الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد– تلمسان
Université Aboubakr Belkaïd-Tlemcen
كلية التكنلوجيا
Faculté de Technologie

Département de Génie Electrique et Electronique (GEE)
Filière : Electronique

MASTER INSTRUMENTATION

PROJET DE FIN D'ETUDES

Présenté par : SI ABDELKADER SOUMIA **&** SOUFI ZINEB MERIEM

Intitulé du Sujet

Conception et réalisation d'un système de contrôle routier à l'aide d'un Raspberry Pi

Soutenu en 2021, devant le jury composé de :

Mr MASSOUM Noureddine	MCB	Univ. Tlemcen	Président
Mr BENHABIB Mohamed Choukri	MCA	Univ. Tlemcen	Examinateur
Mr MOULAI KHATIR Ahmed Nassim	MCB	Univ. Tlemcen	Encadreur

Année Universitaire 2020-2021

Remerciements

Louange à Dieu tout puissant, qui nous a permis de voir ce jour

Ce travail a été effectué au sein de l'Université Abou Bekr Belkaid de Tlemcen et plus précisément au département de génie électrique et électronique de la faculté de technologie.

Nous tenons à remercier d'abord ***Mr. Moulai Khatir Ahmed Nassim****, Maître de Conférences Classe B à l'université de Tlemcen, qui nous a dirigées et encadrées durant ce travail ainsi que de nous avoir honorées en nous proposant ce sujet. Nous le remercions pour sa disponibilité, ses critiques et ses remarques pertinentes.*

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

Les remerciements sont, également, adressés à ***Mr. BENHABIB Mohamed Choukri****, Maître de Conférences Classe A à l'université de Tlemcen, d'avoir accepté d'examiner ce travail et de nous faire l'honneur de participer au jury.*

Je remercie mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

Un grand remerciement à mes chers Sœurs, Frères et mes Neveux pour leurs encouragements, et leur grand soutien durant toutes ces années d'études.

Un remerciement particulier à ma chère sœur " Hayet " pour son aide durant mon parcours universitaire, ses conseils, et son encouragement permanent et leur soutien moral.

Soumia

Je remercie mes très chers parents KHEIREDINE et NAOUAL qui ont toujours été la pour moi je remercie aussi mes frères ABDELKADER ISMAIL et IBRAHIM pour leur encouragement ainsi je remercie ma partenaire de mémoire, ma binôme, mon amie SOUMIA pour son soutien inconditionnel et sa sincère amitié et confiance.
À tous ces intervenants, je présente mes remercîments, mon respect et ma gratitude.

Zineb meriem

Sommaire

Introduction générale **2**

Chapitre I : Généralités sur les radars de contrôle routier **3**
I.1. Introduction 4
I.2. Historique du radar 4
I.3. Définition du radar 5
I.4. Différents types du radar automobile 5
I.4.1. Radar fixe 5
I.4.2. Radar mobile 6
I.4.3. Radar tronçon 7
I.5. Principe de fonctionnement du radar 7
I.6. Angle de détection 8
I.7. Conclusion 9

Chapitre II : Etude et réalisation du système de contrôle routier **10**
II.1. Introduction 11
II.2. Spécification de la technique et langage utilisé 11
II.3. Procédure de détection de Haar-cascade 11
II.4. Type de fichier utilisé 13
II.4.1. Fichier XML 13
II.4.2. Fichier CSV 14
II.4.3. Niveau de gris 14
II.5. Environnement de développement 14
II.5.1. Environnement hardware 14
II.5.1.1. Un ordinateur 14
II.5.1.2. Un Raspberry Pi 15
II.5.1.2.1. Présentation du Raspberry pi 15
II.5.1.2.2. Les modèles de Raspberry Pi 15
II.5.1.2.3. Le Raspberry Pi 3 modèle B 20
II.5.1.3. Un Ecran TFT LCD 5’’ 22
II.5.1.4. Une caméra Pi 22
II.5.2. Environnement software 23
II.5.2.1. Raspbian 23
II.5.2.2. Langage Python 23
II.5.2.3. Bibliothèques (modules) utilisées 23
II.5.2.3.1. Time 23
II.5.2.3.2. Imutils 23
II.5.2.3.3. NumPy 24
II.5.2.3.4. OpenCV 24
II.5.2.3.4.1. Qu’est-ce que OpenCV 24

II.5.2.3.4.2. Historique 24
II.5.2.3.4.3. Fonctionnalités 25
II.6. Conclusion 26

CHAPITRE III : Conception et analyse **27**
III.1. Introduction 28
III.2. Matériel de conception 28
III.3. Préparation du système d'exploitation sur micro SD 29
III.3.1. Formater la carte SD 29
III.3.2. Installer Raspbian sur la carte SD 29
III.3.3. Premier démarrage du système 31
III.4. Première configuration 31
III.5. Mettre à jour le système de la Raspberry Pi 35
III.6. Activer la caméra Pi 35
III.7. Installation OpenCV sur le Raspberry Pi 35
III.8. Programme python 36
III.9. Création de la page HTML 47
III.10. Comment pouvons-nous avoir un système autonome ? 50
III.11. Conception et Organigramme du système de contrôle routier à base de Raspberry 50
III.11.1. Conception du système 50
III.11.2. Organigramme du système 51
III.12. Circuit du système 53
III.13. Conclusion 54

Conclusion générale **56**
Références Bibliographique 57

Table des figures

I.1 Exemple d'un radar fixe 6
I.2 Exemple des radars mobiles classiques 6
I.3 Exemple d'un radar mobile de nouvelle génération 7
I.4 Exemple d'un radar tronçon 7
I.5 Angle de détection 8

II.1 Exemple des premières caractéristiques pseudo-Haar utilisées par Viola et Jones 12
II.2 Exemple de calcul d'image intégrale 13
II.3 Raspberry pi modèle A 15
II.4 Raspberry pi modèle A+ 16
II.5 Raspberry pi modèle 1 B 16
II.6 Raspberry pi modèle 1 B+ 17
II.7 Raspberry pi modèle 2 B 17
II.8 Raspberry pi modèle 3 B 18
II.9 Raspberrypi modèle 3 B+ 18
II.10 Raspberrypi modèle 4 B 19
II.11 Raspberrypi modèle Zero 19
II.12 Rasspbery Pi modèle Zero W 20
II.13 Raspberry Pi modèle Zero WH 20
II.14 Raspberry Pi Modèle 3 B utilisé dans notre projet 21
II.15 Les composants standards d'un Raspberry Pi 3 Modèle B 21
II.16 Ecran TFT LCD 5'' pour Raspberry 22
II.17 Caméra Pi utilisée dans notre projet 22

III.1 Raspberry Pi et ses accessoires 28
III.2 Formatage de la carte SD 29
III.3 Décompression du fichier Zip 30
III.4 Copier le raspbian sur la carte SD 30
III.5 Installation de Raspbian sur la carte SD 31
III.6 Démarrage du Raspberry Pi 3 31
III.7 Début de la configuration du Raspberry Pi 3 32
III.8 Configuration de la langue et la localisation 32
III.9 Modification du mot de passe 33
III.10 Activation de wifi 33
III.11 Activation de SSH 34
III.12 Configuration de Putty 34
III.13 Activation de VNC 35
III.14 Fichier Haar cascade placé dans PFE 37
III.15 Détection de la voiture à l'aide de Haar-cascade 40
III.16 Calcul et affichage de la vitesse 41

III.17 Etapes du traitement d'image pour la détection de la plaque d'immatriculation 42
III.18 Résultat après application du masque PFE 43
III.19 Recadrage de la région du matricule 44
III.20 Fichier CSV 45
III.21 Exemple de la détection de la plaque d'immatriculation 46
III.22 Exemple de la page web créée 50
III.23 Organigramme du système 52
III.24 Schéma descriptif du système 53
III.25 Prototype expérimental global 53

Liste des abréviations

RADAR : Radio Detecting And Ranging

HAAR : High Altitude Acute Response (Réponse aiguë à haute altitude)

AdaBoost : Adaptive Boosting

XML : Extensible Markup Language(Extensible Markup (langue))

CSV : Comma separated values (valeur séparée par des vergules)

HDMI : High definition multimedia interface (interface multimédia haute définition)

USB : Universal Serial Bus (Autobus de série universel)

GPIO : General - Purpose Input/Output (Généralités - Entrée/sortie d'objet)

CPU : Unité central de traitement

LCD : Liquid Crystal Display

OpenCV : Open Computer Vision

SSH : Secure Shell (Ligne de commande sécurisée)

GITHUB: A contraction of the words Git and Hub (Une contraction des mots Git et Hub)

HTML: HyperText Markup Language (HyperText Markup (langue))

PHP: Hypertext Preprocessor(Préprocesseur hypertexte)

OCR: Optical Character Recognition

Introduction générale

Chaque jour, il y a un certain nombre de décès, le plus grand nombre dû aux accidents de la route, les excès de vitesse et le non-respect des règles de circulation constituent des principales causes. Un constat qui pousse le gouvernement à augmenter sans cesse le nombre de radars sur les routes.

Le terme RADAR est issu de l'acronyme anglais RAdio Détection And Raning : c'est un système indispensable et très largement utilisé qu'il sert à surveiller et éviter des collisions dans les routes, ce système de contrôle est lancé en 1865, il a connu un essor rapide durant la Seconde Guerre mondiale, et Jusqu'à présent, les chercheurs travaillent toujours à son développement.

Le but principal de notre projet est l'étude et la conception d'un radar de détection à l'aide d'un Raspberry Pi, d'où trois objectifs ont été visés :

a) Faire une étude générale sur la théorie des radars.

b) Regrouper suffisamment d'informations sur l'une des cartes de prototypage: Raspberry Pi.

c) Réaliser un système de contrôle pour la détection des véhicules à l'aide des composants disponibles.

Afin de réaliser ce projet, nous avons adopté deux parties principales : partie matérielles et partie logicielles. La partie matérielles est basé sur le mini-ordinateur incroyable qui a été lancé en 2012, ainsi ses accessoires, son camera Pi et un afficheur LCD. La deuxième partie est à propos les logiciels : système d'exploitation ainsi le langage de programmation Python et ses bibliothèques.

Ce Manuscrit est réparti en trois chapitres :

- Le premier chapitre est une brève présentation générale sur les radars de contrôle routier, leurs modes de fonctionnement, ainsi que l'angle de détection.
- Le deuxième chapitre présente les différents outils matériels et logiciels nécessaires à la conception du radar.
- Le troisième chapitre est consacré à la conception et la réalisation pratique d'un radar automobile.

Enfin nous allons terminer par une conclusion générale qui résume l'essentiel du travail réalisé.

Chapitre I

Généralités sur les radars de contrôle routier

I.1. Introduction :

Le terme RADAR est l'acronyme de "Radio Detecting And Ranging" qui signifie la détection et la localisation à distance d'un objet.

L'introduction du radar au marché automobile émane de la volonté des constructeurs automobiles d'ajouter plus de confort et de sécurité aux véhicules, en apportant au conducteur plus d'informations sur son environnement, dans un premier temps ; et en autorisant le véhicule à manœuvrer sans intervention du conducteur, dans un second temps. D'ailleurs, c'est le développement du radar automobile en association avec d'autres technologies comme la reconnaissance par caméra, appelée vision, et la détection par laser qui ont amené la voiture autonome au monde réel depuis le monde de la science-fiction [1].

I.2. Historique du radar :

Le RADAR est l'une des merveilles du vingtième siècle. C'est un système électromagnétique utilisé pour détecter la présence des objets mobiles et déterminer leur trajectoire, leur vitesse, et d'autres données.

Il serait vain d'attribuer l'invention du Radar à un savant, ou à une nation unique. On doit considérer le « Radar » comme le résultat de l'accumulation de nombreuses recherches menées antérieurement, et auxquelles les scientifiques de plusieurs pays ont parallèlement participé [2].

- En 1865, le physicien anglais James Clerk Maxwell développe sa théorie de la lumière électromagnétique et en 1886, le physicien allemand Heinrich Rudolf Hertz démontra l'existence physique des ondes électromagnétiques qui confirment ainsi la théorie de Maxwell.
- En 1904, le technicien allemand spécialiste des ondes hertziennes, Christian Hülsmeyer, invente le 'Telemobiloskop' appareil de mesure le temps de parcours de l'onde électromagnétique. Donc, le calcul de distance est possible. C'est un premier test pratique d'un appareil qui suit les mêmes principes d'un radar plus tard. Hülsmeyer dépose un brevet de son invention en Allemagne et en France.
- En 1921, Albert Wallace Hull développe un oscillateur à haut rendement, le magnétron, qui servira plus tard comme source de l'onde radar.

- En 1935, suite à un brevet déposé par Robert Watson-Watt (l'inventeur dit « officiel » du radar), le premier réseau de radars est commandé par les Britanniques et portera le nom de code Chain Home.
- En 1936, Metcalf et Hahn développent le klystron. Utilisé comme amplificateur ou oscillateur, qui est un équipement important du radar.

Différents équipements radar sont développés aux USA, en Russie, en Allemagne, en France et au Japon, accélérées par la montée en puissance vers une guerre qui semble inévitable, et par le développement général de l'arme aérienne. La seconde guerre mondiale a été à l'origine de plusieurs innovations techniques telles que la technologie radar. Pendant la guerre froide, des radars sont déployés en grande quantité de part et d'autre du «rideau de fer», et en particulier le long des frontières allemandes.

I.3. Définition du radar :

Un radar automobile est un ensemble des moyens matériels et logiciels qui communiquent entre eux afin de réaliser la fonction de détection de position et de vitesse des véhicules circulant sur la voie publique. On peut le rencontrer sur le bord des routes signalées par un panneau spécifique, et fonctionne seul sans l'intervention d'un policier ou gendarme.

I.4. Différents types du radar automobile :

Les radars ont été installés sur les routes pour lutter contre l'insécurité routière. Il existe plusieurs types du radar et différentes manières de le contrôler [3] :

I.4.1. Radar fixe :

Il contrôle la vitesse des conducteurs dans les zones les plus dangereuses de route d'un seul sens (*Les radars fixes classiques*) ou dans les deux sens (*Les radars fixes double sens*). Il peut flasher aussi bien de face que par l'arrière, c'est le premier type de radar qui a été installé. Il se trouve dans des cabines fixes à la bordure des routes (Figure I.1), Tous les radars de vitesse fixe sont signalés par des panneaux d'annonce radars.

Figure I.1 : Exemple d'un radar fixe

I.4.2. Radar mobile :

Le Radar mobile est appelé aussi radar embarqué, est l'un des outils permettant de contrôler la vitesse d'un véhicule mais il diffère du premier type, celui-ci n'est pas fixe. Il s'agit d'un radar portatif, ou se placer dans des véhicules de police ou de gendarmerie sans aucune indication. Il existe deux types des radars mobiles :

- *Les radars mobiles classiques* : il calcule de la vitesse dans un seul sens de circulation, il est utilisé sur un trépied ou dans une voiture comme indiqué la Figure I.2.
- *Les radars mobiles de nouvelle génération* : il calcule de la vitesse dans les deux sens de circulation, à l'arrêt ou en circulation par flash infrarouge invisible. Le dispositif de photographie infrarouge est situé dans la plaque avant de la voiture comme indiqué la Figure I.3.

Figure I.2 : Exemple des radars mobiles classiques

Figure I.3 : Exemple d'un radar mobile de nouvelle génération

I.4.3. Radar tronçon :

Le radar tronçon est un radar automatique, qui ne calcule pas la vitesse des voitures à un instant précis comme les autres radars, mais plutôt sur une portion de route. Son fonctionnement est simple, il utilise deux caméras de surveillances reliées à un lecteur de plaque d'immatriculation pour enregistrer les heures de passage des véhicules entre deux points de distance connue (Figure I.4).

Figure I.4 : Exemple d'un radar tronçon

I.5. Principe de fonctionnement du radar :

Le principe de fonctionnement d'un radar automobile est basé sur le principe de l'effet Doppler-Fizeau qui permet de calculer la vitesse des véhicules sur la route. Ces radars émettent une onde entretenue qui est réfléchie par toute cible se trouvant dans la

direction pointée (véhicule).Cette onde réfléchie possède une fréquence légèrement différente de celle émise : plus grande fréquence pour les véhicules proches du radar et plus petite pour ceux qui s'éloignent du radar. La variation de la fréquence entre l'onde émise et celle retournée (effet Doppler-Fizeau) permet de calculer la vitesse de la cible, cela est fait par le calcul du battement entre les deux ondes.

I.6. Angle de détection :

Les systèmes de contrôle routiers sont orientés suivant un angle de 25° par rapport à l'axe de la route, lorsque l'angle est plus faible que 25° par rapport à l'axe de route sera défavorable ou plus grand a 25° sera favorable aux conducteurs. Un radar mesure la vitesse sous un angle de 30° par rapport à l'axe de circulation des véhicules contrôlés, cet angle est très important car il y'a un différentiel de seulement 5° le contrôle effectue une minoration de la vitesse détectée de 4.5%.

C'est la direction de déplacement du véhicule par rapport au radar qui est importante pas seulement l'angle de positionnement du radar par rapport à l'axe de la route.

Pour des mesures correctes de la vitesse radiale entre le radar et le véhicule, il faut que les véhicules se déplacent avec un angle de rotation en même avec le radar. La vitesse est que la projection sur la radiale au radar, soit la vitesse réelle multipliée par le cosinus de l'angle. Pour que les mesures réalisées par les radars fixes ou mobiles soient exactes elles doivent être réglées à 25° par rapport à l'axe du déplacement [4]. L'angle de détection est donné à la Figure I.5.

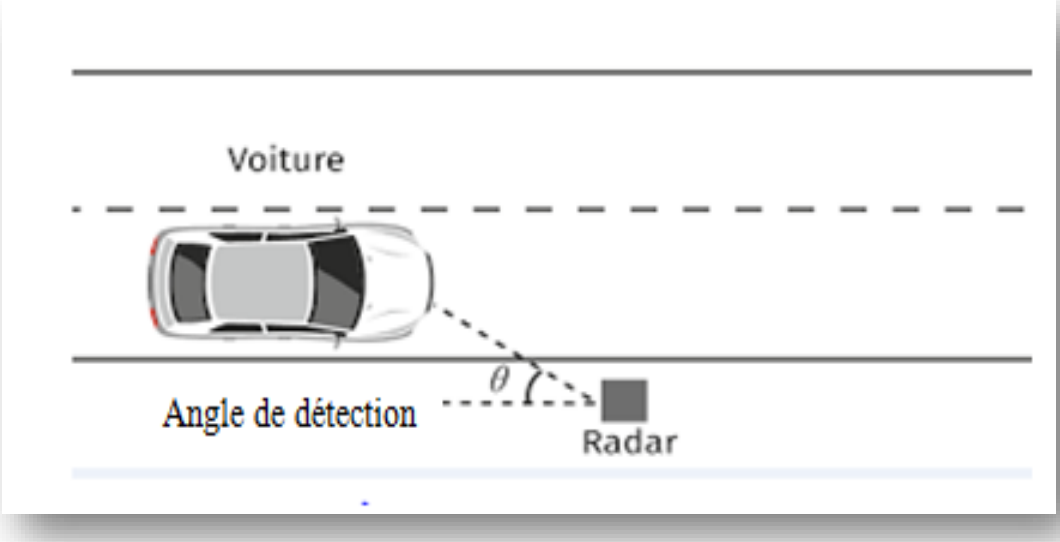

Figure I.5 : Angle de détection

I.7. Conclusion :

Dans ce premier chapitre, nous avons présenté l'état de l'art du sujet. Nous avons expliqué les notions générales sur le radar automobile, l'historique et le mode de fonctionnement d'un système de contrôle routière. Nous avons aussi abordé les différents types de radar automobile et l'angle de détection.

Chapitre II

Etude et réalisation du système de contrôle routier

II.1.Introduction :

Un système de contrôle routier est un instrument servant à mesurer la vitesse des véhicules circulant sur les routes publiques. Avant d'aborder la réalisation de ce système, nous allons présenter des outils, logiciels et matériels tels que le Raspberry Pi utilisé pour la conception et la réalisation.

II.2.Spécification de la technique et langage utilisé :

Pour réaliser notre projet, nous avons utilisé la technique "Haar Cascade", qui permet de détecter les véhicules dans une vidéo en direct.

Afin de faire la programmation, nous avons choisi le langage «Python 3» avec sa bibliothèque « OpenCV » qui facilite les procédures de détection.

II.3.Procédure de détection de Haar-cascade :

Haar-cascade, ou nommée aussi la méthode de Viola et Jones, est une méthode de détection d'objet dans une image numérique. Cette méthode a été proposée en 2001 par Paul Viola et Michael Jones [5], elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Bien que cette méthode avait été spécialement conçue pour détecter des visages, mais les chercheurs ont rendu compte que la méthode s'appliquait aux autres membres du corps, aux objets comme des voitures ou des avions, et de manière générale à la détection de tout type d'image [6]. Par conséquent, cette méthode est devenu un standard en détection d'objet.

La méthode proposée par Viola et Jones se décompose en quatre parties :

a) Partie 1 : Constitution d'une base de données

Dans un premier temps, il faut constituer une base de données composé d'images (des images positives contenant principalement l'objet que nous souhaitons le détecter et des images négatives sans l'objet). Il faut prendre en considération deux choses:

- Les images doivent être proches visuellement de ce que nous souhaitons détecter
- Une certaine quantité d'exemples non négligeable

b) Partie 2 : Caractéristiques pseudo-Haar

Après la constitution de la base de données, Viola et Jones proposent d'utiliser des caractéristiques très simples : les caractéristiques pseudo-Haar. Elles sont des caractéristiques utilisées en vision par ordinateur pour la détection d'objet dans des images numériques, elles sont calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes. Des exemples des premières caractéristiques pseudo-Haar proposées par Viola et Jones sont représentés dans la Figure II.1, dans lesquelles la somme de pixels gris est soustraite de la somme des pixels blancs [5].

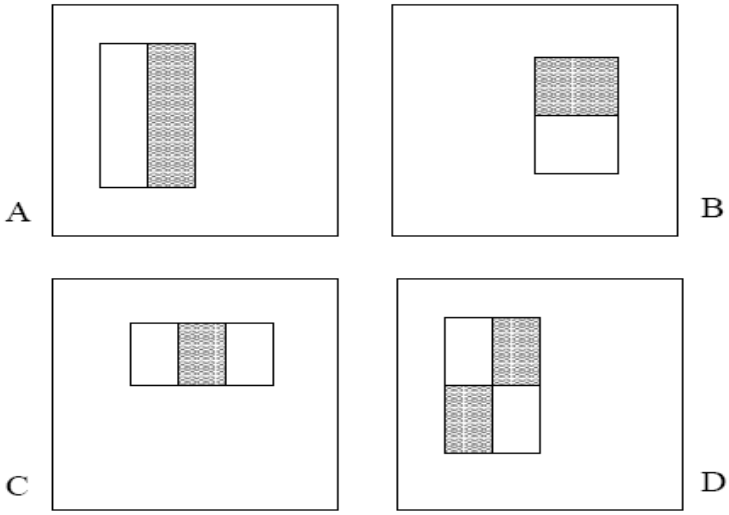

Figure II.1. Exemple des premières caractéristiques pseudo-Haar utilisées par Viola et Jones. Les caractéristiques à deux rectangles sont illustré en (A) et (B). (C) montre une caractéristiques à trois rectangle, et (D) montre une caractéristique à quatre rectangles [5].

Pour calculer rapidement et efficacement les caractéristiques pseudo-Haar, Viola et Jones proposent également une représentation intermédiaire de l'image qu'ils appellent « ***image intégrale*** ». Une image intégrale est une table de correspondance 2D, construite à partir de l'image d'origine, permettant de calculer rapidement des sommes de valeurs dans des zones rectangulaires. Figure II.2 montre un exemple de calcul d'image intégrale ; où La valeur de l'image intégrale au site 1 est la somme des pixels dans le rectangle A, la valeur au site 2 est A + B, la valeur au site 3 est A + C, la valeur au site 4 est A + B + C + D [5].

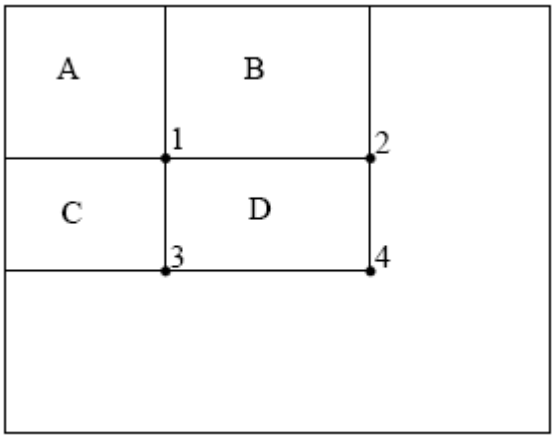

Figure II.2. Exemple de calcul d'image intégrale [5]

c)Partie 3 : Classifieur Haar-cascade

Viola et Jones ont choisi d'utiliser le classifieur AdaBoost (acronyme de ***Ada****ptive* ***Boost****ing*) pour le détecteur de Haar, qui a l'avantage d'avoir une bonne vitesse d'exécution. AdaBoost consiste à sélectionner les meilleures caractéristiques par l'idée de combiner de faibles classifieurs afin d'en obtenir un classifieur fort. Le classifieur en cascade est généralement stocké dans un fichier xml.

d) Partie 4 : Phase de détection

Après entraînement du classifieur par AdaBoost, La détection s'applique sur une image de test, dans laquelle nous souhaitons déceler la présence et la localisation d'un objet.

II.4.Type de fichier utilisé :

II.4.1.Fichier XML :

Le XML est acronyme de *e**X**tensible* ***M****arkup* ***L****anguage* qui signifie : langage de balisage extensible, c'est une langue de description permet de structurer l'information dans des fichiers textes. On peut l'utiliser typiquement comme un fichier de configuration pour des programmes mais aussi pour enregistrer des résultats, cette forme permet d'échanger les données sur réseau Internet parce qu'il est compatible avec le web et il facilite aussi le traitement informatique[7].

II.4.2.Fichier CSV :

Le CSV est le sigle de ***C**oma **S**eparated **V**alues,* son traduction est «valeurs séparées par des virgules». Il est un format texte représentant des données sous forme d'un tableau où chaque champ est séparé par une virgule. La fonction principale des fichiers CSV est de permettre la portabilité des données tabulaires d'un programme à l'autre, et ils peuvent être utilisés lorsque de grandes quantités de données doivent être fusionnées sans être directement connectées les unes aux autres[8].

II.4.3.Niveau de gris :

Niveau de gris (image en gris) est un nuance du noir au blanc, le niveau de gris minimal est 0 et maximum dépend de la profondeur de numérisation de l'image. Pour une image d'une profondeur de 8 bits, c'est 255. Afin de convertir une image en couleur en niveau de gris, il faut remplacer, pour chaque pixel, les trois valeurs représentant les niveaux de rouge, de vert et de bleu (abrégé en *RVB* ou *RGB*) par une seule valeur représentant la luminosité ; et chaque pixel prend une valeur entre 0 et 255. L'objectif d'utilisation du niveau de gris est de réduire l'information fournie pour chaque pixel [9].

II.5.Environnement de développement :

II.5.1.Environnement hardware :

II.5.1.1.Un ordinateur :

Un ordinateur est une machine intelligente du traitement de l'information. L'information traitée par l'ordinateur peut se présenter sous forme numérique, de texte, de dessins, d'images ...Cette machine est capable de collecter des données, d'effectuer des traitements et de diffuser des résultats.

Au sens large, un ordinateur est traditionnellement composé :

- Périphérique d'entrée : les entrées standard comme clavier, souris.
- Périphérique de sortie : les sorties standard comme écran, imprimante.
- Unité centrale ou unité de traitement : est composé des processeurs, des mémoires (disque dur, RAM, ROM), une alimentation électrique, une carte graphique et un boîtier de l'unité centrale qui protège les principaux composants.

II.5.1.2. Un Raspberry Pi :

II.5.1.2.1. Présentation du Raspberry pi :

Le Raspberry Pi est un ordinateur entièrement fonctionnel, il est fourni nu, c'est-à-dire la carte mère seule contrairement à un ordinateur traditionnel qui cache son fonctionnement interne dans un boîtier. Il est conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi.

Le Raspberry Pi est connu comme un mini-ordinateur mono-carte contient un processeur central ARM [10]. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux, notamment Debian, et des logiciels compatibles ; mais il fonctionne également avec le système d'exploitation Microsoft Windows.

II.5.1.2.2.Les modèles de Raspberry Pi :

Depuis sa sortie, Raspberry Pi est proposé dans différentes versions, avec des mises à jour et des améliorations apportées à l'unité Raspberry Pi Model B d'origine, il existe trois modèles de base de Raspberry Pi disponibles:

- **Raspberry Pi modèle A :**

Ce modèle contient 2 types A et A+ :

Modèle A :

Le Raspberry pi A est lancé en 2013 (Figure II.3), il contient une RAM de 256 Mo, 2 sorties vidéo (composite et HDMI), une sortie audio stéréo, et un port USB. Cemodèle ne dispose pas de port Ethernet, et il est moins puissant.

Figure II.3. Raspberry pi modèle A

Modèle A+ :

Le Raspberry Pi A+ est lancé en 2014 (Figure II.4), il est plus petite que le premier modèle et contient un slot microSD. Les inconvénients de ce modèle sont la consommation électronique faible, et il avait qu'un seul port USB et pas de réseau local.

Figure II.4. Raspberry pi modèle A+

❖ Raspberry Pi modèle B :

Modèle 1 B :

Le Raspberry pi B est lancé en 2012 (Figure II.5), il possède deux ports USB sur un seul bus au lieu de l'unique port du modèle A. Il possède 512Mo de RAM et un port Ethernet.

Figure II.5. Raspberry pi modèle 1 B

Modèle 1 B+ :

Le Raspberry pi B+ est lancé en 2014 (Figure II.6), il contient 4 ports USB, meilleur circuit audio et meilleur comportement en cas de surcharge. La consommation électrique est réduite par rapport aux anciens modèles.

Figure II.6. Raspberry pi modèle 1 B+

Modèle 2 B :

Le Raspberry pi 2 B est lancé en 2015 (Figure II.7), il est plus puissant et équipé d'un processeur Broadcom, accompagné de 1 Go de RAM. Il possède les mêmes dimensions et la même connectique que le modèle 1 B+.

Figure II.7. Raspberry pi modèle 2 B

Modèle 3 B :

Le Raspberry pi 3 B est annoncé en 2016 (Figure II.8), il est plus efficace que le Raspberry Pi 2 B. Il est désormais équipé nativement du Wifi et du Bluetooth, et fournit un processeur graphique 64 bits quadricoeur.

Figure II.8. Raspberry pi modèle 3 B

Modèle 3 B+ :

Le Raspberry pi 3 B+ est annoncé en 2018 (Figure II.9), il est une amélioration de type B, Le processeur Broadcom est 200 Mhz plus élevé que son prédécesseur. Il contient un protocole Ethernet plus rapide et d'un réseau local sans fil.

Figure II.9. Raspberrypi modèle 3 B+

Modèle 4 B:

En 2020, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 4 avec un modèle B (Figure II.10), et une version à 8 Go de RAM. C'est le dernier né de la gamme de mini-ordinateurs Raspberry Pi.

Figure II.10. Raspberrypi modèle 4 B

- **Raspberry Pi modèle Zero :**

Modèle Zero :

Ce modèle est annoncé en 2015 (Figure II.11), il fait la moitié de la taille du modèle A + et deux fois plus pratique, et destiné aux applications embarquées. Il a un HDMI et un seul connecteur micro USB mais il ne possède pas de connexion d'affichage intégré.

Figure II.11. Raspberry pi modèle Zero

Modèle Zero W :

Ce modèle est annoncé en 2017 (Figure II.12), le Rasspbery Pi Zero W possède toutes les fonctionnalités du Pi Zero d'origine, mais s'accompagne avec une connectivité Bluetooth et LAN sans fil supplémentaire, et une Mémoire de 512 Mo.

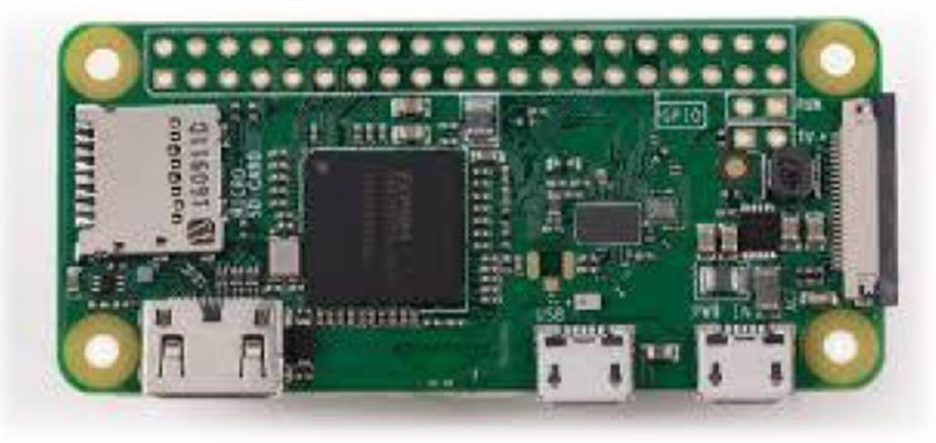

Figure II.12. Rasspbery Pi modèle Zero W

Modèle Zero WH :

En 2018, le Raspberry Pi Zero WH est conçu (Figure II.13), il a la même carte que Raspberry Pi Zero W mais avec un connecteur 40 pins GPIO + Camera + MicroSD.

Figure II.13. Raspberry Pi modèle Zero WH

II.5.1.2.3. Le Raspberry Pi 3 modèle B :

Le Raspberry Pi 3 porte bien son nom de micro-ordinateur, il est 10 fois plus puissant que le première Pi. Ce modèle est en effet équipé d'un nouveau SoC de type BCM2837. Muni de quatre cœurs, le processeur 64 bits du Raspberry Pi 3 fonctionne à 1,2 GHz.

Cette troisième version du modèle B est équipée d'une puce Wifi et du Bluetooth à basse consommation. Plusieurs ports USB pour raccorder différents périphériques. Grâce à ses caractéristiques la carte Raspberry Pi 3 fonctionne avec les systèmes

d'exploitation Linux Android ou Firefox OS installé sur une carte micro SD. On peut l'utiliser aussi comme plateforme de développement, comme serveur web ou comme média centre.

Sans aucune limite, le Raspberry Pi 3 B est le meilleur support pour réaliser notre projet (Figure II.14).

Figure II.14. Raspberry Pi Modèle 3 B utilisé dans notre projet

Les principaux composants du Raspberry Pi 3 Modèle B sont détaillés sur la Figure II.15.

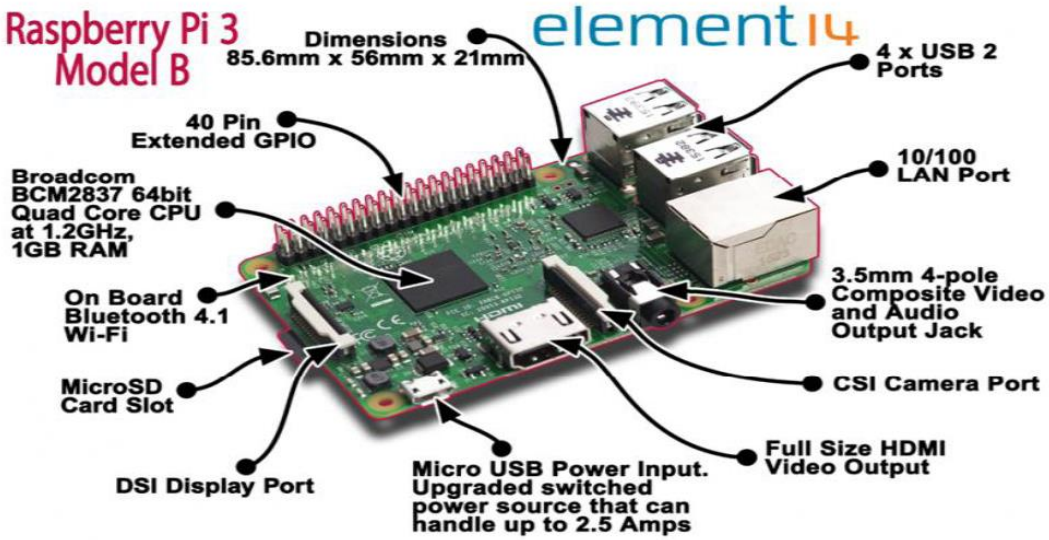

Figure II.15. Les composants standards d'un Raspberry Pi 3 Modèle B

II.5.1.3.Un Ecran TFT LCD 5'':

Un Ecran TFT est un écran hautes performances grâce au module LCD TFT 5 pouces de Displaytech. Il se trouve dans un large éventail d'applications, il est facile de s'interfacer avec un microcontrôleur (Raspberry) ; comme indiqué dans la figure II.16. Un écran LCD (acronyme de ***L**iquid **C**rystal **D**isplay*) est un petit affichage à faible coût. ce module d'affichage est idéal pour une variété de dispositifs numériques tels que des systèmes vidéo parce que L'écran LCD utilise deux circuits d'attaque pour commander l'écran : un circuit d'attaque source avec TCON et un circuit d'attaque de grille. Les deux circuits d'attaque permettent différentes tensions d'alimentation et fonctionnalités de commande de broche[11].

Figure II.16. Ecran TFT LCD 5'' pour Raspberry

II.5.1.4.Une caméra Pi :

Il existe quatre modèles principaux de caméra pour le Raspberry pi, elles servent à prendre des photos et des vidéos. Dans notre projet, nous avons utilisé une caméra de 8 Mégapixels, comme indiquée dans la Figure II.17 ; cette caméra se branche au port avec une nappe.

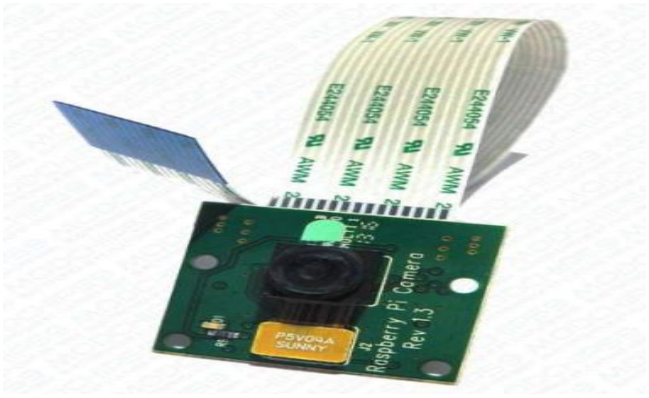

Figure II.17. Camera Pi utilisée dans notre projet

II.5.2.Environnement software :

II.5.2.1. Raspbian :

Raspbian nommé aussi Raspberry Pi OS est un système d'exploitation spécialement conçu pour Raspberry Pi, il est basé sur Linux Debian et il est très régulièrement mis à jour.

Raspbian présente une distribution polyvalente qui permet de familiariser facilement avec le matériel. Il existe plusieurs versions de Raspbian telles que la version buster qui a été utilisée dans notre projet [12].

II.5.2.2.Langage Python :

Python est un langage de programmation, il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. C'est un langage interprété, qui n'a pas besoin d'être compilé pour être exécuté. Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. Il est aussi multiplateforme c'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation comme Windows, Linux, Android, iOS, dès les mini-ordinateurs Raspberry Pi jusqu'aux super calculateurs[13].

II.5.2.3.Bibliothèques (modules) utilisées :

II.5.2.3.1. Time :

Le module time a l'objectif de gérer le temps en python, c'est-à-dire il offre la possibilité de lire, représenter et réinitialiser les informations de temps de nombreuse façon.

Un temps en python est un nombre représentant des secondes. ça permet par exemple d'attendre un certain nombre de secondes, d'afficher une date avec un format spécifique ou encore de connaître le nombre de secondes écoulées depuis une date précis [14].

II.5.2.3.2. Imutils :

C'est une série de fonctions pratiques pour rendre les fonctions de traitement d'image de base comme la traduction, la rotation, le redimensionnement, le squelette,

l'affichage des images, la détection des bords et bien plus encore avec OpenCV et Python [15].

II.5.2.3.3. NumPy :

NumPy est une bibliothèque pour le langage de programmation Python, destinée à manipuler des tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Les tableaux NumPy facilitent les opérations mathématiques avancées et d'autres types d'opérations sur un grand nombre de données. En règle générale, ces opérations sont exécutées plus efficacement et avec moins de code que ce qui est possible en utilisant les séquences intégrées de Python [16].

II.5.2.3.4. OpenCV :

II.5.2.3.4.1. Qu'est-ce que OpenCV :

OpenCV (acronyme d'***Open C****omputer* ***V****ision*) est une bibliothèque open-source libre qui contient plus de 2500 algorithmes qui incluent un ensemble complet d'algorithmes de vision par ordinateur. Ces algorithmes peuvent être utilisés pour détecter, reconnaître et identifier des objets, suivre les mouvements de caméra et aussi suivre des objets en mouvement [17].

OpenCV est une bibliothèque Multi-plateformes : disponible pour Linux/Mac, Windows, iOS et Android, et Multi-langages : python, Java, Matlab, C, Ruby (wrapper).

II.5.2.3.4.2. Historique :

OpenCV a été lancé par Intel en 1999 par Gary Bradsky, et sa première version alpha est sortie en 2000.

La première version 1.0 a été publiée en 2006. Une version 1.1 "pré-version "a été libéré en 2008.

La deuxième version majeure d'OpenCV a été annoncé en 2009, OpenCV 2 inclut des modifications majeures de l'interface C ++ servant à faciliter le développement de nouvelles fonctions et améliorant les performances.

Les versions officielles ont maintenant lieu tous les six mois et le développement est maintenant effectué par une équipe russe indépendante soutenue par des sociétés commerciales [18].

II.5.2.3.4.3. Fonctionnalités :

La bibliothèque OpenCV met à disposition de plusieurs fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques [19]. Cette bibliothèque propose des opérations de traitement des images et des vidéos :

- **Traitement d'images :**

La bibliothèque OpenCV propose la plupart des opérations classiques en traitement bas niveau des images:

- ✓ Lecture, écriture et affichage d'une image.
- ✓ Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- ✓ Lissage, filtrage.
- ✓ Seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- ✓ Segmentation (composantes connexes, GrabCut).
- ✓ Morphologie mathématique.

- **Traitement vidéo :**

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- ✓ Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)
- ✓ Détection de droites, de segment et de cercles par Transformée de Hough
- ✓ Détection d'objet par la méthode de Viola et Jones
- ✓ Cascade de classifieurs boostés
- ✓ Détection de mouvement, historique du mouvement
- ✓ détection de points d'intérêts

II.6. Conclusion :

Dans ce chapitre, nous avons présenté la technique de détection, le matériel et le langage de programmation utilisés dans notre projet.

Dans une première section, nous avons défini la méthode de Viola et Jones (détection de Haar-cascade) en se focalisant sur les principaux étapes de détection ainsi les différents types de fichiers utilisés.

Dans une deuxième section, nous avons exposé les environnements hardware et software, alors nous avons commencé par une présentation de Raspberry Pi ainsi ses différents modèles en justifiant le choix de Raspberry Pi Modèle 3 B, l'écran TFT LCD 5'', et la caméra pi utilisés dans cette étude. Puis, nous avons spécifié le coté software où nous avons résumé le système d'exploitation Raspbian, langage Python et les bibliothèques utilisées.

Chapitre III
Conception et analyse

III.1. Introduction :

Dans ce chapitre d'analyse et conception, nous avons présenté le matériel utilisé et les démarches pour l'installation d'un système d'exploitation, après nous avons présenté le programme de la détection des véhicules et du calcule de leurs vitesses et la détection des matricules ainsi que le stockage des images des matricules dans une base de donné CSV puis nous avons créé la page HTML et on termine notre chapitre par la conception et l'organigramme du système de contrôle routier à base du Raspberry Pi.

III.2. Matériel de conception :

Le matériel utilisé dans cette étude :

a) Raspberry Pi 3 modèle B et ses accessoires

Les accessoires indispensables pour un Raspberry Pi sont représentés dans la Figure III.1.

Figure III.1. Raspberry Pi et ses accessoires

- *Une carte microSD et son adaptateur*

La carte microSD permet le stockage permanent du Raspberry Pi, tous les fichiers créés et les logiciels installés ainsi que le système d'exploitation sont stockés sur la carte microSD. Donc, il faut avoir une carte de bonne qualité et très performante pour ne pas ralentir le Raspberry Pi, et une capacité de stockage plus que 2Go, Pour cela nous avons choisi une carte standard SDHC (classe 10) de 16 Go.

- *Une alimentation*

Le Raspberry Pi comme tous les instruments a besoin d'une alimentation avec un courant de sortie maximum de 2,5 A et une Tension de 5V.

- *Un câble HDMI*

Nous avons utilisé un câble HDMI pour connecter directement le Raspberry Pi à n'importe quel écran.

Dans notre projet, nous avons utilisé un adaptateur HDMI-VGA et un câble VGA.

- *Un clavier et une souris*

b) Une Caméra Pi

c) Un Ecran : TFT LCD 5''

III.3. Préparation du système d'exploitation sur micro SD :

III.3.1. Formater la carte SD :

Même si la carte SD est neuve, il est préférable de la formater avant de copier les fichiers d'installation. Nous avons utilisé le logiciel de formatage SD Card Formatter, comme indiqué sur la Figure III.2.

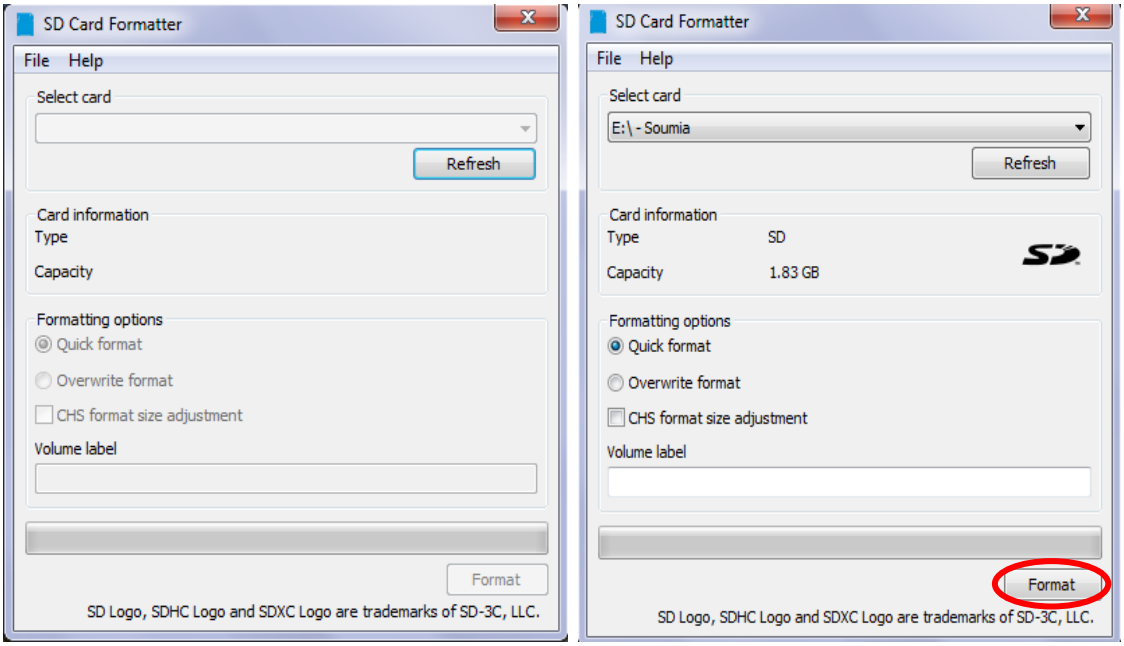

Figure III.2. Formatage de la carte SD

III.3.2. Installer Raspbian sur la carte SD :

Le Raspberry Pi est initialement sans système d'exploitation. D'abord, nous avons téléchargé Raspbian à partir du site officiel : raspberrypi.org/.

Après le téléchargement, nous devons décompresser le fichier .zip pour obtenir un fichier .img (image extraite), comme indiqué sur la Figure III.3.

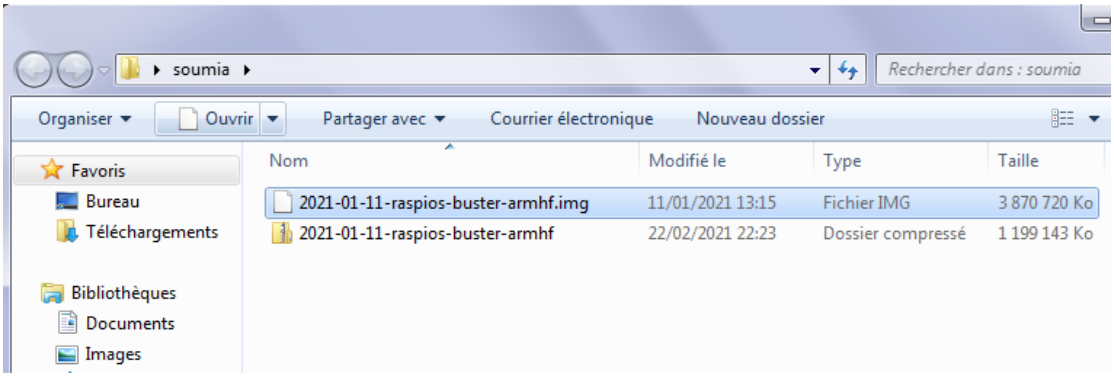

Figure III.3. Décompression du fichier Zip

Puis, nous l'avons installé sur la carte mémoire par le biais d'un logiciel d'écriture d'image : «Win32 Disk Imager ». Il suffit de :

- Sélectionner l'image de Raspbian.
- Choisir la lettre de notre lecteur de carte SD.
- Cliquer sur écrire (Figure III.4).

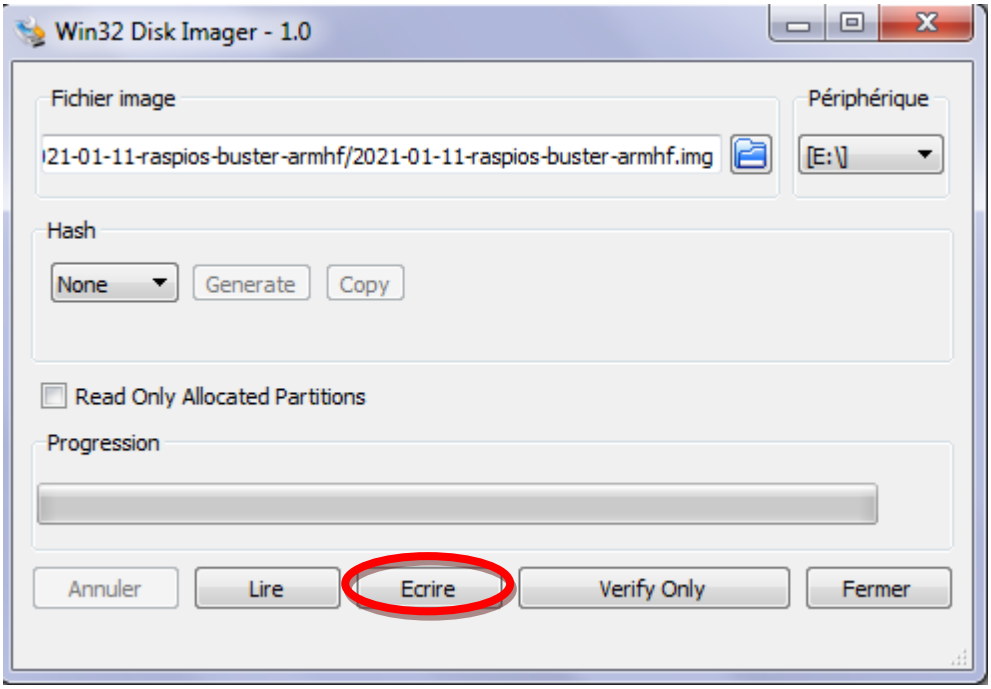

Figure III.4. Copier le raspbian sur la carte SD

Une fois l'écriture terminée, la fenêtre affiche « Terminer » sous la barre de progression (Figure III.5), Le système Raspbian est prête.

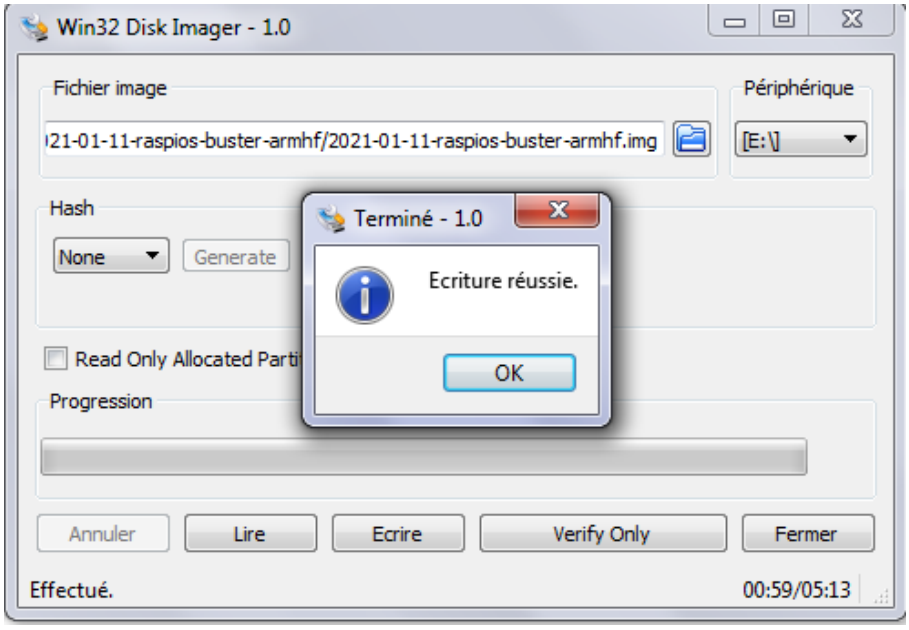

Figure III.5. Installation de Raspbian sur la carte SD

III.3.3 Premier démarrage du système :

Pour le premier démarrage, il faut faire un branchement de tous les équipements ; Après quelque seconde le Raspberry et l'écran s'allument (Figure III.6).

Figure III.6. Démarrage du Raspberry Pi 3

III.4. Première configuration :

Après le démarrage de système, et avant de commencer à l'utiliser, il y a quelques éléments à configurer (Figure III.7):

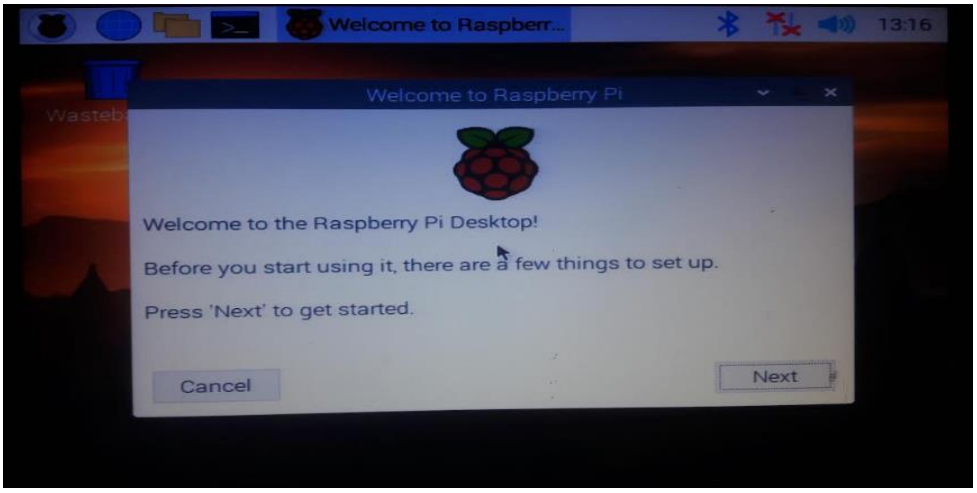

Figure III.7. Début de configuration du Raspberry Pi 3

a) La langue et la localisation :

La configuration de la langue et la localisation est montrée à la Figure III.8.

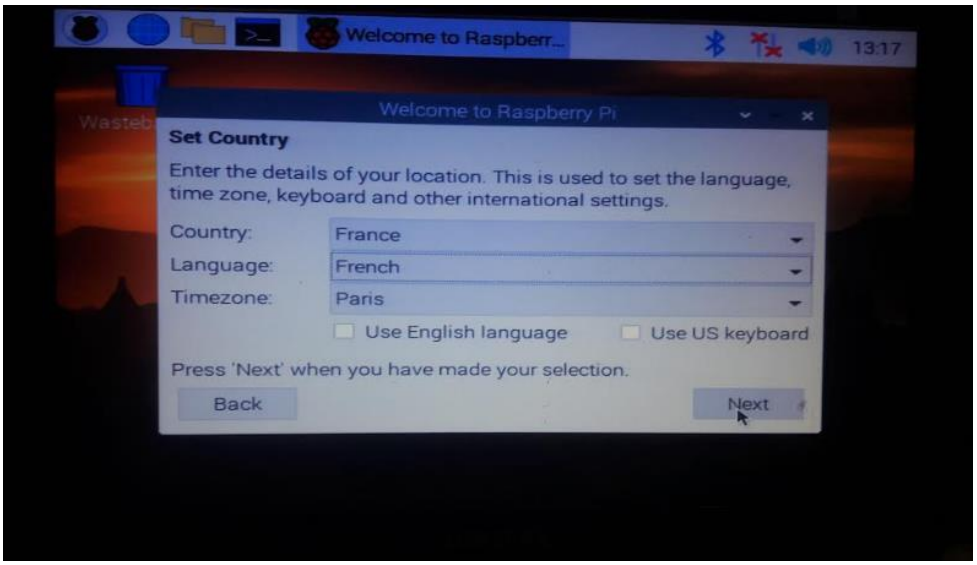

Figure III.8. Configuration de la langue et la localisation

b) Modification du mot de passe :

Le système Raspberry Pi nous oblige de changer le mot de passe, comme indiqué sur la Figure III.9.

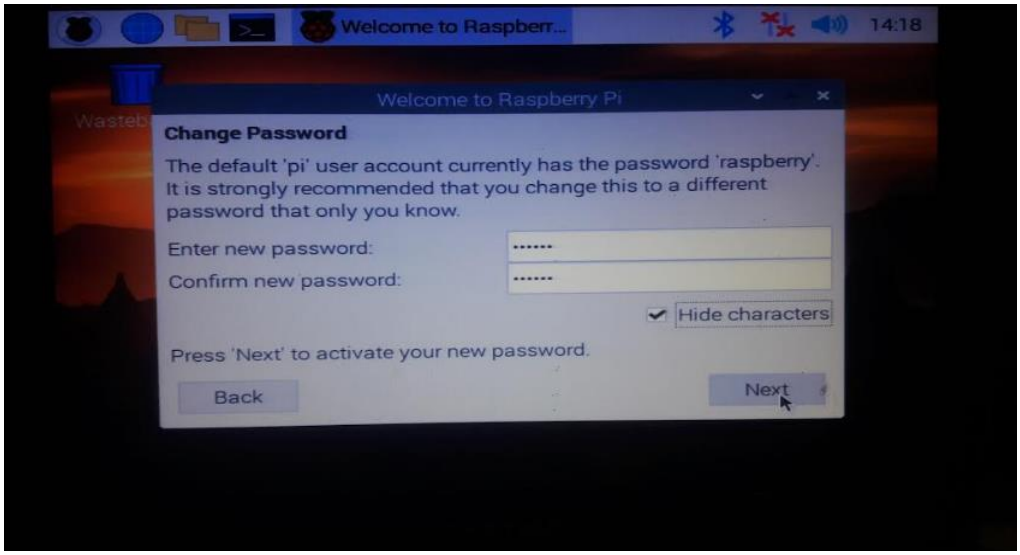

Figure III.9. Modification du mot de passe

c) Activation de wifi :

Pour l'activation de wifi, il faut sélectionner notre réseau et de saisir le mot de passe (Figure III.10). Sinon l'utilisation d'un câble Ethernet pour le connecter directement au box.

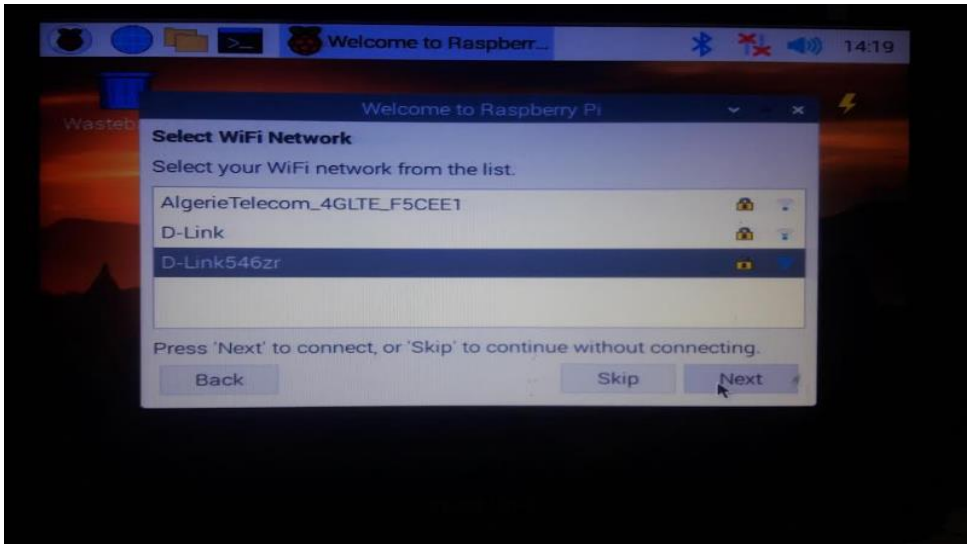

Figure III.10. Activation de wifi

d) L'activation des interfaces utilisées :

- Connexion SSH :

Le protocole SSH (Secure Shell) est un protocole de communication sécurisé dans lequel toutes les données SSH échangées entre deux ordinateurs distants sont cryptées. Ce protocole de sécurité nous a permet de connecter à Raspberry et effectuer

tout ce que nous voulons à partir d'un poste de travail Windows / Mac / Linux sans connecter l'écran à Raspberry.

Pour cela, il faut juste accéder au menu des applications ➜ Préférences ➜ Configuration du Raspberry Pi, une fenêtre s'ouvre nous avons coché le bouton « Activé» sur la ligne SSH (Figure III.11), puis Sélectionné «Terminer» et « Redémarrer » notre Raspberry Pi.

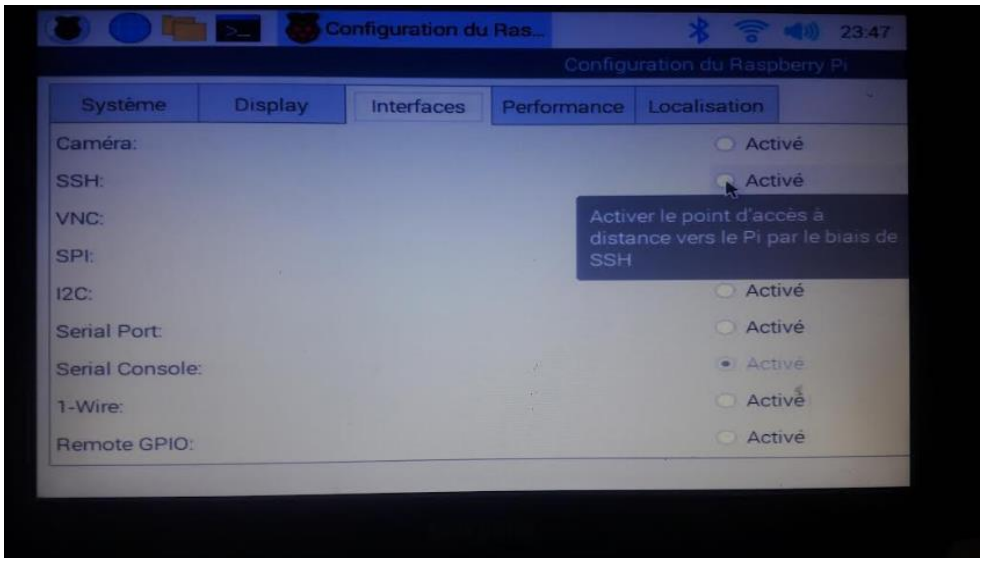

Figure III.11. Activation de SSH

Sous Windows, nous devons installer « Putty » qui est un client SSH, Ensuite, il suffit d'entrer l'adresse IP du Raspberry dans Putty afin de pouvoir se connecter (Figure III.12).

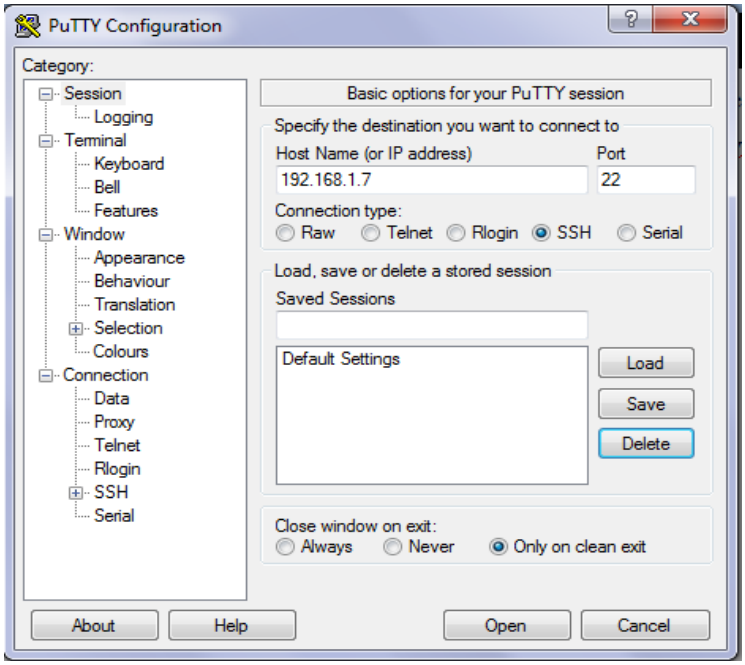

Figure III.12. Configuration de Putty

- Le protocole VNC

Pour activer ce protocole, il faut entrer au menu des applications ➜Préférences ➜ Configuration du Raspberry Pi, nous avons Activé l'interface VNC (Figure III.13), puis Sélectionné «Terminer» et « Redémarrer » notre Raspberry Pi.

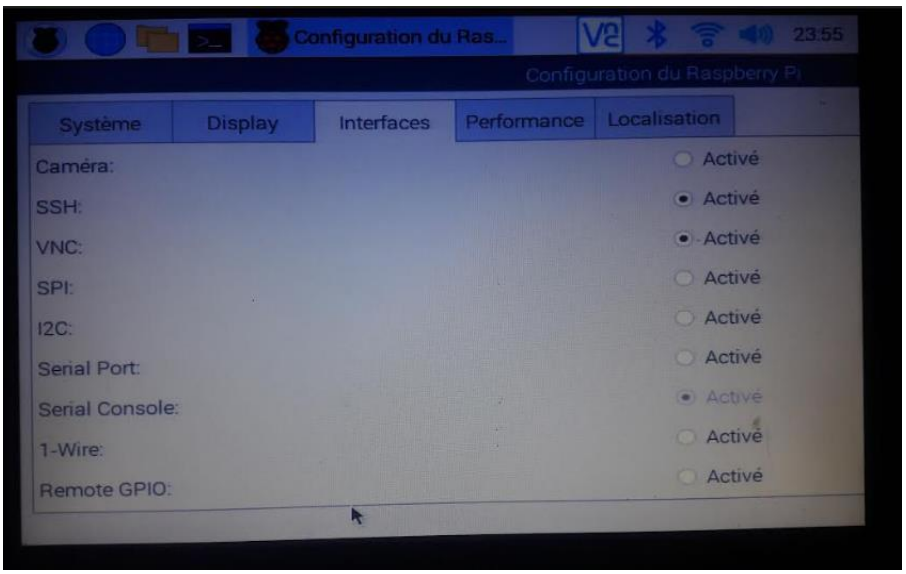

Figure III.13. Activation de VNC

III.5. Mettre à jour le système de la Raspberry Pi :

Mettre à jour le Raspberry Pi est une étape importante pour apporter des correctifs de sécurité, des nouvelles versions de logiciels, de Raspbian…

- Pour mettre à jour la liste des paquets :

```
Sudo apt-get update
```

- Pour télécharger et installer les mises à jour :

```
Sudo apt-get upgrade
```

III.6. Activer la camera Pi :

Pour activer la camera Pi, nous avons ouvert le Menu puis Préférences et enfin le panneau de configuration du Raspberry Pi. Nous sommes allés à l'onglet Interfaces puis nous avons activé le module caméra.

III.7.Installation OpenCV sur le Raspberry Pi :

- Mettre à jour le système d'exploitation :

```
Sudo apt-get update
Sudo apt-get upgrade
```

- Installer les dépendances (Modules OpenCV) :

```
Sudo apt-get install build-essential cmake pkg-config
Sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
Sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv41-dev
Sudo apt-get install libxvidcore-dev libx264-dev
Sudo apt-get install libgtk2.0-dev libgtk-3-dev
dev Sudo apt-get install libatlas-base-dev gfortran
```

- Installation de python 3 et pip 3 :

```
Sudo apt-get install python3-dev
Sudo apt-get install python3-pip
```

- Installer OpenCV :

```
Sudo apt-get install python3-opencv
```

- Dépendance supplémentaire pour OpenCV et la caméra :

```
Sudo apt-get install libqtgui4
Sudo modprobe bcm2835-v412
Sudo apt-get install libqt4-test
```

III.8. Programme python:

Notre système d'exploitation est maintenant bien installé et configuré, le langage python 3 et sa bibliothèque OpenCV sont prêts pour la programmation de notre projet.

Tout d'abord, nous avons créé un répertoire de travail nommé PFE par le bais de la commande :

```
mkdir PFE
```

Le Haar Cascade est fondamentalement un classifieur utilisé pour détecter les objets dans des images numériques, il a été conçu pour détecter les visages mais après il a réussi à tout type d'image (détails en Chapitre 2). Dans notre projet, le classifieur Haar cascade a détecté tout ce qui bouge (arbres, véhicules, animaux....) et en même temps

il sélectionne les objets fermés comme les véhicules. Le classifieur cascade est utilisé comme un fichier XML, le classifieur que nous avons utilisé est disponible sur GitHub.

Après le téléchargement de ce fichier, nous avons le placé dans le répertoire de travail (PFE), comme indiqué sur la Figure III.14.

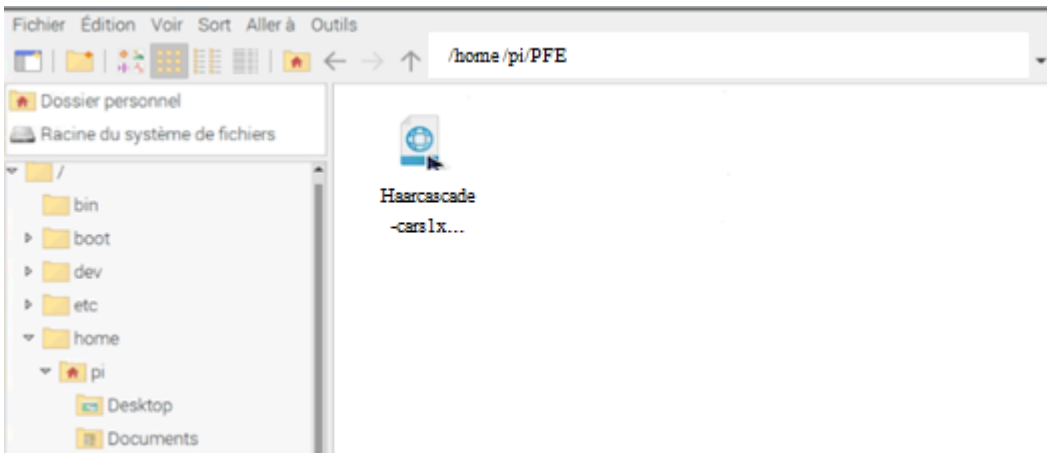

Figure III.14. Fichier Haar cascade placé dans PFE

Pour écrire notre programme python, nous avons créé un nouveau script PFE1.py par la commande :

```
Sudo nano PFE1.py
```

Le code python :

Nous avons importé les bibliothèques et les modules:

```
import time
import cv2
import numpy as np
import imutils
import time
from datetime import datetime
from time import strftime
```

Le module **TIME** sert a la manipulation simple du temps dans un programme python et le module **datetime** affiche et formate des dates et heures avec une méthode un peu plus orientée objets (la date et l'heure seront des objets). **NumPy** est une bibliothèque

destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux, ainsi que **imutils** présente une série de fonctions pour OpenCV.

Après, nous avons utilisé la classe VideoCapture fournit par **OpenCV** pour l'acquisition vidéo en provenance d'une caméra pi et on va faire une instance cap sous la relation **cap = cv2.VideoCapture(0)** avant d'appeler **cap.release()** pour l'affichage de la vidéo

Nous avons utilisé la fonction **cv2.VideoWriter()** pour sauvegarder une vidéo enregistrée, cette fonction à 4 paramètres : le nom du fichier vidéo de sortie, la fréquence d'images définie du flux vidéo de sortie (20Hz) et la taille des images vidéo (640, 480) et fourcc pour définir le codec : est un code de 4octets utilisé pour spécifier le codec vidéo(XVID).

```
cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640,  480))
```

Nous avons appelé le fichier cascade précédemment téléchargé :

```
car_cascade = cv2.CascadeClassifier('cars1.xml')
```

Le code initie une boucle infinie (qui sera interrompue plus tard par une instruction break). La fonction cap.read () utilisée pour lire les images depuis video capture et ret (_) est un booléen vérifie si la lecture a réussi, nous continuons à utiliser le retour frame ; et si ce n'est pas le cas nous revenons.

Après, nous avons redimensionné le frame par la fonction resize () (la largeur : 640, la hauteur :480).Ensuite, nous avons converti le canal BGR de chaque capture en canal gris (le canal gris est facile à traiter et nécessite moins de calculs car il ne contient qu'un seul canal de noir-blanc).

Dans la dernière ligne de cette partie de code, nous utilisons la fonction « detectMultiScale » qui détecte des objets de différentes tailles dans l'image d'entrée et renvoie des rectangles positionnés sur les véhicules. Le premier argument est l'image d'entrée (image en gris), le second est le facteur d'échelle (1.1) et le troisième est le nombre de voisins pour chaque rectangle (2).

```
while(True):
    _, frame = cap.read()
    frame = cv2.resize(frame,(640,  480))
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cars = car_cascade.detectMultiScale(gray, 1.1, 2)
```

Dans cette partie, nous avons tracé deux lignes verticales de la droite vers la gauche par le module cv2 et nous avons défini chaque ligne par deux points (le départ et la fin), la couleur et l'épaisseur.

```
ax1=350
ax2=300
ay1=0
ay2=600
   cv2.line(frame,(ax1,ay1),(ax1,ay2),(255,0,0),1)
   cv2.line(frame,(ax2,ay1),(ax2,ay2),(255,0,0),1)
```

Nous allons parcourir chaque rectangle (chaque voiture détecté) en utilisant ses coordonnées générées par la fonction « detectMultiScale »

Nous dessinons le rectangle dans l'image originale que nous capturons :

Le (0,0, 255) est la couleur du cadre en RVB

Le (2) est l'épaisseur du rectangle

Les coordonnées x, y, w et h sont respectivement la position initiale horizontale, la position initiale verticale, la largeur et la hauteur.

```
for (x,y,w,h) in cars:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
```

Pour la démonstration de la détection des voitures on a pris comme exemple une voiture de notre essai vidéo présentée comme suivant :

Figure III.15. Détection des voitures à l'aide de Haar-cascade

Après, nous avons utilisé deux boucles :

- Tandis que la première ligne = la moitié de la largeur d'un rectangle tracé, nous avons commencé à calculer le temps par la fonction time.time ().
- Tandis que la première ligne >= la moitié de la largeur d'un rectangle tracé et si la deuxième ligne aussi. Nous avons calculé la vitesse en (m/s) et la convertir en (km/h) sachant que la distance entre les deux lignes est connue (10m).

```
font = cv2.FONT_HERSHEY_DUPLEX
start_time = 0
    while int (ax1) == int((x+x+w)/2):
            start_time = time.time()
            break
    while int (ax1) >= int((x+x+w)/2):
        if int(ax2) >= int((x+x+w)/2):
            speed = int((10*3600)/((time.time() - start_time)*1000))
            cv2.putText(frame,str(speed)+' Km/h', (550,380),font, 0.4, (255,255,255),1);
```

Afin de terminer cette partie, nous avons utilisé la méthode putText () pour écrire la vitesse calculé en (km/h) sur frame (l'image capturé), cette méthode a plusieurs

arguments : la couleur (255,255,255), l'épaisseur (1), facteur d'échelle de police (0.4), le font : type de police (FONT_HERSHEY_DUPLEX) et les coordonnées de point d'où nous commençons à écrire(550,380) comme le montre la figure suivante pour le même exemple

Figure III.16. Calcule et affichage de la vitesse

Après la détection des véhicules et le calcul de leurs vitesses et les affichées, nous avons passé au deuxième objectif de ce programme qui est la détection de la plaque d'immatriculation des voitures.

Nous définissons img qui présente une capture de la détection des voitures en même que le calcule de la vitesse, et c'est dans cette image qu'on fera notre traitement pout la détection de la plaque d'immatriculation. Nous avons redimensionné l'image (Fig. III.17-a) à la taille requise, puis nous mettons en niveaux de gris (Fig. III.17-b)

```
img=frame[y:y+h, x:x+w]
img = cv2.resize(img, (600,400) )
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Dans cette partie de code, nous avons utilisé un filtre bilatéral **(Flou),** ce filtre permet de supprimer les détails indésirables de l'image en niveau de gris (Fig. III.17-c).

Après nous utilisons la fonction canny () qui permet la détection des contours (Fig. III.17-d)

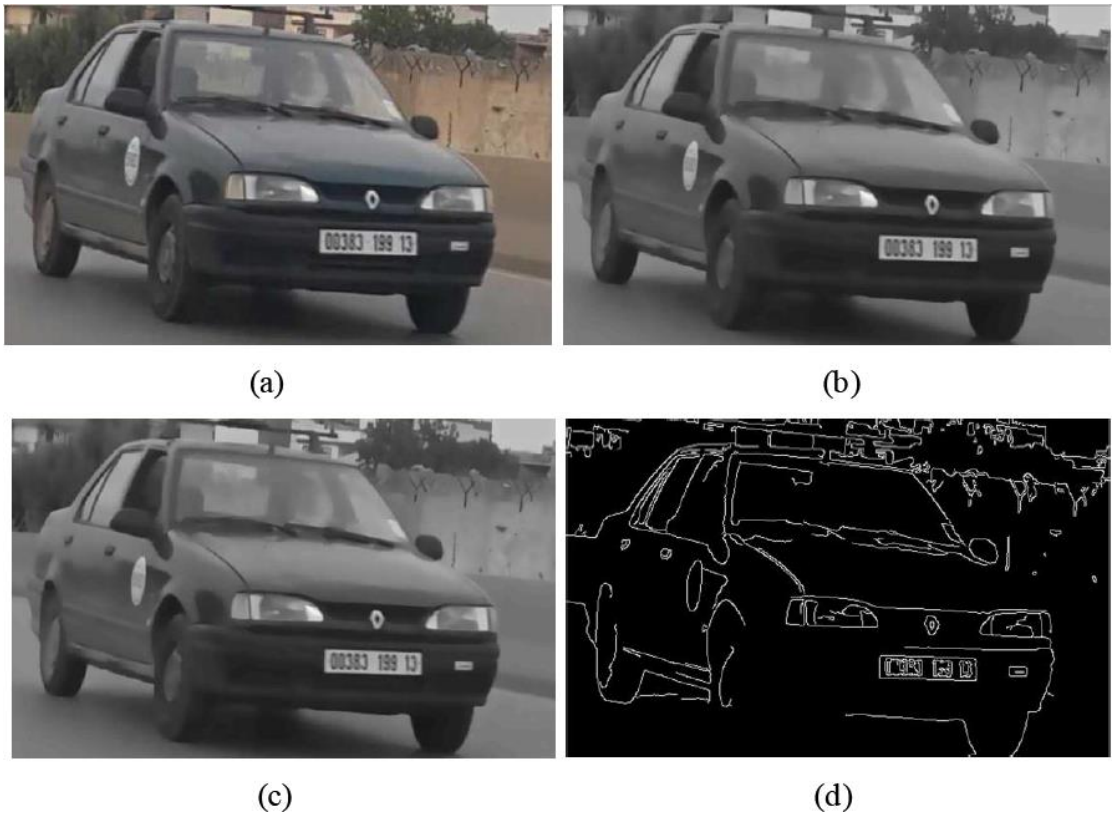

Figure III.17. Etapes du traitement d'image pour la détection de la plaque d'immatriculation

Canny() est une méthode utilise un algorithme de détection des contours astucieux pour trouver les contours de l'image. Elle contient 3 paramètres: l'image source, la valeur de seuil haut et la valeur de seuil bas. Seuls les bords qui ont un gradient d'intensité supérieur à la valeur seuil minimum et inférieur à la valeur seuil maximum seront affichés. Cet algorithme est basé sur des images en niveaux de gris.

```
gray = cv2.bilateralFilter(gray, 13, 15, 15)
edged = cv2.Canny(gray, 30, 200)
```

Pour rechercher les contours sur notre image, nous avons pris tout ce qui est une surface fermée. Nous avons trié les images du plus gros au plus petit et considéré que les 10 premiers résultats en ignorant les autres .La plaque d'immatriculation est également présente car il s'agit aussi d'une surface fermée.

```
contours = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(contours)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:10]
screenCnt = None
```

Pour filtrer l'image de la plaque d'immatriculation parmi les 10 résultats obtenus, nous avons vérifié laquelle a un contour en forme de rectangle avec quatre côtés (une figure fermée).

Une fois trouvée la bonne figure, nous avons l'enregistré dans une variable appelée screenCnt, puis nous avons dessiné un rectangle autour de la variable pour assurer que nous avons correctement détecté la plaque d'immatriculation.

```
for c in contours:
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
    if len(approx) == 4:
        screenCnt = approx
        break
if screenCnt is None:
    detected = 0
else:
    cv2.drawContours(img, [screenCnt], -1, (0, 0, 255), 3)
```

Une fois détecté la plaque d'immatriculation, tous ce qui reste ce n'est pas important pour nous, donc nous allons utiliser un masque pour masquer toute l'image sauf l'endroit où se trouve la plaque d'immatriculation comme le montre la figure

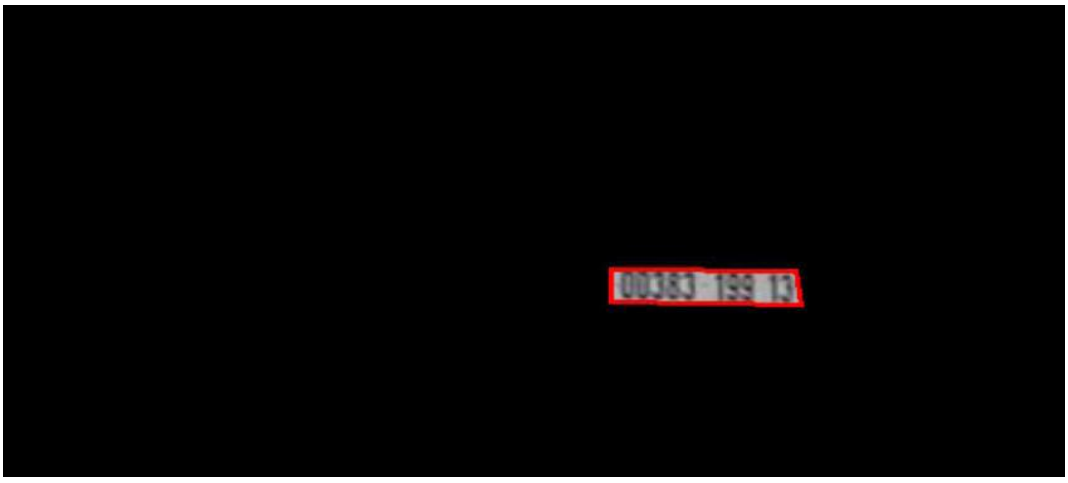

Figure III.18. Résultat après application du masque

```
mask = np.zeros(gray.shape,np.uint8)
new_image = cv2.drawContours(mask,[screenCnt],0,255,-1,)
new_image = cv2.bitwise_and(img,img,mask=mask)
```

Dans cette dernière partie de code, nous avons segmenté la plaque d'immatriculation de l'image en la recadrant et en l'enregistrant en tant que nouvelle image (Cropped) montrée en Fig. III.19

```
(x, y) = np.where(mask == 255)
(topx, topy) = (np.min(x), np.min(y))
(bottomx, bottomy) = (np.max(x), np.max(y))
Cropped = gray[topx:bottomx+1, topy:bottomy+1]
```

Figure III.19. Recadrage de la région du matricule

Nous pouvons ensuite utiliser cette image pour détecter les caractères et les chiffres qu'elle contient en utilisant la technique **OCR** (**Optical Character Recognition**) - Reconnaissance Optique de Caractères qui, à partir d'un procédé optique, permet à un système informatique de lire et de stocker un texte a partir dune image.

Après la détection des matricules, nous avons besoin d'une base de donné CSV (un fichier CSV) pour stocker les images des plaques d'immatriculation.

Pour créer ce fichier nous avons utilisé la commande :

```
Sudo nano vitesse.csv
```

Puis, nous avons utilisé la fonction imwrite () pour enregistrer les images (Cropped) qui contient les plaques d'immatriculation en spécifiant le chemin de la base de données (vitesse.csv), le format d'image (.jpg) et le nom de chaque figure (fig).

```
i=1
cv2.imwrite('/var/www/html/fig'+str(i)+'.jpg',Cropped)
```

Ensuite, nous avons ouvert le fichier vitesse.csv et on ajoute l'heure actuelle (heures, minutes et seconds) et la date actuelle (jour, mois, année) et la vitesse des véhicules calculée (Figure III.20), et nous avons fermé le fichier.

```
TIME = datetime.now().strftime('%H:%M:%S')
DATE = datetime.now().strftime('%A %d %B %Y')
        file = open('/var/www/html/vitesse.csv','a')
        file.write(DATE+' - '+TIME+','+speed+','+'fig'+str(i)+'.jpg'+',\n')
        file.close ()
        i=i+1
```

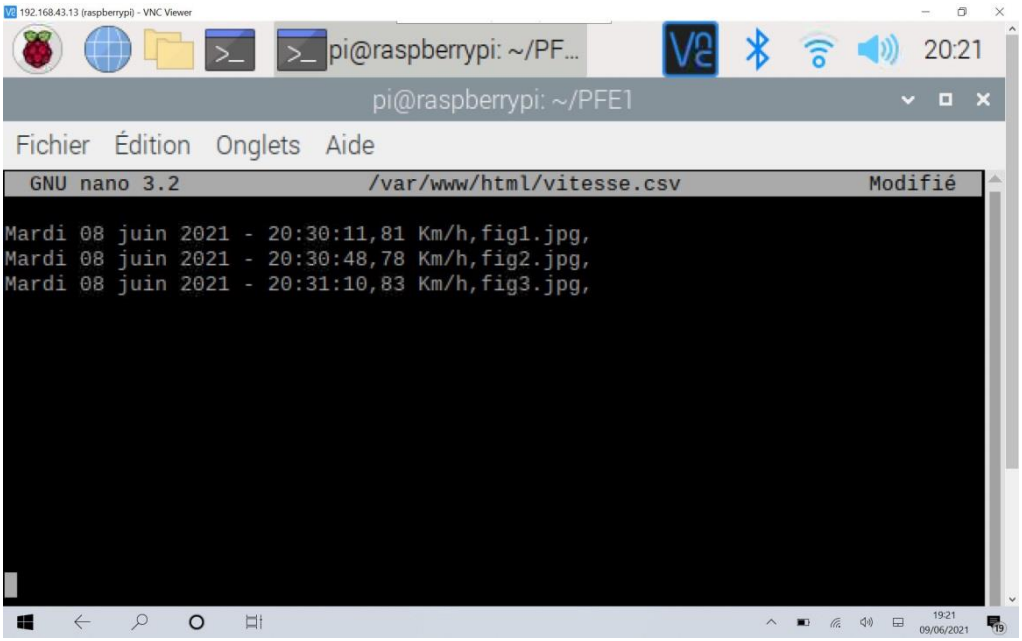

Figure III.20. Fichier CSV

Après, nous avons utilisé la fonction « imshow »pour afficher la partie de la plaque d'immatriculation (Cropped), comme le montre la FigureIII.21.

```
        cv2.imshow('Cropped',Cropped)
        break
else:
        break
```

Figure III.21. Exemple de la détection de la plaque d'immatriculation

Puis, nous avons ajouté la localisation, la date et l'heure actuelle sur le cadre de la vidéo capturé.

```
cv2.putText(frame, "Tlemcen ", (2,10),font, 0.4, (255,255,255),1);
cv2.putText(frame, "- "+strftime("%d/%m/%Y"), (60,10),font, 0.4, (255,255, 255) ,1);
cv2.putText(frame, "- "+strftime("%H:%M:%S"), (160,10),font, 0.4, (255, 255,255),1);
```

Dans cette dernière partie de code, nous avons affiché l'enregistrement de la vidéo capturé, après nous avons utilisé la fonction waitKey (1) pour arrêter le programme si nous appuyons sur le bouton échap et le capture vidéo est fermé et tous les fenêtres imshow ().

```
    cv2.imshow ('Enregistrement', frame)
    out.write(frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release ()
out.release ()
cv2.destroyAllWindows ()
```

III.9.Création de la page HTML :

L'étape suivante est d'afficher le contenue de la base de donné 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 devrons 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>
<body>
<?php
$objCSV = fopen("vitesse.csv", "r");
?>
<table
width = "600";>
```

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 vitesse des véhicules calculés.
- 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"">VITESSE</div></th>
<th width="80"> <div align="center"">MATRICULE</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 $objArr[1];?></td>
<td><div align="center"><?php echo "<img src=".$objArr[2]. " height='60' width='180'>";?></div></td>
</tr>
<?php
}
fclose($objCSV);
?>
</table>
</body>
</html>
```

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. Un exemple de la page web créée est illustré sur la Figure III.22.

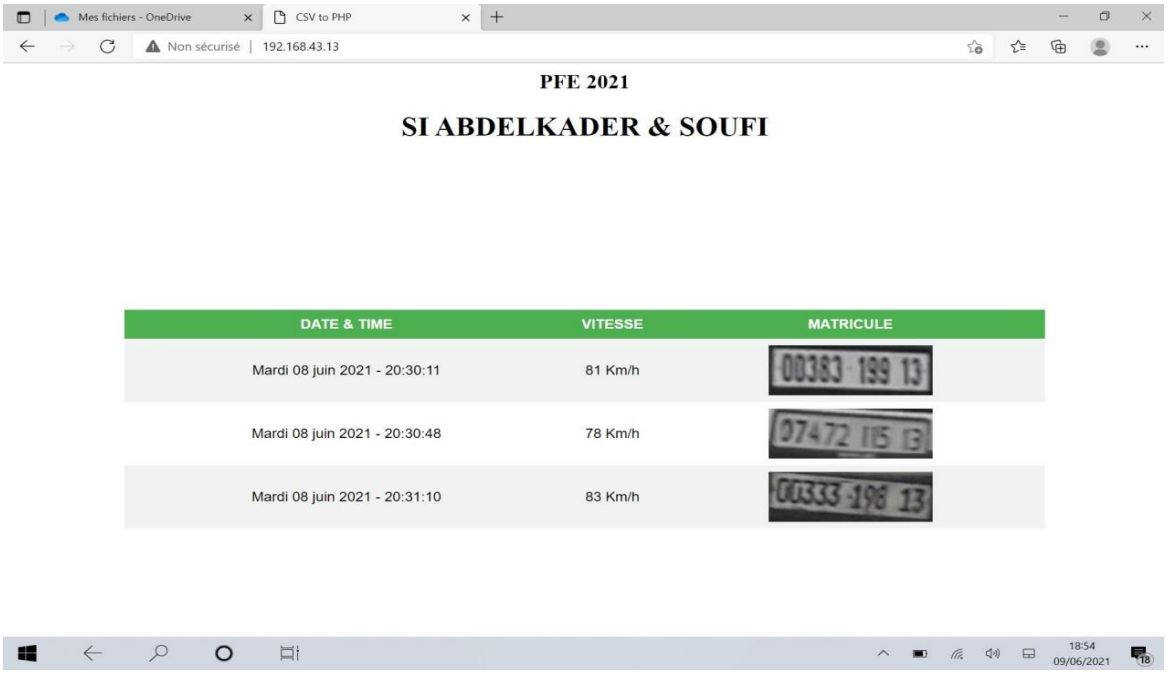

Figure III.22. Exemple de la page web créée

III.10. Comment pouvons-nous avoir un système autonome ?

Afin d'avoir un système autonome, il faut que le système soit :

- Autonome en termes d'alimentation : nous utilisons une alimentation interne (batterie).
- Autonome en termes d'exécution : nous utilisons un auto démarrage c'est-à-dire une exécution du programme au moment où notre Raspberry Pi démarre, pour cela, nous utilisons la commande suivante pour lancer l'éditeur de texte nano et édité le fichier autostart :

```
sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

Une fois le fichier autostart ouvert, nous ajoutons l'instruction d'exécution de notre programme : **Sudo python3 PFE1.py** et après, nous avons enregistré et quitté l'éditeur nano, et nous arrêtons notre système par la commande :

```
Sudo shutdown  now
```

III.11. Conception et Organigramme du système de contrôle routier à base de Raspberry :

III.11.1. Conception du système :

Pour la réalisation de notre projet "système de contrôle routier", nous avons suivi les étapes suivantes :

- La détection des véhicules à partir d'une vidéo
- Le calcul de vitesse des véhicules détectés
- La détection des plaques d'immatriculation
- La création d'une base de données (CSV) pour le stockage
- L'installation d'Apache 2 et PHP
- La conversion de CSV en PHP
- L'affichage sur une page web

III.11.2. Organigramme du système :

La Figure III.23 montre l'organigramme du système de contrôle routier que nous avons étudié.

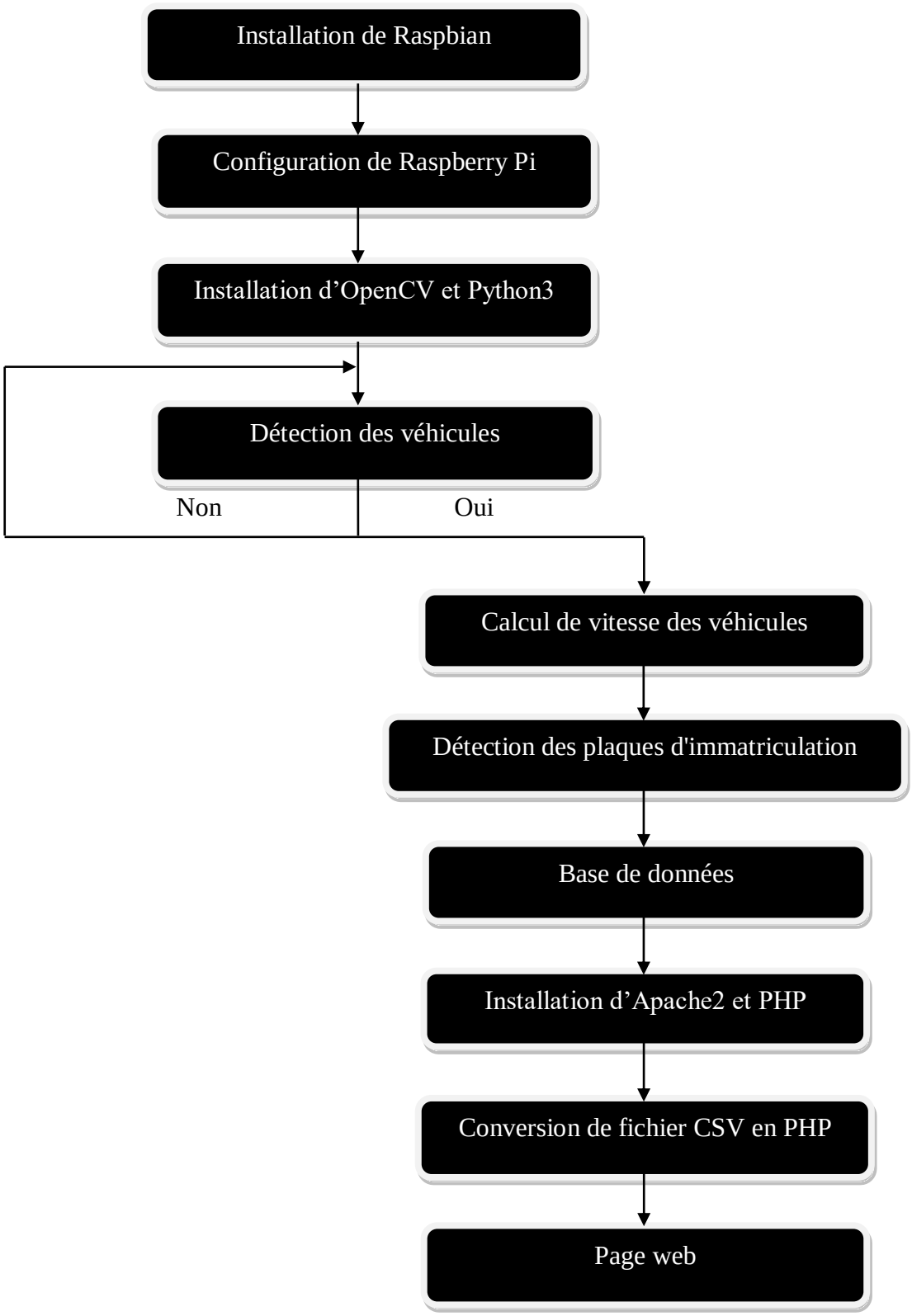

Figure III.23.Organigramme du système

III.12. Circuit du système:

La Figure III.24 illustre le câblage des différents modules constituant le système.

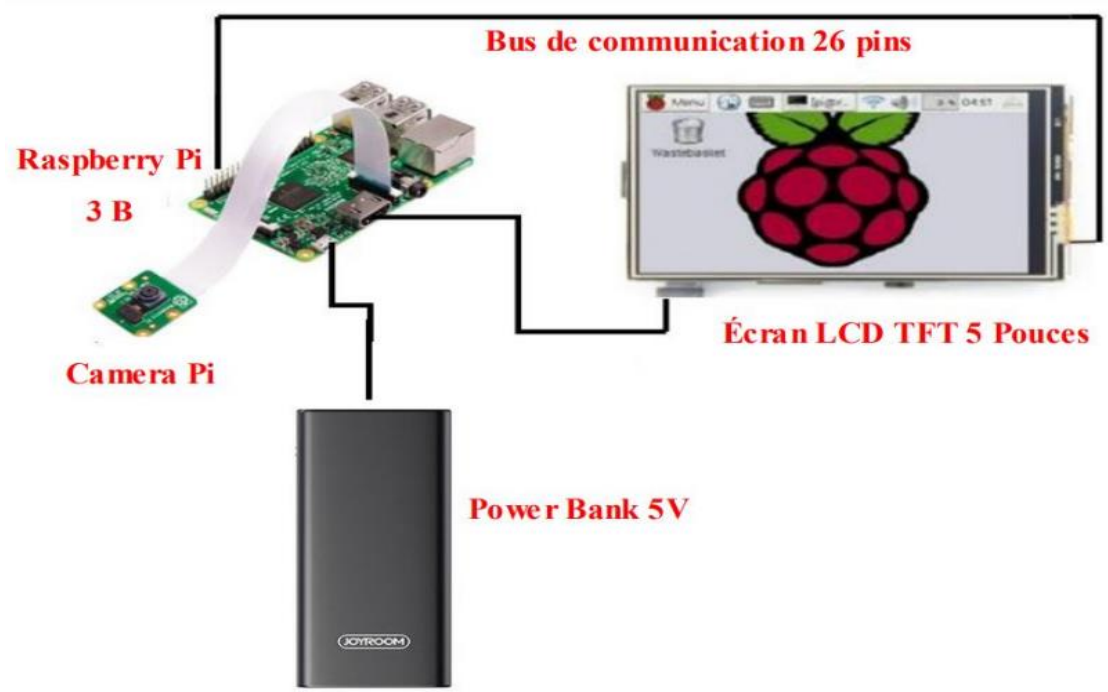

Figure III.24.schéma descriptif du système

Le montage expérimental est montré dans la Figure III. 25.

Figure III.25.Montage expérimental global

III.13. Conclusion:

Dans ce dernier chapitre, nous avons présenté les étapes de l'analyse et la conception du radar de détection à base d'un Raspberry Pi. Ces étapes ont consisté en l'interconnexion entre les différents composants, ensuite la configuration et l'installation d'un système d'exploitation, du langage utilisé pour la programmation. Enfin nous avons présenté la réalisation pratique de ce projet.

Conclusion générale

Notre mémoire a pour objectif l'étude et la réalisation d'un radar de détection 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 routier. Nous nous sommes intéressés aux différents types et le principe de fonctionnement du radar automobile.

La deuxième partie qu'est l'étude, où nous avons présenté la technique de détection, le matériel et le langage de programmation utilisés dans notre projet. Nous avons commencé par la définition de la méthode de Viola et Jones (détection de Haar-cascade) en se focalisant sur les principaux étapes de détection ainsi que les différents types de fichiers utilisés. 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, l'écran TFT LCD 5'', et la caméra pi utilisés dans cette étude. 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, où nous avons présenté les étapes de la conception et l'analyse. En premier lieu, nous avons commencé par l'installation et la configuration du système d'exploitation, le langage de programmation Python 3 et sa bibliothèque OpenCV. Ensuite, nous avons écrit un programme qui permet la détection des voitures et leurs plaques d'immatriculation, puis le calcul de leurs vitesses. Nous avons par la suite crée une page HTML pour stoker les informations collectées. Enfin, nous avons rendu notre système autonome. Pour conclure cette étude, nous avons essayé de faire un organigramme et un schéma descriptif du système réalisé.

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 vitesse limite qui ne doit pas être dépassée par les conducteurs.

Références Bibliographique

[1] A. Mcauslin, “SafeAutonomousSystems: Centralized or distributedprocessing?”, October 27, 2016, www.blog.nxp.com

[2] Éditeur: Christian Wolff, Traduction En Langue Française Et Révision:Pierre Vaillant Et Christophe Paumier, Version 9 Juillet 2011.
Site Internet:Www.Radartutorial.Eu.

[3] https://www.securite-routiere.gouv.fr/radars/differents-types-de-radars

[4]https://fr.wikipedia.org/wiki/Radar_de_contr%C3%B4le_routier#R%C3%A9f%C3%A9rences

[5] P. Viola, M. Jones, Rapid Object Detection using a boosted cascade of simple feature, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. December 2001.

[6] https://kongakura.fr/article/haar-cascade

[7]https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/

[8] https://debitoor.fr/termes-comptables/fichier-csv

[9] https://techlib.fr/definition/grayscale.html

[10] H. Gareth, “THE OFFICIAL Raspberry PiBeginner’s Guide,“ 2018.

[11]https://www.digikey.fr/fr/product-highlight/d/displaytech/5-inch-color-tft-lcd-display

[12]https://www.ionos.fr/digitalguide/serveur/know-how/systeme-exploitation-raspberry-pi/

[13] https://python.sdv.univ-paris-diderot.fr/cours-python.pdf

[14] https://he-arc.github.io/livre-python/time/index.html

[15] https://pydigger.com/pypi/imutils

[16] http://www.numpy.org/

[17] https://opencv.org/about/

[18] https://www.memoireonline.com/

[19] https://fr.wikipedia.org/wiki/OpenCV

Résumé :

Un système de contrôle routier est un instrument de détection des véhicules et mesure des vitesses, ce système est utilisé pour diminuer le nombre d'accident de la circulation et de décès.

Notre projet consiste à réaliser un radar de contrôle routier 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 logicielles.Puis,nous avons utilisé la technique de la détection des véhicules, leurs vitesses et aussi lesplaques d'immatriculation. Enfin, nous avons stocké les informations collectéesdans une base de donné et créé une page HTML contient ces informations.

***Mots-clés*:**Radar, Système de contrôle, Raspberry Pi, ClassifieurHaar cascade, Python3.

Abstract :

A road control system is an instrument for detecting vehicles and measuring speeds, this system is used to reduce the number of accidents and deaths.

Our project consists to realize a radar system using a Raspberry Pi. For this, we presented the used equipment and the procedures of installing an operating system and software. Then, we used the technique of detecting vehicles, their speeds and also the license plates. Finally, we stored the collected informations in a database and created an HTML page containing these informations.

***Key-words*:** Radar, Control system, Raspberry Pi,Haar Cascade classifier,Python3.

ملخص

نظام التحكم على الطريق هو جهاز لكشف المركبات و قياس السرعات, ويستخدم هذا النظام لتقليل من عدد حوادث المرور و الوفيات.

ينص مشروعنا على إنجاز رادار للتحكم في الطريق باستخدام Raspberry Pi. لهذا, قدمنا المعدات المستخدمة و إجراءات تثبيت نظام التشغيل و البرامج. ثم استخدمنا تقنية الكشف عن المركبات و سرعتها و كذلك لوحات الترخيص.

أخيرا, قمنا بتخزين المعلومات التي تم جمعها في قاعدة بيانات و إنشاء صفحة HTML تحتوي على هذه المعلومات.

كلمات مفتاحية: رادار،نظام التحكم, Raspberry Pi , مصنف Haar Cascade, Python3.