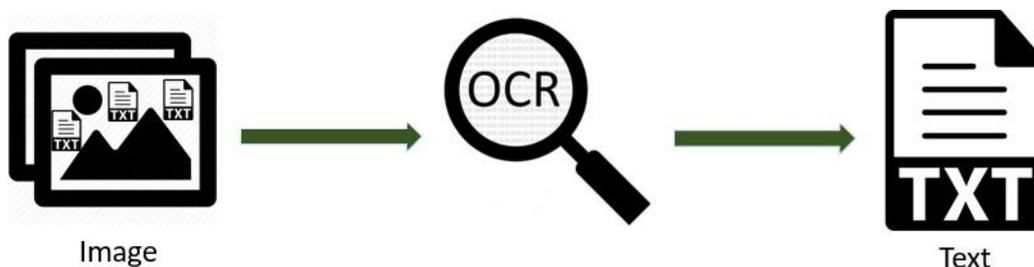


Figure I.1 : Etapes d'une reconnaissance optique des caractères

I.4. L'utilisation d'OCR :

Le système OCR est une technologie polyvalente qui peut être utilisée dans diverses situations : documents officiels, ou mêmes jeux occasionnels.

Il y en a même d'autres cas à utiliser :

Saisie automatique de données : chèques, passeports, factures, etc...

Reconnaissance automatique des plaques d'immatriculation : (et c'est le sujet de notre projet)

Reconnaissance des passeports : dans un aéroport par exemple

Extraction automatique d'informations clés

Extraction des informations de carte d'affaires

Numérisation de gros documents imprimés : des livres, des mémoires de fin d'études...

Conversion de l'écriture manuscrite en temps réel :

Par exemple, les centres de tri du courrier utilisent aujourd'hui souvent des systèmes OCR pour gérer et trier le courrier. Cela permet un tri plus rapide et des délais de livraison améliorés. ... [5]

Le système est également populaire dans les applications pour Smartphone. Il fonctionne pour différentes fonctions : numériser des reçus ou numériser et reconnaître des documents manuscrits. [6]

Prétraitement : La plupart des logiciels OCR prétraite l'image pour augmenter les chances de reconnaissance.

Les techniques de prétraitement comprennent :

1. Réalignement :

Il peut avoir besoin d'être tourné de quelques degrés dans le sens horaire ou antihoraire pour s'assurer que les lignes de texte soient parfaitement. [6]

2. Déparasitage :

Supprimer les taches sur les photos et lisser les bords. [6]

3. Binarisation :

Convertir une image en noir et blanc La tâche de binarisation est effectuée comme un moyen facile et précis de distinguer le texte de l'arrière-plan. [6]

4. Suppression de la ligne :

Nettoyer les boîtes et les lignes non glyphs. [6]

5. Zonage :

Détecte les colonnes, les paragraphes, les légendes, sous forme de blocs. Particulièrement utile dans les mises en page et les tables multi colonnes. [6]

6. Détection de lignes et de mots :

Établir des formes avec les mots et les caractères de base, diviser les mots au besoin. [6]

7. Reconnaissance du script :

Le script peut se transformer au niveau des mots. Donc l'identification du script est essentielle avant que l'OCR puisse être utilisé pour gérer un script en particulier. [6]

8. Segmentation :

Divers caractères liés par des artefacts d'images doivent être divisés. [6]

9. Normalisation :

Normaliser le ratio des dimensions de l'image et l'échelle. [6]

I.5. La lecture des plaques d'immatriculation :

La lecture des plaques d'immatriculation se fait à l'aide de la reconnaissance optique de caractères (OCR) par une prise d'image de caméra. C'est une méthode de surveillance généralement utilisé dans : des parkings, entreprises, maisons,. Cette lecture se manipule d'un ordinateur pour extraire les caractères numériques (pour les matricules de l'Algérie). [7]

I.5.1. Les plaques d'immatriculation en Algérie :

La plaque d'immatriculation en Algérie se compose de 10 Chiffres, le matricule permet d'identifier les véhicules.

La plaque d'immatriculation est constituée de 3 groupes de chiffres séparés par un espace, voici les 3 groupes :

- 1^{er} groupe : 5 chiffres et à la wilaya d'Alger 6 chiffres, et c'est le numéro de dossier de la voiture.
- 2^{ème} groupe : 3 chiffres et chaque chiffre à une signification :
 1. Le 1^{er} chiffre : le type de véhicule :
 - ✓ 1 : véhicule particulier
 - ✓ 2 : les camions
 - ✓ 3 : les camionnettes
 - ✓ 4 : les véhicules de transport ou un autobus

- ✓ 5 : tracteur routier
 - ✓ 6 : un autre type de tracteur
 - ✓ 7 : véhicule spécial
 - ✓ 8 : une remorque
 - ✓ 9 : une moto
2. Les 2 autres chiffres représentent l'année de mis en circulation du véhicule
- Le 3^{ème} groupe : 2 chiffres qui représentent la wilaya d'immatriculation de 01 à 58 selon la dernière mise à jour des wilayas en 2022. [8]

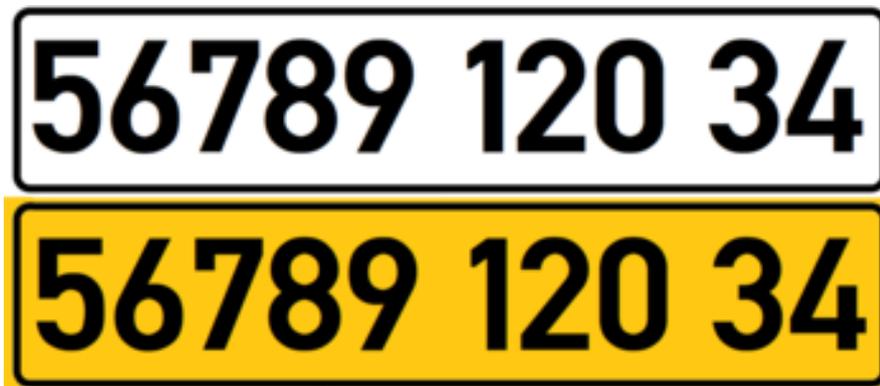


Figure I.2 : Plaque d'immatriculation en Algérie

I.5.2. Méthodes existantes pour la détection des plaques d'immatriculation :

Les humains peuvent reconnaître des objets sans trop d'effort. Cependant, d'accord Les objets dans les images sont une tâche très difficile en vision par ordinateur.

La détection des plaques d'immatriculation est un défi majeur dans les images de véhicules.

En raison de différentes tailles de plaques, de différents styles et de différentes conditions de prise de vue Éclairage lors de l'acquisition d'images du véhicule. Les différents systèmes de détection de plaque d'immatriculation proposés sont basés sur différentes propriétés. Certaines techniques utilisent une approche simple basée sur des règles déterministes, tandis que d'autres utilisent des systèmes d'apprentissage et de classification. [9]

I.5.3. Méthode d'extraction des régions d'intérêt :

La région d'intérêt dans une image donnée est la plaque d'immatriculation. Cette région se trouve par application d'un concept pour une image donnée, la région contenant une plaque d'immatriculation devra avoir un nombre maximum de bords par rapport à toute autre partie dans une image.

L'application de ce concept à tous les segments extraits, les coordonnées de la région requise sont extraites. Ces valeurs de coordonnées sont ensuite utilisées pour extraire la plaque

d'immatriculation. [9]

I.5.4. Méthode basée sur la texture :

La texture est une fonctionnalité qui peut être utilisée pour détecter les plaques d'immatriculation en visibilité directe

Les caractères sur la plaque ont une texture similaire. Dans cet algorithme, nous traitons des caractères

Faites du tableau une texture distincte du reste des objets inclus dans l'image. Le chemin est Divisé en quatre étapes suivantes :

- 1) Analyse de la texture des images ;
- 2) Décomposition d'images multi-segments ;
- 3) Sélection de masques
- 4) Analyse des composantes de corrélation. [9]

I.5.5. Méthode basée sur le contour

Des conditions de luminosité inégales et des changements de distance entre la caméra et le véhicule peuvent affecter les résultats de la détection de la plaque d'immatriculation. À cette fin, il existe une méthode pour détecter les plaques d'immatriculation en fonction des caractéristiques de contour et des attributs des plaques d'immatriculation.

L'algorithme proposé est basé sur les caractéristiques suivantes Les pixels avec des caractères de plaque d'immatriculation ont généralement des valeurs de contraste plus élevées par rapport aux pixels voisins.

Le contour des caractères d'une plaque est toujours un contour fermé. Il y a une relation de voisinage entre les personnages.

De plus, cette méthode basée sur les contours et les gradients se compose de cinq processus :

- Détection des contours
- Sélectionnez la zone de caractère de la plaque d'immatriculation candidate
- Calculer l'amplitude du gradient des composants connectés
- Extraction de caractères
- Emplacement de la plaque d'immatriculation. [9]

I.6. Domaines et exemples d'utilisation des systèmes de détection des plaques d'immatriculation :

I.6.1. Le péage urbain :

Le système de péage urbain fonctionne sous l'analyse des plaques d'immatriculation enregistrées par un réseau de caméras de surveillances disposées en entrées et sorties des zones de péage des villes, autoroutes, ...etc. il compare ensuite les données recueillies avec la base de données centralisée pour vérifier dans les délais de paiement effectif. Ce type de système permet de réduire et faciliter la circulation.



Figure I.3 : Péage urbain

I.6.2. L'accès au parking

C'est presque le même principe du péage urbain, l'analyse des plaques d'immatriculations se fait avec une caméra, puis le système fait une comparaison avec la base de données pour donner l'accès au parking.



Figure I.4 : Système d'accès automatique au parking

I.7. L'image :

I.7.1. Définition de l'image :

L'image est obtenue en transformant la scène réelle par le capteur, elle est donc Signal 2D (x, y), représentant généralement la réalité 3D (x, y, z). Sous un certain angle Mathématiquement, une image est une matrice de nombres représentant un signal, et il existe plusieurs outils Manipulez ce signal. Pour les ordinateurs, une image est un ensemble de pixels. Un pixel est un élément d'image. [10]

I.7.2. Représentation d'une image :

Une image numérique à visualiser sur un écran d'ordinateur est Il existe deux catégories principales : les images matricielles et les images vectorielles. [10]

I.7.2.1. Image matricielle :

Les images matricielles (images bitmap) sont constituées de collections de points ou de pixels. Lorsque ces images sont imprimées ou utilisées pour l'impression (photos, publicités, cartes...etc.), on parle de points, on parle de pixels d'une image stockée sous forme "binaire" ou "numérique". Une image est constituée d'une matrice (tableau) de points à plusieurs dimensions, chaque dimension représentant une dimension spatiale (hauteur, largeur, profondeur), temporelle (durée), ou autre (par exemple, niveau de résolution). [10]

I.7.2.2. Image vectorielle :

Une image vectorielle informatique (ou image en mode ligne) est une image numérique composée d'un seul objet géométrique (segment de ligne, polygone, arc, etc.), chacun défini par diverses propriétés telles que la forme, la position, la couleur, etc... (Définition mathématique). Par exemple, un diagramme vectoriel d'un cercle est défini par des propriétés de type : position du centre, rayon, etc. Essentiellement, la carte vectorielle est redessinée à chaque fois qu'elle est visualisée, générant des calculs sur la machine. L'intérêt est de pouvoir redimensionner l'image à volonté sans aucun effet d'escalier. L'inconvénient est que pour obtenir une qualité photo-réaliste, il faut beaucoup de puissance de calcul et beaucoup de mémoire. Les avantages de ce type d'image sont qu'elle peut être agrandie à l'infini sans perte de qualité initiale, et qu'elle a un faible encombrement. [10]

I.7.3. La différence entre une image matricielle ou vectorielle

La principale différence entre les deux formats est que les images vectorielles peuvent être agrandies sans perdre leur qualité, tandis que les images matricielles perdent leur netteté lorsqu'elles sont agrandies. [10]



Figure I.5 : Différence entre une image matricielle et une image vectorielle

I.8. Le traitement de l'image :

Les images doivent être traitées dont le traitement sert à manipuler et analyser une image numérique.

Les tâches de ce traitement d'image sont :

- L'affichage des images
- Le recadrage
- Le retournement
- La rotation
- La classification
- Les extractions des caractéristiques
- La reconnaissance d'image. [11]

Le traitement de l'image se divise en deux parties :

1. Le prétraitement d'une image
2. L'analyse

On commence par :

I.8.1. Le prétraitement d'une image :

Le prétraitement est un ensemble de méthodes appliquées pour obtenir une image pour faciliter la recherche d'informations. Les images sont souvent déformées ou dégradées En raison d'un éclairage irrégulier ou d'interférences lors de l'acquisition

Il existe 3 grandes familles de procédés pour corriger une image :

➤ **L'amélioration :**

Cela comprend la correction des aspects visuels de l'image, c'est-à-dire Images contrastées. Les informations contenues dans le pixel ne changent pas. Parmi Méthode d'amélioration : expansion dynamique, compensation d'exposition, Égalisation de l'histogramme, amélioration du contraste, etc. [12]

➤ **Le dé-bruitage :**

Il comprend les informations contenues dans le pixel récupéré (valeur de luminance d'origine). Cette information est infectée par du bruit

(Variation aléatoire) lors de l'acquisition ou de la numérisation. Pour minimiser ce bruit, les filtres mentionnés suivants sont appliqués : filtre moyen, filtre gaussien, filtre médian, filtre à longue traîne, etc. [12]

➤ **Le flou :**

C'est un phénomène qui masque les détails et réduit caractéristiques des contours. Ce phénomène est causé par un mouvement uniforme ou non uniforme. Utilisez le filtre Netteté ou le filtre Le filtre passe-haut de l'image réduit l'effet de flou. [12]

I.8.2. L'analyse d'une image :

L'analyse d'image consiste à décomposer l'image prétraitée en -Le but est de trouver, détecter et identifier les objets contenus dans la scène observée afin d'avoir Une description du contenu sémantique de la scène. L'analyse d'images est un domaine très vaste qui a été connu et est encore connu, Évolutions importantes au cours des dernières décennies.

La segmentation est l'un des processus de base du traitement d'images numériques vision par ordinateur. Cela comprend la définition de régions présentant des similitudes internes, telles que les statistiques, la géométrie, la similarité des formes... l'image est segmentée en Les régions permettent à chaque région d'être traitée et analysée indépendamment Basé sur la politique ou le but du traitement. La zone couvre une plus grande surface plus petite que l'image globale mais généralement plus grande que les entités ponctuelles Local, afin que les

applications puissent utiliser des métriques régionales et locales plus petites Points d'intérêt (keypoints) comme signatures d'objets. [12]

La segmentation repose sur plusieurs méthodes. Parmi ces méthodes :

- Segmentation morphologique.
- Segmentation par couleur.
- Segmentation par seuillage.
- Segmentation par détection de contour.

La détection des bords est un outil courant dans le traitement d'images numériques. Il consiste à identifier les points de l'image numérique où la luminosité de l'image change. Exprime fortement ou plus formellement la discontinuité. Le point Les changements soudains de luminosité de l'image sont généralement organisés en groupe Segments courbes appelés profils. La détection des bords peut être facilement Reconnaitre de formes et reconnaissance d'objets. [12]

I.8.3. La localisation de la plaque d'immatriculation

La localisation sert à identifier la région de la plaque à partir de l'image prise par la caméra de surveillance.

La détection de la plaque est une partie très difficile à accomplir, en raison de la rapidité du véhicule.

Depuis des années, des recherches et des études ont été mises en œuvre pour trouver des méthodes vraiment utiles. On peut se référer dans ce cas à la méthode basée sur la détection de contour, méthode basée sur les opérations morphologiques, et même autre méthode basée sur la caractéristique de l'image. [13]

I.8.4. La détection de contour :

Identifie l'occurrence de changement entre les couleurs, cette détection est un ensemble de méthodes mathématiques qui sert à détecter l'occurrence de changement survenant dans l'image. La détection des contours est tellement efficace car elle se concentre sur le filtrage des informations nécessaires. [13]

Il existe plusieurs façons de détecter les contours d'une image. Ces méthodes sont Le filtre appliqué à l'image. Donnons quelques exemples :

Filtre Canny :

Le filtre Canny (ou détecteur Canny) est utilisé dans le traitement d'images pour la détection des contours. L'algorithme, conçu par John Canny en 1986, est optimisé selon les étapes suivant :

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



Figure I.6 : Application du filtre Canny

Et ceci est utilisé dans la reconnaissance optique des plaques d'immatriculations. Nous définissons d'ailleurs les bibliothèques nécessaires pour cette opération. [12]

Filtre dérivée première :

Ce filtre calcule les différences des pixels voisins sur les horizontales puis sur les verticales. [13]

Filtre de Prewitt :

Ce filtre applique le flou sur l'image, c'est un produit de dérivation par un filtre de flou rectangulaire. [13]

Filtre de Sobel :

La méthode de Prewitt est améliorée mais au lieu de du filtre rectangulaire on a un filtre triangulaire. [13]

Filtre de Deriche :

Un filtre varié du filtre Canny. [13]

Filtre dérivée secondes :

Un filtre qui se calculent en différences finies et c'est un changement qui concerne un point d'un contour, on l'utilise par la somme qui est le Laplacien. [13]

Filtre de Marr-hildreth:

Il est fait pour filtrages des hautes fréquences en calculant le Laplacien et même avant le lissage Gaussien avec 2 variances ajustées. [13]

I.8.5. Opération morphologique :

Cette méthode donne des opérateurs non-linéaires utilisés pour filtrer, segmenter et quantifier des images, d'abord aux images au niveau du gris, après au niveau des contours.

Les opérateurs morphologiques sont nombreux, on cite quelques-uns : Gardient morphologique, dilatation, ouverture et fermeture. [13]

I.8.6. Méthodes basées sur les caractéristiques de l'image :

Dans ce cas on se base sur les caractéristiques de l'image et cela en détectant la présence d'un objet ou son absence, calcul les caractéristiques de l'image.

La technique est d'utiliser les opérateurs de « Brique logicielle », et c'est la base de vision industrielle. [13]

Conclusion du chapitre I :

Dans ce chapitre nous avons présenté une vue générale sur la reconnaissance optique de caractères et de l'image, plus précisément l'image numérique. Ou on a abordé les caractéristiques de ce dernier suivi par ses techniques de traitement.

Dans le chapitre suivant nous allons voir le hardware & le software utilisés dans la conception de notre système

Chapitre II :
Descriptions matérielle et logicielle
du système

II.1. Introduction :

Le but de notre projet de fin d'étude est de réaliser un système embarqué de Reconnaissance Automatique de Plaques d'Immatriculation à l'aide d'une caméra pi et d'un Raspberry Pi, et on a choisi comme langage de programmation le Python car il est intégré dans le système d'exploitation LINUX utilisé sur Raspberry Pi

II.2. Le Raspberry PI :

Raspberry : De nombreux fabricants d'ordinateurs tirent leur nom du fruit, et la framboise ne fait pas exception. Avant le Raspberry Pi, nous avons des ordinateurs Apple, Acons et Abricot. Ainsi, le nom Raspberry suit cette tendance, mais c'est aussi une référence humoristique à l'expression "coup de framboise".

Pi : "Pi" fait référence à Python, qui a été utilisé pour créer le premier design. Ces premières machines démarrent sur un terminal où il faut entrer du code Python pour obtenir ce que l'on veut.

C'est pour le distinguer des autres ordinateurs qui utilisent BASIC. [14]

II.2.1. Description du Raspberry pi :

Le Raspberry est un mini PC mono carte ou un ordinateur réduit, lorsqu'on branche la souris, le clavier, le câble HDMI (pour l'écran) et l'alimentation, il fonctionne comme un ordinateur ordinaire ou un PC. C'est possible de l'utiliser pour le besoin personnel mais lent par rapport aux autres ordinateurs.

Le Raspberry est utilisé pour le traitement des images en temps réel, il est possible, il peut apporter des fonctionnalités ou des capacités attendus et consomme même une faible énergie (économique).

La fonction Raspberry donne officiellement un système d'exploitation Raspbian basé sur Debian, qui est disponible gratuitement.

Pour stocker le système d'exploitation nous utilisons une carte SD classe 10.

Le Raspberry est un processeur qui permet de brancher du matériel sur sa puce grâce au GPIO comme un moteur pas à pas ou un moteur à CC, une LED, un capteur, pour les contrôler.

La vitesse de cet ordinateur varie entre 700MHz à 1.2GHz, et un SDRAM de 256Mo à 1Go

Il y en a plusieurs versions du Raspberry et sont : [14]

1. Raspberry Pi 1 Modèle A
2. Raspberry Pi 1 Modèle A+

3. Raspberry Pi 1 Modèle B
4. Raspberry Pi 1 Modèle B+
5. Raspberry Pi 2 Modèle B
6. Raspberry Pi 3 Modèle B
7. Raspberry Pi Zéro [14]

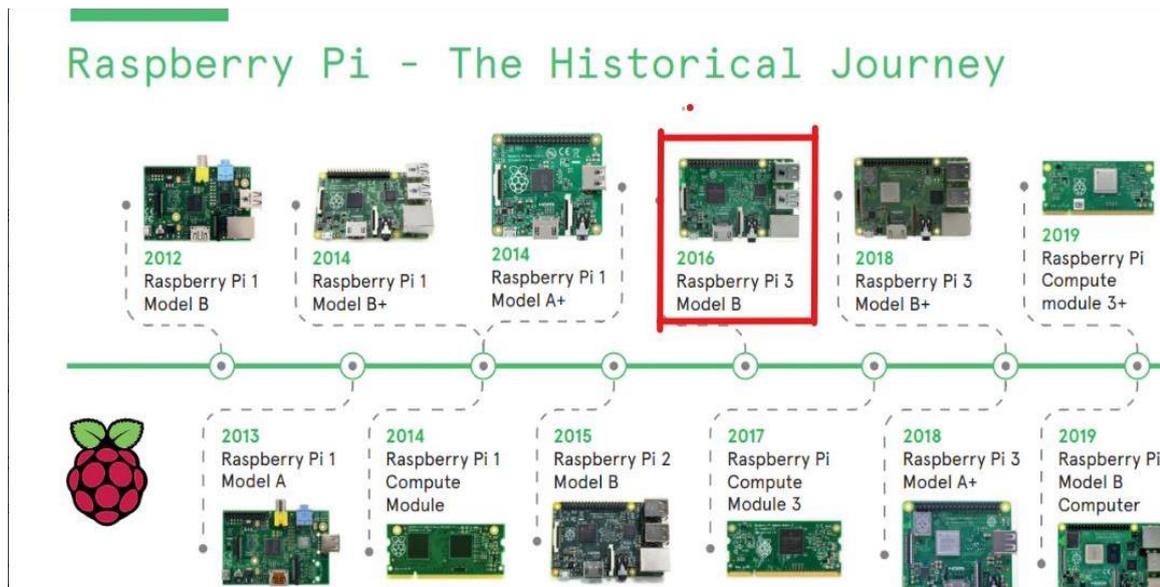


Figure II.1: Classement des différents types du Raspberry

Nous utilisons pour notre projet le Raspberry Pi 3 Modèle B. [15]

II.2.2. Historique :

L'arrivée et la naissance du Raspberry est récente, le produit est sorti et lancé le 29 février 2012.

Le Raspberry est un ordinateur qui fait la taille d'une carte bancaire (85mm*56,2mm), [] une invention magique et à mini prix. L'idée de sa création a commencé en 2006 simultanément avec la création des premiers prototypes inspirés par BBC micro par un ingénieur Britannique Eben Upton qui a étudié physique à l'université de Cambridge et a travaillé à l'entreprise Broadcom, Intel et IBM.

Au début, il testait son invention sur une grande plaque électronique mais son but était clair, un mini PC et à un prix raisonnable pour les étudiants, il voulait les aider pouvoir posséder un ordinateur. [15]

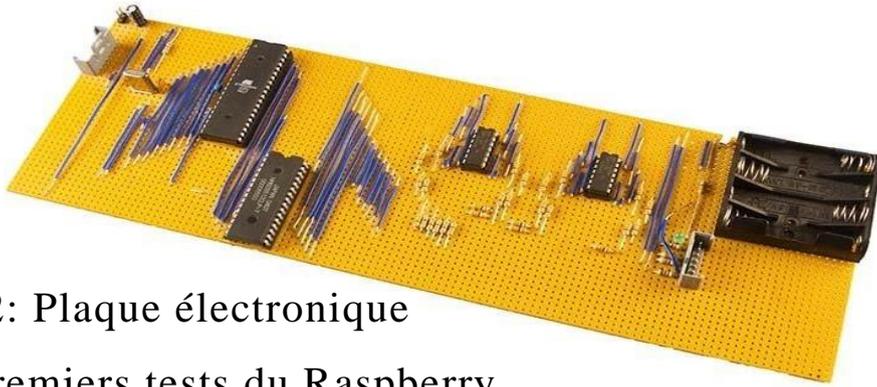


Figure II.2: Plaque électronique pour les premiers tests du Raspberry

Aujourd'hui plus de 15 millions de cet ordinateur se vend au monde.

II.2.3. Raspberry PI 3 Modèle B :

Le Raspberry Pi 3 modèle B fonctionne bien. Les cartes Raspberry Pi 3 Model B sont désormais nativement équipées du Wifi b/g/n d'une interface de 2,4GHz et 5GHz et du Bluetooth 4.1. Il porte une carte mère idéale pour un système d'architecture ARM.

Ce processeur est Broadcom BCM2837e il porte un GPU Dual-Core VideoCore capable de décoder le flux de vidéo HD 1080p. Le stockage de sa RAM est 1Go. [16]

Le Raspberry PI3 est une carte intégrée, on trouve plusieurs composants.

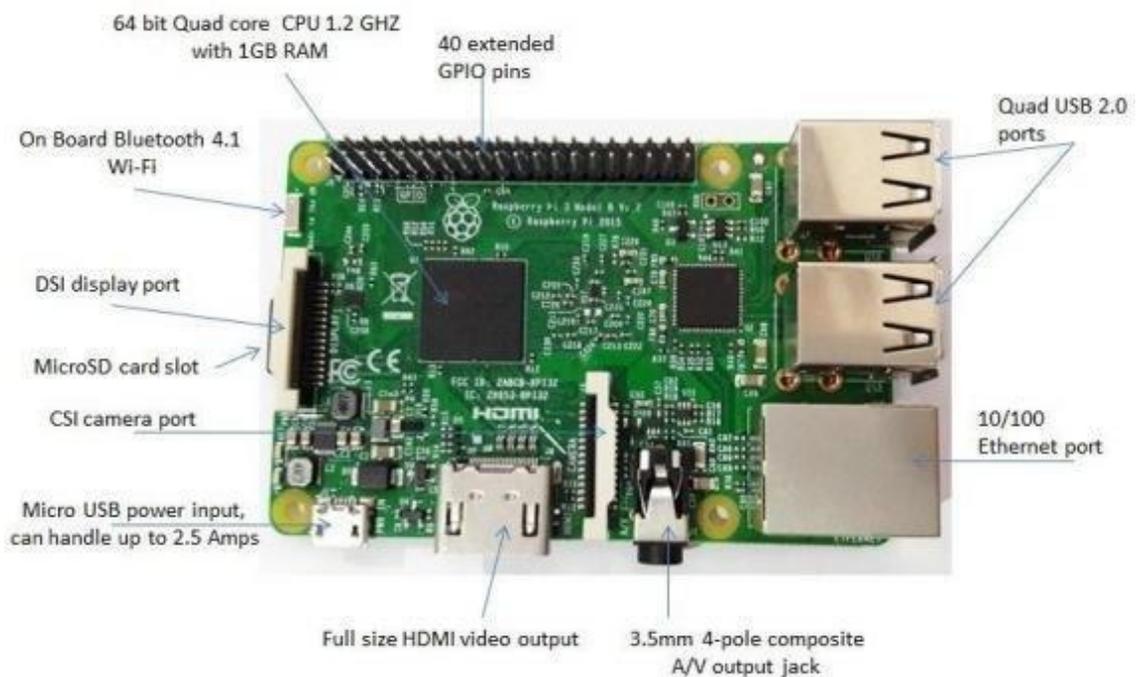


Figure II.3: Anatomie du Raspberry Pi 3B

II.2.3.1. Les composants du Raspberry Pi3 :

GPIO :

Les ports GPIO (General Purpose Input/output ou plutôt Entrée/Sortie à usage général) sont des ports d'entrée et de sortie largement utilisés dans le monde des microcontrôleurs, et justement dans le domaine de l'électronique depuis les années 80. Ces ports sont intégrés sur les circuits électroniques pour transmettre les informations avec des composants électroniques et circuits externes. Alors qu'il peut même fonctionner comme détecteurs ou senseurs ou contrôler des commandes.

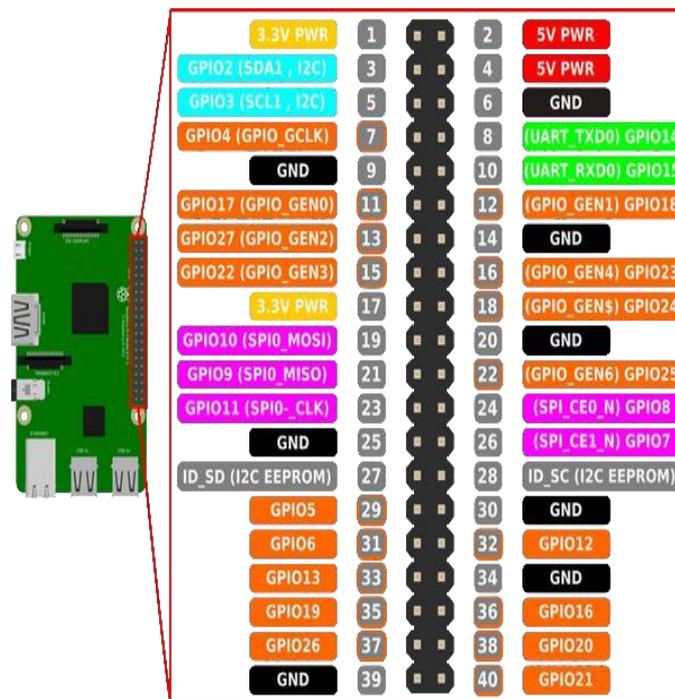


Figure II.4: GPIO d'un Raspberry Pi B 3

Connecteurs :

Le connecteur à 40 broches est moyen offert par les fondateurs du Raspberry PI pour accéder aux ports des entrées/sorties du Broadcom BCM2837.

Chaque broche de GPIO à un rôle déterminé.

Nous avons les broches d'alimentation de 5v et 3,3v et les masses. Ainsi les autres ports qui sont identifiés par des chiffres : GPIO1, GPIO2,... ; quelques ports ont des fonctions supplémentaires mais cela ne veut pas dire qu'on ne pourra pas leurs utiliser de façon classique comme entrée/sortie (0/1)

Il n'y en a pas de port analogique avec une tension continue variable, et si l'utilisateur a besoin de plus de ports analogiques, il suffit d'ajouter une carte d'extensions. [17]

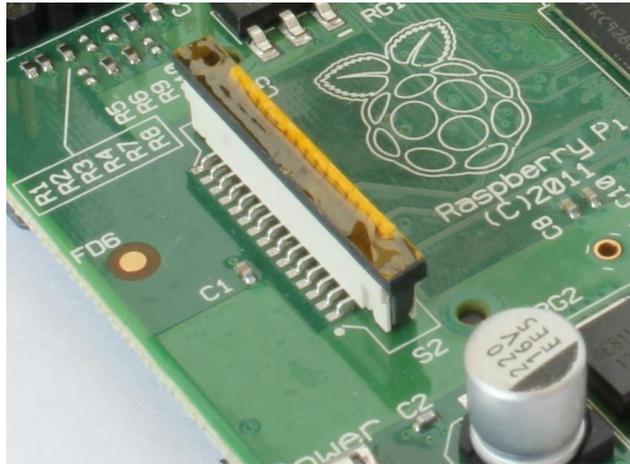


Figure II.6: Port d'écran DSI

Port HDMI :

Le Raspberry pi conditionné d'un port HDMI identitaires à celui de la télévision, l'écran de l'ordinateur et moniteurs modernes, ce qui permet de brancher le processeur avec ces écrans cités qui disposent d'un port VGA. Donc il faut avoir un câble et adaptateur HDMI VGA.[20]

Micro Card slot :

Le Raspberry a besoin d'une carte mémoire classe 10 d'au moins 8Gb pour bien fonctionner et pour pouvoir stocker les fichiers et le système d'exploitation. Il existe aussi des cartes mémoires avec le Raspbian (le système) installé.[20]

Audio et vidéo composite :

Le Raspberry dispose d'un port audio standard comme celle des téléphones et lecteur MP3, pour écouter le son de l'ordinateur il faut brancher les écouteurs, baffles, ou haut-parleurs, ou encore mieux si l'écran est connecté à un Raspberry avec un haut-parleur intégré pour écouter le son. [20]

Broadcom BCM2837 :

Est une puce intégrée et plantée sur le Raspberry PI3 Modèle B. l'architecture de BCM2837 est compatible à celle de BCM2836utilisé sur Raspberry PI2B, la différence est le cluster Quadri cœur ARMv7 par le cluster Quadri cœur ARM Cortex A53 (ARMv8).

Il fonctionne à 1,2GHz et cela rend le processeur 50% plus rapide que celle du Raspberry PI2. La VideoCore 6 fonctionne à 400MHz [21]

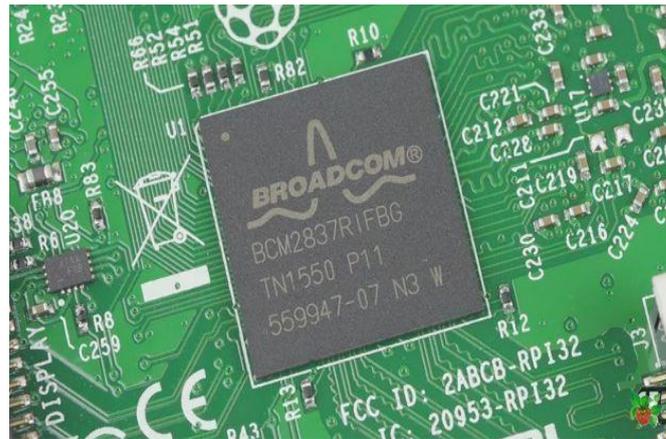


Figure II.7: Broadcom BCM2837

Circuit d'alimentation :

Détection d'alimentation incorrecte lors d'une alimentation faible ou un court-circuit le circuit détecte l'erreur.

L'alimentation électrique :

A l'aide d'un câble qui est généralement identique à celle des téléphones portables (USB-C). L'alimentation doit fournir au loin 2,5 ampères et une tension de 5v pour éviter tous les problèmes d'alimentation, car si l'alimentation n'est pas parfaite le risque de perdre le Raspberry est trop élevé. [20]

Port RJ45 :

Il permet au Raspberry de se connecter à un routeur sans fil, ADSL ou même autre appareils qui partage une connectivité.

Les voyants lumineux vert et rouge :

La lumière verte s'allume pour indiquer le statut de la carte SD, il clignote lorsqu'un programme est en exécution.

La lumière rouge s'indique pour indiquer si le Raspberry reçoit une alimentation suffisante ou pas, si la lumière rouge clignote cela veut dire que l'alimentation ne fournit pas assez de courant, et cela arrive si la tension de l'alimentation est inférieure de 4,6v. [22]

Le système d'exploitation utilisé dans ce projet est le Raspbian qui est un système d'exploitation libre et gratuit basé sur Debian optimisé pour fonctionner sur les différents Raspberry Pi

II.2.4. Les étapes d'installation et de configuration du système d'exploitation :

Pour installer Raspbian sur la carte SD il faut suivre les étapes suivantes.

Étape 1 : Installation de SDFormatter V4.0 pour formater la carte SD

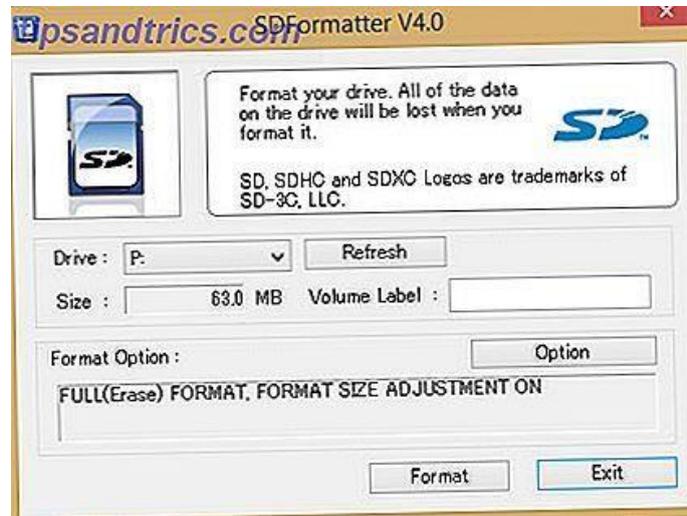


Figure II.8: SDFormatter V4.0

Étape 2 : télécharger Imager Raspberry Pi sur l'ordinateur (Windows, Mac et Ubuntu).



Figure II.9: Raspberry Pi imager

Étape 3 : brancher la carte SD sur l'ordinateur et lancer Raspberry Imager et installer une image personnalisée (système d'exploitation Raspbian).

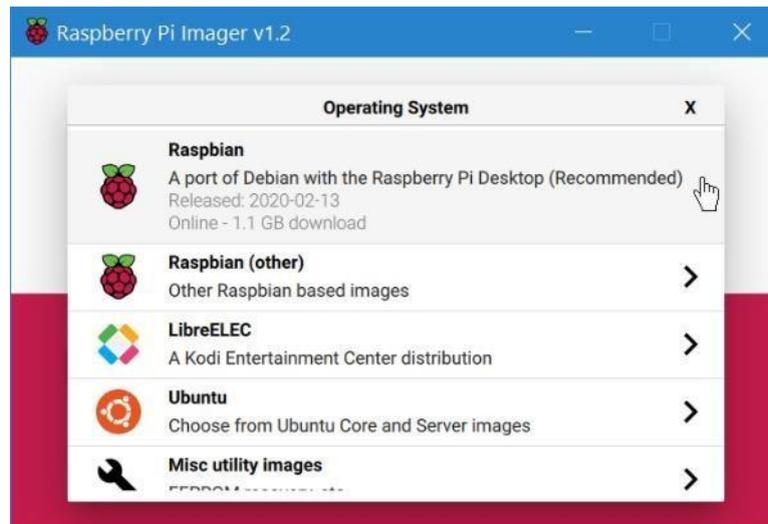


Figure II.10: L'installation du Raspbian

Étape 4 : lorsque l'installation est terminée retirer la carte SD de l'ordinateur et brancher la sur le Raspberry et on aura l'accès au bureau directement pour faire la configuration. [29]



Figure II.11: La configuration du Raspberry

II.2.5. Connexion SSH :

Le SSH ou Secure Shell est un protocole de communication et programme informatique. Il sert à faire la connexion d'un serveur en toute sécurité pour transmettre des fichiers ou des commandes, et même manipuler via le terminal depuis un autre ordinateur. Ce qui est vraiment pratique avec le Raspberry Pi, elle est très utilisée pour les projets. [30]

Pour réaliser cette connexion il faut installer PUTTY sur un ordinateur ensuite activer la connexion SSH sur le Raspberry, cela en ouvrant le menu de configuration Raspberry Pi par la commande : **sudo raspi-config** ensuite sélectionner « 5- Interfacing Options »

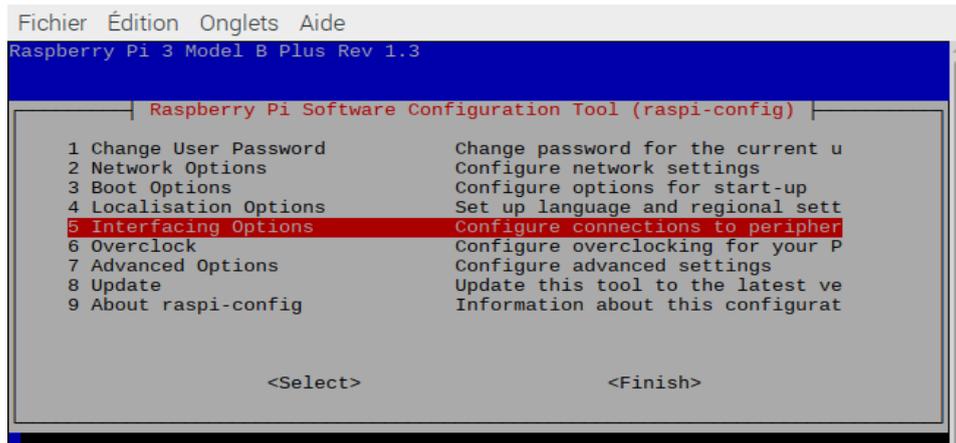


Figure II.12: La configuration des connexions

Puis activer « P2 SSH » :

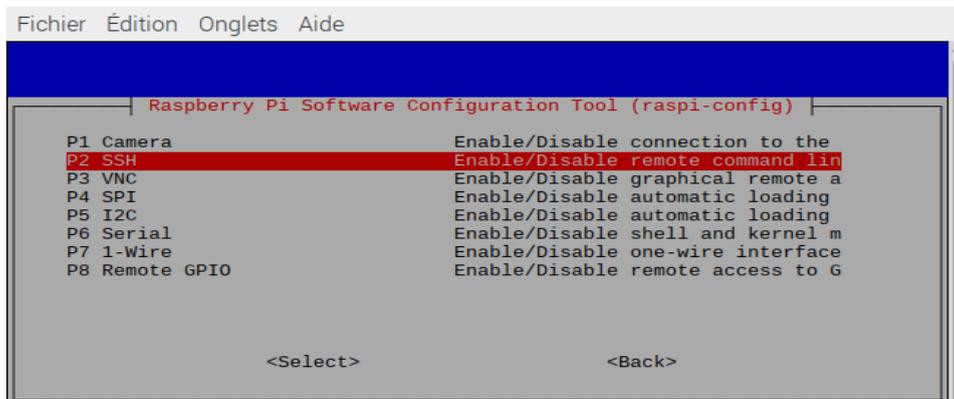


Figure II.13 : L'activation du SSH

Après sur l'autre ordinateur il faut ajouter l'adresse IP du Raspberry dans l'application PUTTY :

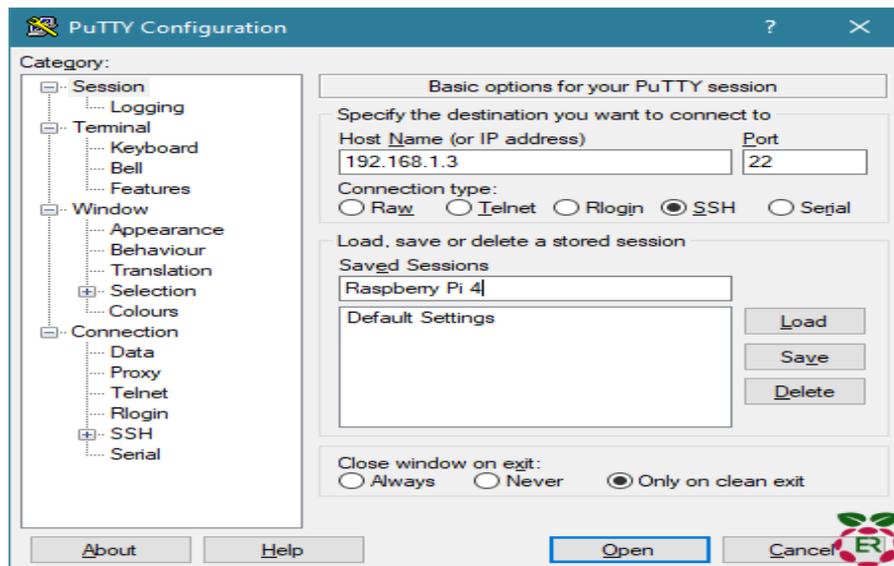


Figure II.14: Putty

Et pour obtenir l'adresse IP il faut entrer la commande **ifconfig**

Désormais l'ordinateur peut avoir accès au terminal du Raspberry

II.2.6. Connexion VNC :

Virtual Network Computing, ce système permet de prendre le contrôle à distance du Raspberry ou même autre ordinateur.

Pour utiliser cette connexion il faut passer par plusieurs étapes :

Entrer la commande **sudo raspi-config**, puis sélectionné « 5- interfacing Options », après activer le « P3 VNC » sur le Raspberry.

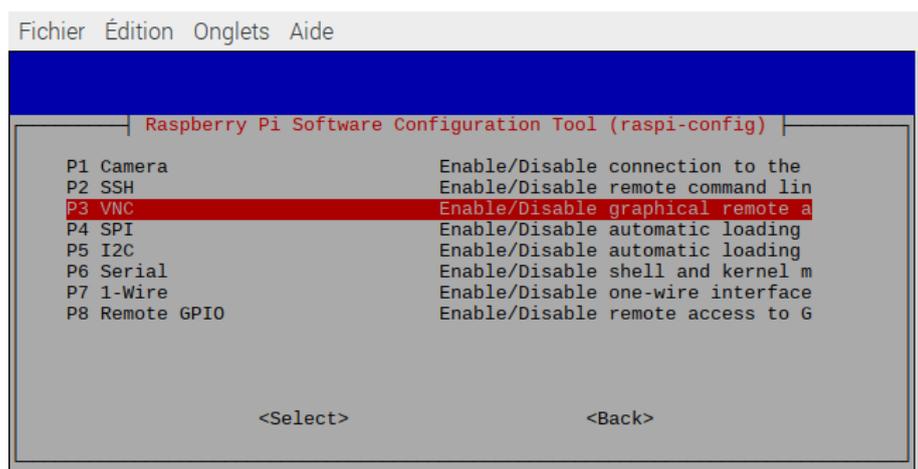


Figure II.15: L'activation de VNC

Puis installer le VNC sur l'ordinateur et entrer l'adresse IP :

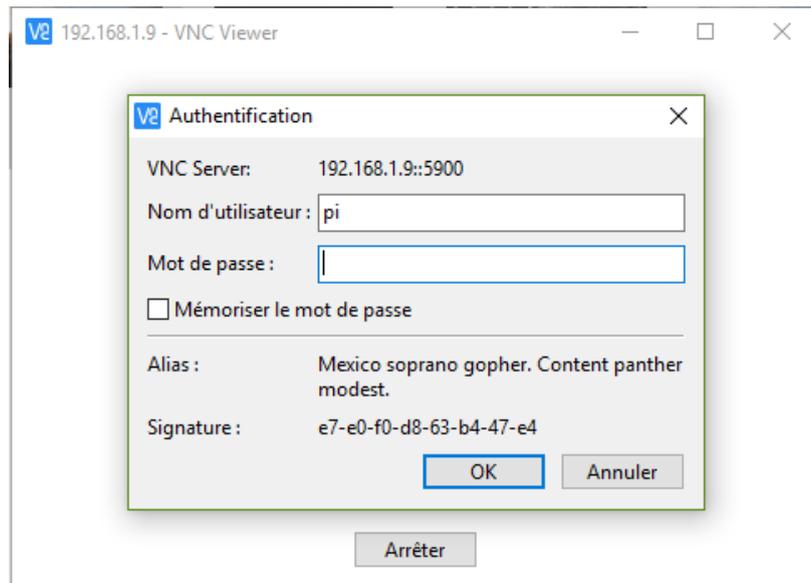


Figure II.16: VNC

Et le signe VNC apparaît sur le Raspberry et se connecte avec l'autre ordinateur. [31]

Pour la conception de notre système nous avons choisis le langage de programmation python

II.3. Python :

Python est le langage principal du Raspberry PI, il est préinstallé sur le système Rasbian, ainsi qu'il est simple pour la programmation surtout pour des débutants

Les points forts de Python sont :

- Langage simple et facile pour programmer aux débutants
- Il est adapté aux petits projets qu'aux gros projets.
- Peut-être cross-Platform
- Stable, c'est un langage ancien
- Simple à comprendre [23]

II.3.1. Pourquoi on a choisi python :

Python est un langage de programmation relativement facile à apprendre qui suit une structure organisée et une syntaxe simple, et permet une variété d'utilisations. Cela en fait un excellent choix pour un large éventail de projets, des simples applications Web aux systèmes d'exploitation complets. En plus python présente des bibliothèques et des modules intégrés avancés qui permettent la résolution d'une bonne partie de notre travail

II.3.2. Les bibliothèques python :

Numpy : Est une bibliothèque d'open source principale dans la programmation python, maîtrise plusieurs fonctions pour la manipulation des tableaux, vecteurs, matrices... [5]

PIL/PILLOW : Python Imaging Library est une bibliothèque open source aussi en python, qui se charge de l'ouverture et l'enregistrement de multiples fichiers d'images.

Cette bibliothèque n'est plus entretenue depuis 2009 et a été remplacée par PILLOW à partir de 2010. [5]

Opencv-Python :

Open Computer Vision développé par Intel et sa première version est sorti en juin 2000, Opencv est l'une des bibliothèques les plus utilisées, la plus utile et même rapide, car son arrière-plan est constitué de code C/C++, elle est l'API python pour Opencv. [5]

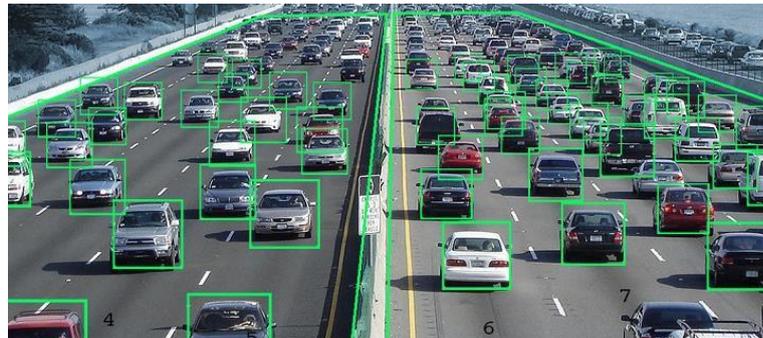


Figure II.17: Exemple d'une fonction de la bibliothèque OpenCV

Opencv est et notre choix pour programmer notre système en utilisant les modules suivants :

Datetime : Le module Datetime fournit des classes pour manipuler les dates et les heures. Bien que les calculs de date et d'heure soient pris en charge, l'accent est mis lors de l'implémentation sur l'extraction efficace des propriétés pour la manipulation et le formatage pour l'affichage.

Locale : Le module Locale ouvre l'accès à la base de données et aux fonctions locales de POSIX. Le mécanisme de paramètres régionaux POSIX permet aux programmeurs de traiter certains problèmes culturels dans une application sans exiger que le programmeur connaisse tous les détails de chaque pays dans lequel le logiciel s'exécute.

Pytesseract : Python-tesseract est un outil de reconnaissance optique de caractères (OCR) pour Python. En d'autres termes, il reconnaîtra et "lira" le texte intégré à l'image.

Python-tesseract est un Wrapper pour le moteur Tesseract-OCR de Google. Il peut également

être utilisé comme script d'appel autonome pour tesseract, car il peut lire tous les types d'images pris en charge par les bibliothèques d'images Pillow et Leptonica, y compris jpeg, png, gif, bmp, tiff, etc. De plus, s'il est utilisé comme script, Python-tesseract imprimera le texte reconnu au lieu de l'écrire dans un fichier.

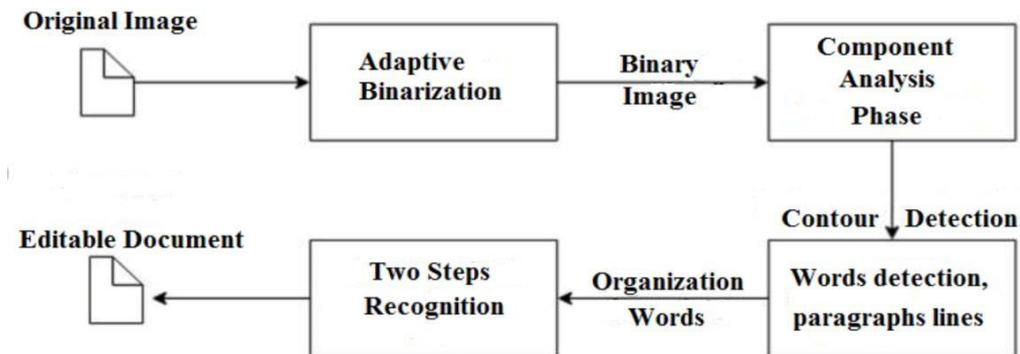


Figure II.18: Principe du module **Pytesseract**

II.4. Mini servomoteur 9G SG90:

Ce mini servomoteur est un moteur de haute qualité et de faible coût que nous avons utilisé dans le prototype de notre système pour la barrière qui permet l'accès aux automobiles. Il a un câble d'alimentation et contrôlé par 3 broches.

La tension de son alimentation est 4,8v-6v.



Figure II.19: Mini servomoteur 9G SG90

Conclusion du chapitre II :

Dans ce chapitre, nous avons décrit le matériel et le logiciel utilisé dans notre système, et nous avons présenté le Raspberry et ses caractéristiques ainsi que ses composants et les bibliothèques et les modules que nous avons utilisés sous Python.

Dans le prochain chapitre nous allons voir les étapes, le programme et la mise en œuvre de notre projet.

Chapitre III :
Mise en œuvre du système

III.1. Introduction :

Après avoir défini et décrit les différents modules qui composent notre système, nous allons décrire donc en détaille les différentes étapes de la réalisation pratique.

Les systèmes RAPI (Reconnaissance Automatique des Plaques d'Immatriculation) sert à automatiser la lecture et l'identification des plaques d'immatriculation à travers des techniques de traitement d'image ainsi que les techniques OCR (Optical Character Recognition, en anglais).

Nous allons nous concentrer sur les systèmes RAPI pour l'accès au parking car c'est notre cas d'étude.

Ces systèmes sont conçus pour la surveillance de flux de véhicules à travers des images ou des vidéos pour détecter et lire les plaques d'immatriculation des véhicules. Ils peuvent aussi être utilisés pour stocker les images capturées par les caméras : l'image du véhicule ainsi qu'une photo du conducteur.



Figure III.1: Système RAPI

III.2. Méthodes de détection des plaques d'immatriculation :

Afin de détecter, repérer et extraire les plaques d'immatriculation de l'image, les systèmes de Reconnaissance Automatique des Plaques d'Immatriculation utilisent en général des séries de techniques et méthodes de traitement d'image comme :

III.2.2. Technique basée sur les propriétés de la plaque

Cette technique utilise les propriétés suivantes :

- La couleur de la plaque.
- Segmentation d'image noir et blanc après l'utilisation des seuils.
- La taille des caractères et la distance qui existe entre eux.

Les algorithmes basés sur les couleurs ne sont pas efficaces car les couleurs changent en fonction d'éclairage, donc la détection d'autres régions non correspondantes à la plaque.



Détection de contour :



Matricule :



Résultat :

Detected license plate Number is: 00004 120 19}

Figure III.2: Technique basée sur les propriétés de la plaque

III.2.2. Technique basée sur l'intelligence artificielle :

Cette technique utilise :

- Un réseau de neurones à deux entrées comme un filtre horizontal et vertical pour détecter les plaques, de telle façon l'intersection entre les deux filtres localise la région de la plaque.
- Un réseau de neurones pour déterminer les régions susceptibles qu'on va analyser par la suite par une transformé de fourrier pour détecter la bonne région, l'algorithme sera répété jusqu'à la détermination de la plaque. Mais la réussite de cet algorithme dépend de l'éclairage

III.2.3. Technique basée sur la signature de la plaque :

Cette technique utilise un des algorithmes suivants :

- L'algorithme d'Adaboost.

- Un algorithme qui choisit une région du texte à partir d'un ensemble puis il exploite de telle façon une plaque contient des caractères et/ou des symboles qui sont clairement visibles. Le problème est que certaines autres régions peuvent contenir du texte.

III.2.4. Technique basée sur les contours de la plaque :

C'est cette technique qui est utilisé pour notre système. Cette technique utilise les contours des caractères et des plaques comme des points de référence pour l'extraction. L'intensité des pixels dans les contours des caractères et des plaques est complètement différente que celle des voisins.

Le point faible de cette technique est lorsque les bords de la plaque ne présentent pas une grande variation d'intensité par rapport au reste de l'image.



Figure III.3: Technique basée sur les contours de la plaque

III.2.5. L'emplacement physique de notre système :

Vu le nombre important de véhicules qui accèdent chaque jour aux différents parkings, nous avons décidé d'implanter notre système RAPI à l'entrée du parking de tel façon à avoir le matricule du véhicule en face d'une webcam implémentée dans notre système

III.3. Matériels utilisés :

Donc comme nous avons expliqué dans le chapitre 2, nous avons choisi d'utiliser un Raspberry Pi 3 modèle B+ par rapport à Son prédécesseur, le Raspberry Pi 3 B+, est un boost complet de vitesse et de puissance 15 %. Ses caractéristiques uniques sont :

- 1) Accélération du processeur quadricœur : environ 15 % de puissance de traitement Additionnel

- 2) Prend en charge le décodage H.264 MPEG-4 (1080p30), l'encodage H.264 (1080p30) et Ouvrez le gestionnaire de graphiques GL ES 1.1 et 2.0
- 3) Communication sans fil double bande 802.11ac (2,4 GHz/5,0 GHz)
- 4) Meilleure gestion de l'alimentation,
- 5) Amélioration de la gestion thermique.

Et pour la capture des vidéos on a utilisé une webcam, les caractéristiques de cette caméra :

- 500K pixels de résolutions maximum
- Véritable interface USB facile à brancher et à utiliser
- Capture CMOS de haute qualité
- Capture d'image fixes et de vidéos claires et nettes
- Idéalement conçu pour fonctionner avec Raspberry
- Objectif réglable pour une prise de vue précise
- Comprend une variété de contrôles d'image
- Balance des blancs et exposition automatiques



Figure III.4: Webcam utilisée

Pour la réalisation d'un prototype expérimentale nous avons utilisé un servomoteur 9G SG90

Pour notre Raspberry, nous avons installé le système sur une carte mémoire SD base 10 Système d'exploitation Raspbian (il s'agit d'un système d'exploitation gratuit et open source basée sur Debian optimisation) pour faire fonctionner le Raspberry Pi.

III.3.1. Branchement du Raspberry PI 3 :

D'abord il faut avoir le matériel nécessaire, le clavier, la souris, l'écran. Donc le Raspberry PI3 a 4 ports USB, nous branchons les périphériques USB (le clavier et la souris), ainsi que l'écran avec le port HDMI à l'aide du câble VGA, et vu que pour le projet nous aurons besoin d'une caméra pour la reconnaissance, nous utiliserons une webcam USB avec un port USB du Raspberry.

Finalement on branche l'alimentation et quelques secondes après il s'allume et nous accédons directement au bureau.

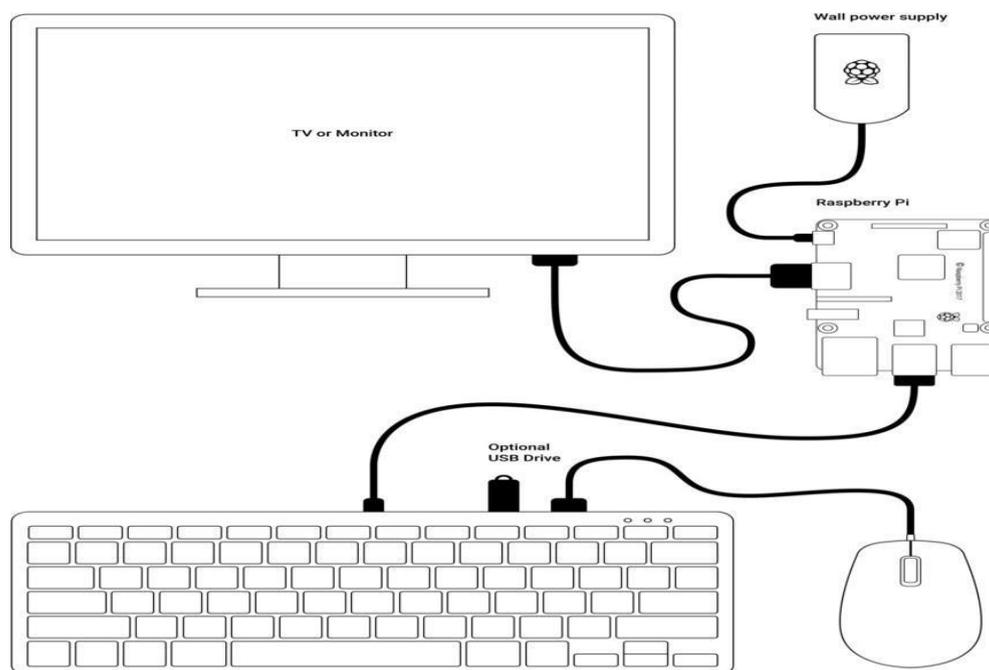


Figure III.5: Branchement du Raspberry

III.4. Composition du programme python

III.4.1. Les bibliothèques importées :

Les bibliothèques utilisées pour notre système sont :

Import time : c'est un module pour gérer les tâches liées au temps. [35]

Import locale : ce module donne accès à la base de données et aux fonctionnalités des paramètres linguistiques définis. [36]

From datetime import datetime : Le module fournit des classes pour manipuler les dates et les heures. [37]

From datetime import strftime : convertit la date et l'heure en format de chaîne.

Import cv2 : importation de la bibliothèque Open-cv.

Import imutils : facilite les fonctions de traitement d'image.

Import numpy as np : une bibliothèque numérique

Import pytesseract : pour la reconnaissance optique des caractères.

From PIL import image : une bibliothèque qui manipule les images.

Import csv : on utilise cette bibliothèque qui vient avec Python pour lire et écrire dans des fichiers CSV [38].

Import RPi.GPIO as GPIO : pour le branchement de Raspberry avec le moteur

III.4.2. Préparation du cadre et identification de la plaque :

Notre programme commence par une instruction qui permet de mettre l'heure et le nom de la ville (Tlemcen) dans l'encadrement :

```
cv2.putText(image, "-" + strftime("%H/%M/%S"), (350,20),font, 0.5,
(255,255,0),1)
```

```
cv2.putText(image, "Tlemcen", (5,20),font, 0.5, (255,255,0),1)
```

Puis une instruction pour mettre aussi la date du jour dans l'encadrement :

```
cv2.putText(image, "-" + strftime("%d/%m/%y"), (100,20),font, 0.5,
(255,255,0),1)
```

Puis l'instruction qui permet de lancer une diffusion en directe à partir de la webcam

```
font= cv2.VideoCapture(0)
```

Nous avons déclaré 2 paramètres **ret** et **image**, **ret** est une variable booléenne qui renvoie vrai si le cadre est disponible et **frame** est un vecteur de tableau d'images capturé sur la base des images par seconde par défaut définies explicitement ou implicitement :

```
Ret, image= cap.read()
```

Pour changer la taille du cadre d'image on a utilisé :

```
Image = cv2.resize(image, (600,400))
```

Pour afficher la vidéo sur l'écran :

```
cv2.imshow("frame", image)
```

Pour le contrôle d'accès ; les véhicules doivent être positionnés en face de la webcam, une fois on a la bonne position, le chauffeur peut utiliser un premier bouton proche de sa gauche pour la capture puis la détection de la plaque d'immatriculation.

```
if cv2.waitKey(25) & 0xFF == ord('q')
```

2- Ensuite, il faut passer par une étape de conversion l'image en échelle de gris. Pour le faire, nous devons appeler la fonction `cvtColor`, qui permet de convertir l'image d'un espace couleur à un autre.

Comme nous voulons convertir notre image originale de l'espace couleur BGR en gris, nous utilisons l'instruction

```
gray= cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) [39]
```

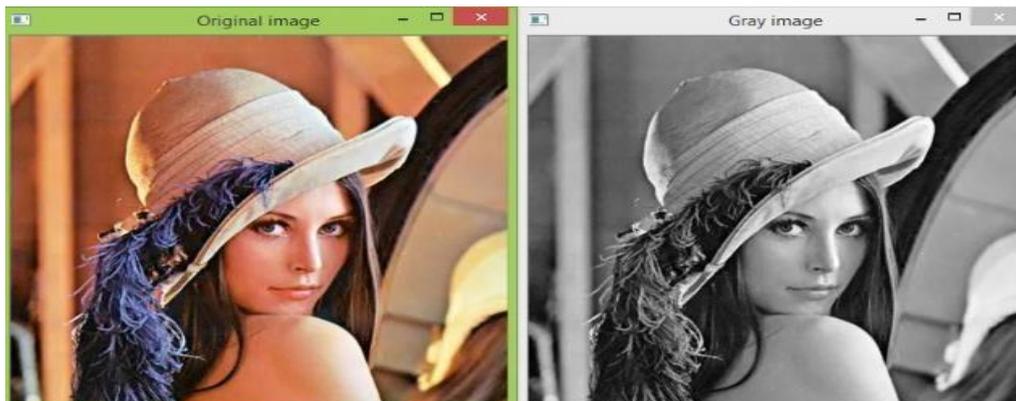


Figure III.6: Conversion l'image en échelle de gris [35]

3. Ensuite, en utilisant la fonction de `gray= cv2.bilateralFilter(gray, 11 , 17, 17)` d'OpenCV, nous réduisons le bruit dans l'image pour une meilleure détection des contours. Puis on passe à l'application du filtre canny sur la l'image en gris pour détecte les contours de l'image

```
Edged= cv2.Canny(gray, 30, 200)
```



Figure III.7: Réduction du bruit et application du filtre Canny [40]

4. Les contours sont des courbes qui aident à identifier les zones d'une image qui ont la même intensité. Les contours ici nous aideront à identifier la plaque d'immatriculation de la voiture à partir de l'image. La fonction accepte trois paramètres :

```
Cnts=cv2.findContours(edged.copy(),  
cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
```

Le premier paramètre est l'image source. Ici, nous faisons une copie de l'image avec des bordures, car cette fonction recherche à plusieurs reprises les bords de l'image, ce qui signifie qu'elle rend l'image inutilisable pour une utilisation future. De plus, les images bordées facilitent l'identification de courbes d'intensité similaires.

Le deuxième paramètre est le mode de récupération des bords. Ici, le type **RETR_LIST** est utilisé pour récupérer tous les bords détectés dans l'image.

Le troisième paramètre est la méthode d'approximation des bords.

CHAIN_APPROX_SIMPLE stocke les extrémités du contour détecté. [41]

5. L'instruction **Cnts=imutils.grab_contours(cnts)**, retournant les compteurs (contours) en cnts, sans distinguer entre opencv2 ou opencv3.[42]

6. **Sorted(cnts key=cv2.contourarea reverse=True)[:10]**

Ici, nous sélectionnons les 10 premiers contours par ordre croissant de surface, car nous n'avons pas besoin de différents petits contours. Cela réduit les contours redondants et les petits contours indésirables [41]

7. Pour éviter la prise des captures flou et on risque de ne pas pouvoir détecter la plaque d'immatriculation

```
screenCnt= None
```

III.4.3. Résultat de la détection :

For c in cnts

```
peri= cv2.arclength(c,True)
```

```
approx= cv2.approxPolyDP(c,0.018*peri,True)
```

1) En utilisant la fonction peri, nous calculons le périmètre de chaque contour.

```
if len (approx)==4:
```

```
    screenCnt=approx
```

On a mis cette condition pour choisir un contour qui forme un rectangle ou un carré ça veut dire qu'il contient 04 angles et on le capture

```
cv2.drawContours(img, [screenCnt], -1, (0, 255, 0), 3)
```

Le premier argument est l'image dans laquelle les contours détectés seront dessinés. Le deuxième argument est la variable qui stocke tous les contours détectés. Ensuite, le troisième argument est l'index des contours. Ici, nous utilisons la valeur -1 qui utilisera les index de tous les contours détectés dans notre image. Ainsi, tous les contours seront dessinés sur l'image. Ensuite, le quatrième argument est la couleur dans laquelle les contours seront dessinés.

Le cinquième argument est l'épaisseur de la courbe de contour à dessiner [41]

Nous avons ajouté les instructions suivantes pour masquer toute autre partie que la plaque d'immatriculation :

```
mask = np.zeros(gray.shape, np.uint8)
```

```
new_image = cv2.drawContours(mask, [screenCnt], 0, 255, -1,)
```

```
new_image = cv2.bitwise_and(image, image, mask=mask)
```

Pour recadrer une image aux limites d'un contour on utilise :

```
(x, y) = np.where(mask == 255)
```

```
(topx, topy) = (np.min(x)-5, np.min(y)-5)
```

```
(bottomx, bottomy) = (np.max(x), np.max(y))
```

```
Cropped = gray[topx:bottomx + 5, topy:bottomy + 5]
```

Pour lire la plaque d'immatriculation :

```
text = pytesseract.image_to_string(Cropped, config='--psm 11')
```

```
print("Detected license plate Number is:", text)
```

Cette instruction est pour visualiser l'image :

```
cv2.imshow('frame', image)
```

Pour visualiser l'encadrement ou la capture de la plaque toute seule :

```
cv2.imshow('Cropped', Cropped)
```

Pour supprimer les caractères (espaces aux débuts et à la fin) :

```
matricule = text.strip()
```

III.4.4. Programmation des Bases de données :

Dans notre projet, nous avons 2 bases de données CSV : **base1.csv** et **base2.csv**

III.4.4.1. La base de données « CSV » :

En anglais c'est Comma Separated Values, c'est un fichier qui stocke les données et d'ailleurs il est appelé une base de données. Les champs sont séparés par une virgule dans ce fichier.

Le CSV est utilisé en plusieurs applications, dans notre projet nous nous sommes servis de ce fichier pour stocker les matricules reconnues par notre système ainsi que les matricules non-reconnues, et bien sûr dans des fichiers CSV différents. [33]

Et cela va aider à ouvrir ou fermer la barrière.

III.4.4.2. L'importance des bases de données :

La détection et la lecture des plaques d'immatriculation renvoient un texte représentant le numéro d'immatriculation, ce numéro est comparé avec les autres enregistrements de la base de données base1.csv (les matricules des véhicules qui ont l'autorisation d'accès)

Après la comparaison on a deux cas possibles :

1. Le numéro d'immatriculation n'existe pas dans la base de données base1.csv, donc le véhicule n'est pas autorisé à entrer et la barrière ne s'ouvre pas.
2. Le numéro d'immatriculation existe dans la base1.csv (les matricule des véhicules qui ont l'autorisation d'accès) donc on fait l'enregistrement dans la base2.csv et on ouvre la barrière avec la partie de programmation qui suit.

La base2.csv contient la liste des matricules autorisés avec la date et l'heure et la capture du matricule en photo.

III.4.4.3. Lecture et écriture :

Cette commande est pour ouvrir le fichier csv :

```
With open('base1.csv') as file_csv :
```

Pour la lecture du fichier on utilise :

```
Csv_reader= csv.reader(file_csv)
```

Cette commande veut dire que pour chaque ligne dans le fichier lu ; chercher si un champ est identique à un champ dans la base **base1.csv**

```
For row in csv_reader:
```

```
    For field in row:
```

```
        If field== matricule.split(' ')[0]:
```

Si la voiture existe lors de l'exécution le commentaire suivant s'affiche :

```
            Print ('cette voiture existe')
```

III.4.5. Programme de la barrière d'accès :

Pour l'autorisation d'accès à travers une barrière nous avons utilisé pour notre prototype expérimental un servomoteur 9G SG90.

La programmation du servomoteur est initialisée par la déclaration des trois variables suivantes :

```
Min_duty= 3
```

```
Max_duty = 12
```

```
centre = Min_duty(max_duty-min_duty)/2
```

Ces trois variables présentent le minimum et le maximum ainsi que le centre de la rotation de la barrière d'accès.

Ensuite il faut définir le PIN choisi comme output à partir du Raspberry pour la PWM (Pulse Width Modulation) ou la Modulation de Largeur d'Impulsions pour déterminer la vitesse de rotation

```
Servo_pin = 05
```

```
Duty_cycle = centre
```

```
GPIO.Setmode( GPIO.BCM)
```

```
GPIO.setup(servo_pin,GPIO.OUT)
```

```
Pwm_servo = GPIO.pwm(servo_pin,50)
```

```
Pwm_servo.start (duty_cycle)
```

On a choisi de créer un canal PWM sur la broche du servomoteur avec une fréquence de 50Hz:

```
try
```

```
    while True:
```

```
        pwm_servo.ChangeDutyCycle(max_duty )
```

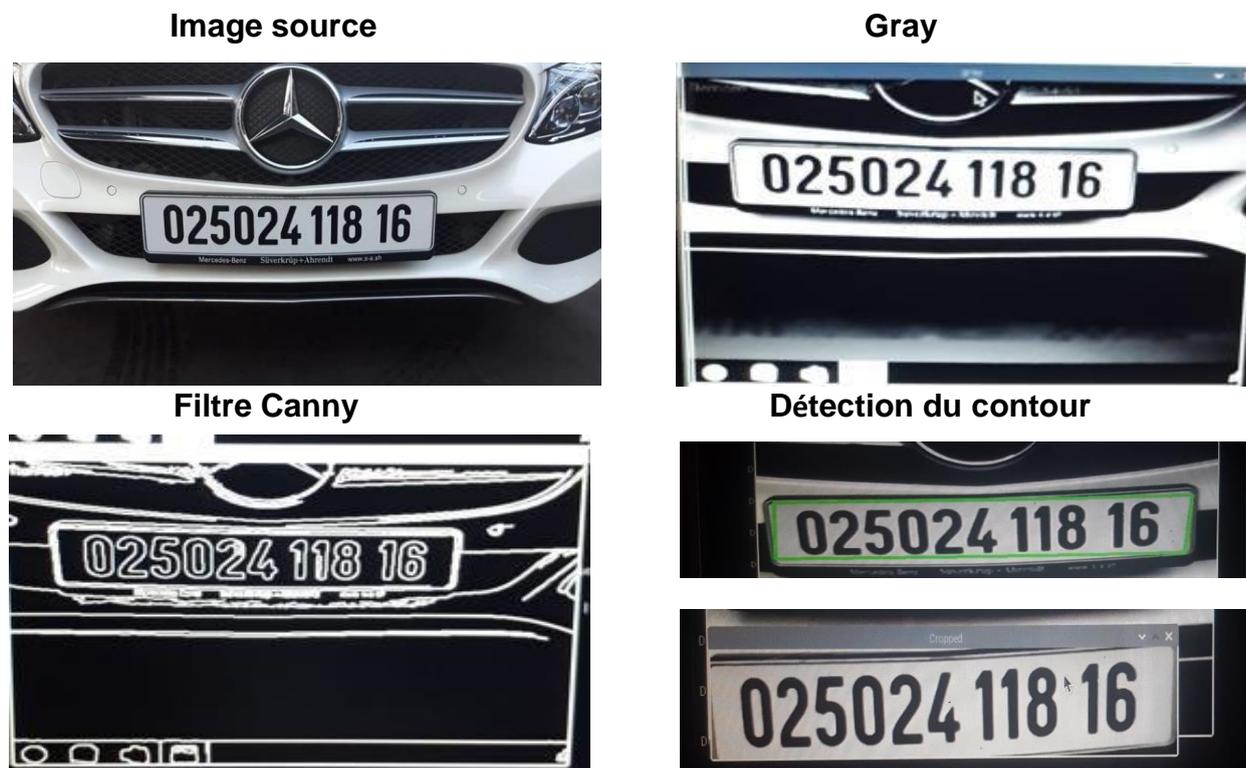
```
        time.sleep(10)
```

```
        pwm_servo.ChangeDutyCycle(min_DUTY)
```

Cette boucle est exécutée si la voiture est autorisée à accéder. Après l'ouverture de la barrière on a une temporisation de 10seconde pour permettre aux véhicules de rentrer puis la barrière est refermée une deuxième fois

```
except KeyboardInterrupt
    print "CTRL-C: Terminating program."
finally
    print "Cleaning up GPIO..."
    pwm_servo.ChangeDutyCycle(CENTRE)
    time.sleep(10)
    GPIO.cleanup()
```

Pour permettre aux véhicules de sortir le chauffeur peut utiliser un deuxième bouton comme commande manuelle pour la réouverture de la barrière. Les figures suivantes représentent la détection et la conversion en texte de quelques exemples de plaque d'immatriculation par notre système.



Résultat :

```
Detected license plate Number is: 025024 118 16 |
```

Figure III.8: Exemple 1 d'une détection de plaque d'immatriculation

Image source



Gray



Filtre Canny



Détection du contour



Résultat :

```
Detected license plate Number is: 027616 113 05
```

Image source



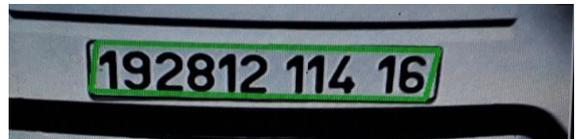
Gray



Filtre Canny



Détection du contour



Résultat :

```

Cette voiture existe
Detected license plate Number is: 192812 114 16

Cette voiture existe


```

Figure III.9: Exemple 2 d'une détection de plaque d'immatriculation

III.5. Conversion csv to PHP:

Pour la consultation à distance des accès de véhicules on a pensé à convertir la base de données base2.csv vers une page web

Cette étape consiste à afficher le contenu de la base de données CSV sur une page web en introduisant l'adresse IP du Raspberry dans la barre du navigateur, mais nous ne pouvons pas le faire directement ; il faut convertir le fichier CSV en PHP. Pour cela, nous avons besoin d'installer Apache2 et PHP. Avant installer le serveur web Apache2, il faut mettre une mise à jour pour le système, nous utilisons les commandes suivantes :

```

Sudo apt-get update
Sudo apt-get upgrade

```

Une fois la Raspberry Pi à jour, nous allons installer le serveur Apache2 par la commande :

```

sudo apt-get install apache2

```

Au passage, nous allons en profiter pour donner des droits au dossier d'apache qui nous permettra d'administrer facilement les sites. Pour cela, nous lançons les commandes suivantes:

```

sudo chown -R pi : www-data /var/www/html/
sudo chown -R 770 /var/www/html/

```

Une fois l'installation terminée, nous pouvons tester qu'Apache fonctionne correctement, nous utilisons la commande suivante :

```

wget -O verif_apache.html http://127.0.0.1

```

Cette commande va enregistrer le code HTML de la page dans le fichier

« verif_apache.html » dans le répertoire courant, après la lecture de ce fichier, nous

voyons marqué à un endroit dans le code « It Works! », ça veut dire qu'Apache est fonctionné.

Apache utilise le répertoire `/var/www/html` comme racine pour notre site. Cela signifie que quand nous appelons notre Raspberry sur le port 80 (http), Apache cherche le fichier dans `/var/www/html`. Pour ajouter de nouveaux fichiers, sites.., nous devons donc les ajouter dans ce répertoire. Nous pouvons dès à présent utiliser notre Raspberry pour faire un site en HTML, CSS et JavaScript pur, en interne.

Cependant, nous voulons rapidement permettre la conversion du fichier `vitesse.csv`, Pour cela, nous allons avoir besoin d'installer l'interpréteur PHP, nous utilisons la commande suivante :

```
sudo apt install php php-mbstring
```

Pour savoir si PHP fonctionne correctement, nous allons en premier lieu supprimer le fichier « `index.html` » dans le répertoire « `/var/www/html` », par la commande :

```
sudo rm /var/www/html/index.html
```

Puis créer un fichier « `index.php` » dans ce répertoire, avec cette commande :

```
echo '< ?php phpinfo() ; ?>' > /var/www/html/index.php
```

Après avoir installé `apache2` et PHP, nous pouvons maintenant écrire le programme qui va créer une page HTML pour afficher les informations de la base de données `vitesse.csv` sous forme d'un tableau.

Pour cela, nous avons entré au fichier `index.php` supprimé son contenu ; puis nous avons écrit ce script en PHP :

```
<html>
<head>
<title>CSV to PHP</title>
<style>

table {
  border-collapse: collapse;
  width: 80%;
  font-family: Gill Sans Extrabold, sans-serif;
  margin-left: 10%;
  margin-top: 10%;
}

th, td {
  text-align: left;
  padding: 8px;
}

tr:nth-child(even){background-color: #f2f2f2}

th {
  background-color: #4CAF50;
  color: white;
}

</style>
</head>
```

Dans cette première partie de code, nous avons spécifié le titre de code, la création d'un tableau ainsi la couleur et le style d'écriture. Ensuite, nous avons utilisé la fonction « fopen » pour sélectionner le nom du fichier CSV qui contient les informations que nous voulons traduire en tableau. Après, nous avons spécifié le contenu de tableau, ce tableau contient 3 colonnes :

- La première colonne pour la date et l'heure actuelle.
- La deuxième colonne pour la conversion en texte du matricule détecté.
- La troisième colonne pour l'image de matricule détecté.

Puis, nous avons utilisé la fonction « fgetcsv » pour lire physiquement le fichier spécifié dans « fopen » avec son paramètre : le délimiteur, pour éliminer n'importe quel caractère de séparation (point-virgule, virgule) dans le fichier CSV, et enfin nous utilisons la fonction « fclose » pour fermer le fichier CSV

```

<br/>
<br/>
<br/>
<tr>
<th width="91"> <div align="center">DATE & TIME</div></th>
<th width="80"> <div align="center">MATRICULE</div></th>
<th width="80"> <div align="center">CAPTURE</div></th>
</tr>

<?php
while (($objArr = fgetcsv($objCSV, 1000, ",")) !== FALSE) {
?>

<tr>
<td><div align="center"><?php echo $objArr[0];?></div></td>
<td><div align="center"><?php echo $objArr[1];?></div></td>
<td><div align="center"><?php echo "<img src='\"' . $objArr[2]. \"' height='60' width='180'>";?></div></td>
</tr>

```

Lorsque nous avons terminé le code et l'enregistrer, nous pouvons faire un test : en tapant l'adresse IP de Raspberry dans la barre de recherche de navigateur à partir de Raspberry pi ou bien un périphérique connecté au même réseau local.

La base de données pour les exemples présentés précédemment et la page web créée pour ces exemples s'affiche comme suivant :

```

30/05/2022 - 12:42:14 ,025024 118 16 , img1.jpg,
30/05/2022 - 12:44:34 ,192812 114 16 , img2.jpg,

```

Figure III.10: Base de données base2.csv



DATE & TIME	MATRICULE	CAPTURE
30/05/2022 - 12:42:14	025024 118 16	
30/05/2022 - 12:44:34	192812 114 16	

Figure III.11: Page web affichée pour quelques exemples de plaques d'immatriculation

Pour le contrôle à distance de notre système on utilise le VNC Viewer de RealVNC qui permet d'accéder instantanément à distance au Raspberry, de n'importe où dans le monde. Il permet de visualiser graphiquement le bureau du Raspberry, et de contrôler sa souris et son clavier.

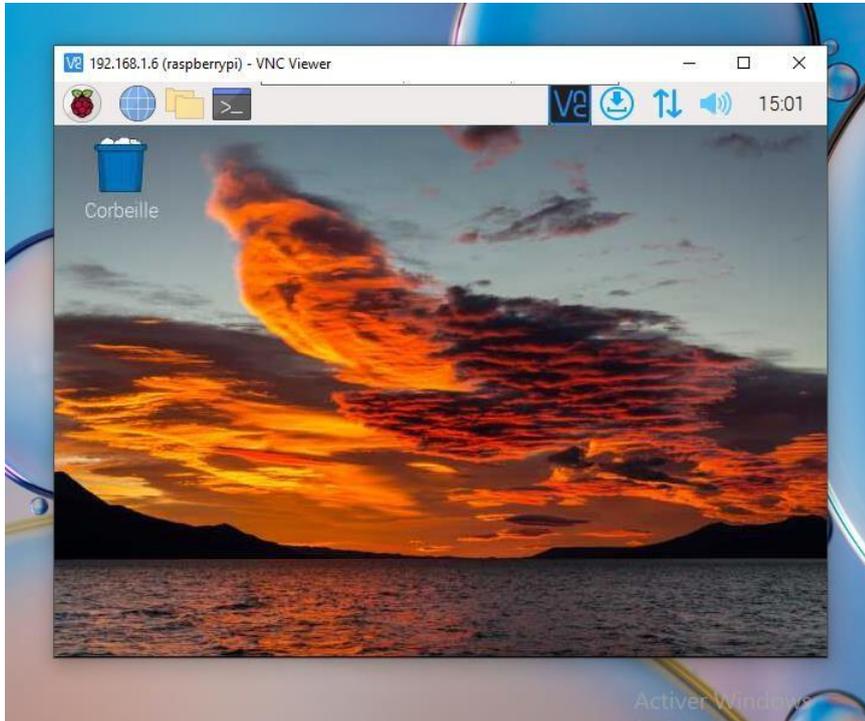


Figure III.12: VNC Viewer

III.6. La maquette réalisée :

Notre maquette expérimentale est un parking en miniature réalisé sous impression 3D, nous avons placé notre webcam juste en face de l'entrée pour avoir la bonne capture des plaques d'immatriculation. Quand la caméra capture le matricule du véhicule positionné en face et vérifie son existence dans la base de données, le système déclenche l'ouverture de la barrière ou interdit l'accès.

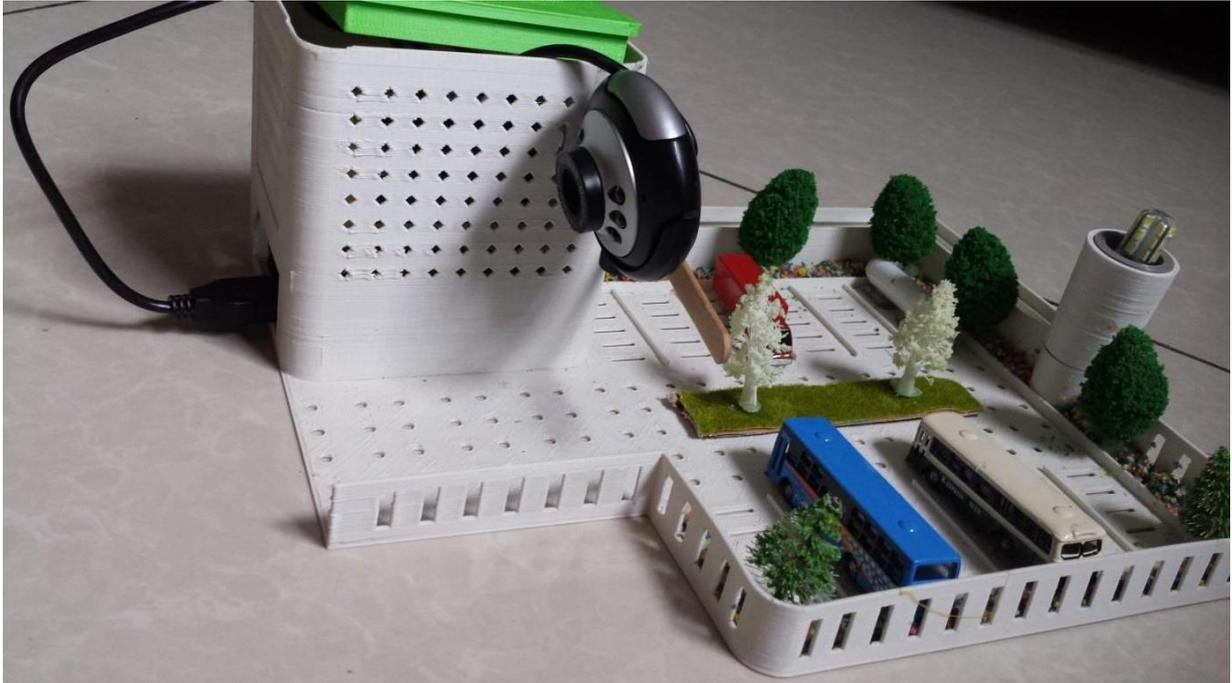


Figure III.13: Maquette réalisée pour notre système

Conclusion du Chapitre III :

Dans ce chapitre nous avons détaillé les différentes parties de notre programme en commençant par l'importation des différentes bibliothèques et modules nécessaire, puis les différentes étapes utilisées pour la détection de la plaque et la vérification de son existence pour autoriser l'accès par l'ouverture de la barrière.

Ensuite on a ajouté un affichage des autorisations sur une page web dans un tableau qui présente la date et l'heure d'accès, le matricule détecté en texte puis en image à l'aide d'une conversion csv vers PHP avec un programme rédigé en langage html.

Conclusion Générale

Conclusion générale :

Notre mémoire a pour objectif la réalisation d'un système de contrôle automatique d'accès de véhiculé par la reconnaissance optique des plaques d'immatriculation en utilisant la carte Raspberry Pi. Ce travail est fait en trois grandes parties :

Dans la première partie, nous avons rappelé l'intérêt d'un système de contrôle d'accès automatique. Nous nous sommes intéressés aux différents types et du principe de fonctionnement de ces systèmes avec les technologies actuelles.

Dans la deuxième partie nous avons présenté la technique de détection, ainsi que le matériel et le langage de programmation utilisés dans notre projet. Nous avons commencé par la définition de la méthode de détection utilisé dans notre système. Ensuite, nous avons exposé les environnements hardware et software. Nous avons présenté le Raspberry Pi avec ses différents modèles en justifiant le choix de Raspberry Pi Modèle 3 B, la webcam, le servomoteur. Puis, nous avons spécifié le coté software où nous avons résumé le système d'exploitation Raspbian, langage Python et les bibliothèques utilisées.

La troisième partie qu'est la réalisation expérimentale, où nous avons présenté les étapes de la conception et l'analyse. En premier lieu, nous avons commencé par l'installation et la définition des différentes bibliothèques et modules du langage de programmation Python 3. Puis, nous avons écrit un programme qui permet la détection des plaques d'immatriculation et leurs stockages dans une base de données csv.

Nous avons par la suite créé une page HTML pour avoir un affichage à distance des informations collectées des différents accès autorisé sous un réseau local.

Ce projet ouvre plusieurs perspectives, comme la possibilité de consulter ces informations dans un navigateur web à partir d'un réseau externe, ainsi que l'amélioration du programme avec l'ajout de la détection des visages des conducteurs

Bibliographie :

- [1] : Système OCR - Qu'est ce qu'un système OCR ?,
<https://sumup.fr/factures/termes-comptables/systeme-ocr/>
- [2] : Reconnaissance optique des caractères, Wikipédia,
https://fr.wikipedia.org/wiki/Reconnaissance_optique_de_caract%C3%A8res
- [3]: Système OCR - Qu'est ce qu'un système OCR ?,
<https://sumup.fr/factures/termes-comptables/systeme-ocr/>
- [4] :**AKACEM Oum el kheir, RAHMANI Nassira**, Mémoire de fin d'étude, Système de reconnaissance des plaques d'immatriculation Algérienne
- [5] : La reconnaissance optique des caractères,
<https://moov.ai/fr/blog/reconnaissance-optique-de-caracteres-ocr/#:~:text=La%20reconnaissance%20optique%20des%20caract%C3%A8res,u n%20fichier%20de%20format%20texte.>
- [6] : <https://moov.ai/fr/blog/reconnaissance-optique-de-caracteres-ocr/>
- [7] :**Aboubacar Beidari Abdoul Nasser, Bourezg Mohamed**, Mémoire de fin d'études, Développement d'une application de fiable détection et de reconnaissance de plaques d'immatriculation
- [8] : Plaque d'immatriculation algérienne,
https://fr.wikipedia.org/wiki/Plaque_d%27immatriculation_alg%C3%A9rienne
- [9] :**AKACEM Oum el kheir, RAHMANI Nassira**, Mémoire de fin d'étude, Système de reconnaissance des plaques d'immatriculation Algérienne
- [10] :**Krim Smail**, Mémoire de Master, Reconnaissance Automatique des plaques d'immatriculation (R.A.P.I) (Implémentation sur Raspberry Pi)
- [11] : 10 outils de manipulation et traitement d'images en python,
<https://moncoachdata.com/blog/10-outils-de-traitement-dimages-en-python/>
- [12] : Traitement d'images,
https://fr.wikipedia.org/wiki/Traitement_d%27images
- [13] : **Krim Smail**, Mémoire de Master, Reconnaissance Automatique des plaques d'immatriculation (R.A.P.I) (Implémentation sur Raspberry Pi)
- [14] : Raspberry Pi Introduction,
<https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction>
- [15] : Filtre de Canny, https://fr.wikipedia.org/wiki/Filtre_de_Canny
- [16] : L'histoire incroyable du Raspberry Pi,
<https://raspberrytips.fr/histoire-du-raspberry-pi/>
- [17] : <https://www.ldlc.com/fiche/PB00205573.html>
- [18] : Scratch et Raspberry Pi : Composant,
<https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/>
- [19] : Choisir et utiliser une caméra sur un Raspberry Pi, Prendre photos et vidéos, <https://raspberry-lab.fr/Composants/Utilisation-Camera-sur-Raspberry-Pi-Francais/>

[20] : Le bus DSI du Raspberry Pi 3 B+, [https://www.editions-
eni.fr/open/mediabook.aspx?idR=906fbd0ca61db53845943061514dca60](https://www.editions-
eni.fr/open/mediabook.aspx?idR=906fbd0ca61db53845943061514dca60)

[21] : Configurer ton Raspberry Pi, [https://projects.raspberrypi.org/fr-
FR/projects/raspberry-pi-setting-
up/1#:~:text=Les%20Raspberry%20Pi%201%2C%202.un%20c%3%A2
ble%20standard%20HDMI%2DHDMI](https://projects.raspberrypi.org/fr-
FR/projects/raspberry-pi-setting-
up/1#:~:text=Les%20Raspberry%20Pi%201%2C%202.un%20c%3%A2
ble%20standard%20HDMI%2DHDMI)

[22] : Processor, <https://www.raspberrypi.com/documentation/computers/processors.html>

[23] : Raspberry Pi : Signification des LEDs Rouge et Verte (Table), [https://raspberrytips.fr/led-rouge-verte-raspberry-
pi/#:~:text=Les%20voyants%20lumineux%20rouges%20et,est%20suffisa
mment%20aliment%3%A9%20ou%20non](https://raspberrytips.fr/led-rouge-verte-raspberry-
pi/#:~:text=Les%20voyants%20lumineux%20rouges%20et,est%20suffisa
mment%20aliment%3%A9%20ou%20non).

[24] : Caméra Raspberry-Pi - 8MegaPixels - V2, [https://shop.mchobby.be/fr/pi-
camera/858-camera-raspberry-pi-8-megapixels-
3232100008588.html#:~:text=Camera%20Pi%20officielle%20avec%20une%20
r%3%A9solution%20de%208%20Megapixel&text=La%20cam%3%A9ra%2
0se%20branche%20%3%A0,donn%3%A9es%20graphiques%20\(pixel%20d
ata\).](https://shop.mchobby.be/fr/pi-
camera/858-camera-raspberry-pi-8-megapixels-
3232100008588.html#:~:text=Camera%20Pi%20officielle%20avec%20une%20
r%3%A9solution%20de%208%20Megapixel&text=La%20cam%3%A9ra%2
0se%20branche%20%3%A0,donn%3%A9es%20graphiques%20(pixel%20d
ata).)

[25] : Écran HDMI 5 pouces 800 × 480 TFT avant Raspberry Pi 3B +, [https://elektronicavoorjou.nl/fr/Les-produits/%3%A9cran-tft-hdmi-5-pouces-
800x480-pour-raspberry-pi-b2b3b/](https://elektronicavoorjou.nl/fr/Les-produits/%3%A9cran-tft-hdmi-5-pouces-
800x480-pour-raspberry-pi-b2b3b/)

[26] : <https://www.raspberrypi-france.fr/guide/raspian-programmer-python/V>

[27] : OpenCV, c'est quoi ?, [https://www.axopen.com/blog/2019/09/open-cv-
cest-quoi/](https://www.axopen.com/blog/2019/09/open-cv-
cest-quoi/)

[28] : **locale** — Internationalization services, <https://docs.python.org/3/library/locale.html>

[29] : Comment configurer le système d'exploitation Raspberry Pi sur un Raspberry Pi, [https://www.moyens.net/tech/comment-configurer-le-systeme-
dexploitation-raspberry-pi-sur-un-raspberry-pi/](https://www.moyens.net/tech/comment-configurer-le-systeme-
dexploitation-raspberry-pi-sur-un-raspberry-pi/)

[30] : Se connecter en SSH à son Raspberry Pi sous Raspbian, [https://www.raspberrypi-france.fr/guide/connecter-ssh-
raspbian/#:~:text=Le%20SSH%2C%20aussi%20appel%3%A9%20%3A%20%20C2%AB
%20Secure,clavier%20et%20de%20la%20souris](https://www.raspberrypi-france.fr/guide/connecter-ssh-
raspbian/#:~:text=Le%20SSH%2C%20aussi%20appel%3%A9%20%3A%20%20C2%AB
%20Secure,clavier%20et%20de%20la%20souris).

[31] : Connexion à distance au Raspberry Pi

<https://www.aranacorp.com/fr/connexion-a-distance-au-raspberry-pi/>

[32] : <https://www.bibl.ulaval.ca/geostat/statistiques/CSV-guide.pdf>

[33] : Github, <https://fr.wikipedia.org/wiki/GitHub>

[34]: Python time models,

<https://www.programiz.com/python-programming/time>

[35]: **locale — Services d'internationalisation,**

<https://docs.python.org/fr/3.6/library/locale.html#module-locale>

[36] : **datetime** — Basic date and time types,

<https://docs.python.org/3/library/datetime.html>

[37] : Le module CSV – lire et écrire dans des fichiers,

<https://pythonforge.com/module-csv-lire->

[ecrire/#:~:text=Le%20module%20csv%20en%20Python,%C3%A9crire%20dans%20des%20fichiers%20CSV.](https://pythonforge.com/module-csv-lire-ecrire/#:~:text=Le%20module%20csv%20en%20Python,%C3%A9crire%20dans%20des%20fichiers%20CSV.)

[38] : Quel est la différence entre une image matricielle ou vectorielle ?,

<https://www.baches-publicitaires.com/blog/actualites/vectorisation->

[cest/#:~:text=La%20principale%20diff%C3%A9rence%20entre%20ces,pa](https://www.baches-publicitaires.com/blog/actualites/vectorisation-cest/#:~:text=La%20principale%20diff%C3%A9rence%20entre%20ces,pa)
[s%20%C3%A0%20l'%C5%93il%20nu.](https://www.baches-publicitaires.com/blog/actualites/vectorisation-cest/#:~:text=La%20principale%20diff%C3%A9rence%20entre%20ces,pa)

[39] : Python OpenCV: Converting an image to gray scale,

<https://techtutorialsx.com/2018/06/02/python-opencv-converting-an-image-to-gray-scale/>

[40] : [Python] Using OpenCV with Python (Edge Detection),

<https://linuxtut.com/en/494cc16836738b5394c8/>

[41] : License Plate Recognition using OpenCV in Python,

<https://www.codespeedy.com/license-plate-recognition-using-opencv-in-python/>

[42] : Le rôle d'utils. grab_contours,

<https://blog.spacepatroldelta.com/a?ID=01050-219e9a3f-9120-4080-ba3c-615dc6c5a34c>

[43]: MEMOIRE Présenté pour obtenir le diplôme de

MASTER EN ELECTRONIQUE DES SYSTEME EMBARQUES

Par Ould Mammar Mohammed Elamine et Midoun Mohamed Essedik

Résumé :

Un système de contrôle d'accès de véhicule est un instrument de détection des véhicules et d'identification de plaque d'immatriculation utilisé pour faciliter les tâches d'autorisation et d'interdiction d'accès aux différents parkings

Notre projet consiste à réaliser un système autonome de contrôle d'accès à l'aide d'un Raspberry Pi. Pour cela, nous avons présenté le matériel utilisé et les démarches d'installation d'un système d'exploitation et les logiciels. Puis, nous avons utilisé la technique de la détection des plaques d'immatriculation, ainsi que la reconnaissance optique des caractères écrits pour réaliser des recherches et des enregistrements dans des bases de données utilisées. Enfin, nous avons affiché ces informations collectées dans ces bases de données dans une page HTML pour la consultation à distance.

Mots-clés : OCR, Contrôle d'accès, Raspberry Pi, Python3.

Abstract:

A vehicle access control system is a vehicle detection and license plate identification instrument used to facilitate the tasks of authorizing and prohibiting access to the various car parks.

Our project consists in realizing an autonomous access control system using a Raspberry Pi. For this, we have presented the equipment used and the procedures for installing an operating system and software. Then we used the technique of detection of license plates, as well as optical recognition of written characters to carry out searches and recordings in the databases used. Finally, we have displayed this detailed information in these databases in a page HTML for remote viewing.

Key-words: OCR, access control, Raspberry Pi, Python3..

ملخص:

نظام التحكم في ولوج السيارات هو أداة للكشف عن المركبات وتحديد هوية لوحة الترخيص تُستخدم لتسهيل مهام الترخيص و حظر الوصول إلى مواقف السيارات المختلفة يتمثل مشروعنا في تحقيق نظام تلقائي للتحكم في دخول السيارات باستخدام Raspberry Pi لهذا، قدمنا المعدات المستخدمة و شرحنا إجراءات تثبيت نظام التشغيل و البرمجيات. ثم استخدمنا تقنية الكشف عن لوحات الترخيص و كذلك التعرف على الحروف المكتوبة لإجراء عمليات البحث و التسجيل في قواعد البيانات المستخدمة و أخيراً، عرضنا هذه المعلومات التي تم جمعها في قواعد البيانات هذه في إحدى الصفحات HTML للاطلاع عنها من بعد.

، Raspberry Pi ، Python 3 ، OCR الكلمات المفتاحية: التحكم في ولوج السيارات،